BIG DATA ANALYTICS LAB
FACULTY MANUAL

IV YEAR I SEMESTER

R20

Prepared by:

Dr. B Srinuvasu Kumar
Associate Professor

Mr. B Sobhan Babu
Assistant Professor

Mrs. M Divya Sumitra
Assistant Professor

Mr. M Rama Koteswara Rao
Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadrirao Knowledge Village, Gudlavalleru – 521356

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK,Kakinada)
Seshadrirao Knowledge Village, Gudlavalleru – 521356

INSTITUTE VISION & MISSION

Institute Vision:

To be a leading institution of engineering education and research, preparing students for leadership in their fields in a caring and challenging learning environment.

Institute Mission:

- To produce quality engineers by providing state-of-the-art engineering education.
- To attract and retain knowledgeable, creative, motivated and highly skilled individuals whose leadership and contributions uphold the college tenets of education, creativity, research and responsible public service.
- To develop faculty and resources to impart and disseminate knowledge and information to students and also to society that will enhance educational level, which in turn, will contribute to social and economic betterment of society.
- To provide an environment that values and encourages knowledge acquisition and academic freedom, making this a preferred institution for knowledge seekers.
- To provide quality assurance.
- To partner and collaborate with industry, government, and R&D institutes to develop new knowledge and sustainable technologies and serve as an engine for facilitating the nation's economic development.
- To impart personality development skills to students that will help them to succeed and lead.
- To instil in students the attitude, values and vision that will prepare them to lead lives of personal integrity and civic responsibility.
- To promote a campus environment that welcomes and makes students of all races, cultures and civilizations feel at home.
- Putting students face to face with industrial, governmental and societal challenges.

DEPARTMENT VISION & MISSION

VISION

To be a centre of innovation by adopting changes in Information Technology, imparting quality education, researchto produce visionary computer professionals and entrepreneurs.

MISSION

- To provide an academic environment in which students are given the essential resources for solving real-world problems and work in multidisciplinary teams.
- To impart value based education and research among students, particularly belonging to rural areas, for their sustained growth in technological aspects and leadership.
- To collaborate with the industry for making the students adoptable to evolving changes in Information Technology and related areas.

PROGRAM EDUCATIONAL OBJECTIVES(PEOs):-

PEO1: To exhibit analytical skills in modeling and solving computing problems by applying mathematical, scientific and engineering knowledge and to pursue their higher

PEO2: To communicate effectively with multi-disciplinary teams to develop quality software systems with an orientation towards research and development for lifelong learning.

PEO3: To address industry and societal needs for the growth of global economy using emerging technologies by following professional ethics.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

Students will be able to

PSO1: Organize, maintain and protect IT Infrastructural resources.

PSO2: Design and Develop web, mobile, and smart apps based software solutions to the real world problems.

Course Objectives:

- To demonstrate the basic concepts of MapReduce, Hadoop and its ecosystem.
- To introduce design and demonstration of Hadoop ecosystem components.

Course Outcomes:

Upon successful completion of the course, the students will be able to

CO1: choose suitable LINUX commands to work in Hadoop environment.

CO2: use HDFS file structure and MapReduce framework to solve complex problems.

CO3: analyze data using Pig and Hive.

Mapping of course outcomes with program outcomes

BIGDATA ANALYTICS	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2
CO1: choose suitable LINUX commands to work in Hadoop environment	3	3	2	3	3			2	2	2		3	3	3
CO2: use HDFS file structure and MapReduce framework to solve complex problems	3	3	2	3	3			2	2	2		3	3	3
CO3: analyze data using Pig and Hive.	3	3	2	3	3			2	2	2		3	3	3

Index

BIG DATA ANALYTICS

EXPERIMENT_NO-01

Aim: Practice on basic Linux commands.

Program:

1. pwd Command

The pwd command is used to display the location of the current working directory.

Input: ~$ pwd

Output: /home/user

2. mkdir Command

The mkdir command is used to create a new directory under any directory.

Input: ~$ mkdir gec it cse

Output: ~$

3. cd Command

The cd command is used to change the current directory.

Input: ~$ cd it

Output: ~/it$

4. cat Command

The cat command is a multi-purpose utility in the Linux system. It can be used to create a file, display content of the file, copy the content of one file to another file, and more.

Input: ~/it$ cat > languages.txt

java

python

c++

c

.net

html

css

reactjs

java script

angular js

devops

express js

mongodb

sql

mysql

oracle

c#

node js

rubi

linux

Output: ~/it$ cat languages.txt

java

python

c++

c

.net

html

css

reactjs

java script

angular js

devops

express js

mongodb

sql

mysql

oracle

c#

node js

rubi

linux

5. rmdir Command

The rmdir command is used to delete a directory.

Input: ~$ rmdir gec

Output: ~$ cat gec

cat: gec: No such file or directory

6. ls Command

The ls command is used to display a list of content of a directory.

Input: ~/it$ ls

Output: dharshi.txt languages.txt sathwika.txt

7. cp Command

The cp command is used to copy a file or directory.

Input: ~/it$ cp sathwika.txt dharshi.txt

Output: ~$

8. head Command

The head command is used to display the content of a file. It displays the first 10 lines of a file.

Input: ~/it$ head languages.txt

Output:

java

python

c++

c

.net

html

css

reactjs

java script

angular js

9. tail Command

The tail command is similar to the head command. The difference between both commands is that it displays the last ten lines of the file content.

Input: ~/it$ tail languages.txt

Output:

devops

express js

mongodb

sql

mysql

oracle

c#

node js

rubi

linux

10. tac Command

The tac command is the reverse of cat command, as its name specified. It displays the file content in reverse order (from the last line).

Input: ~/it$ tac languages.txt

Output:

linux

rubi

node js

c#

oracle

mysql

sql

mongodb

express js

devops

angular js

java script

reactjs

css

html

.net

c

c++

python

java

11. grep Command

The grep is the most powerful and used filter in a Linux system. The 'grep' stands for "**global regular expression print**." It is useful for searching the content from a file. Generally, it is used with the pipe.

Input: ~/it$ cat languages.txt | grep +

Output: c++

12. tr Command

The tr command is used to translate the file content like from lower case to upper case.

Input: ~/it$ cat languages.txt | tr 'js' 'JS'

Output:

Java

python

c++

c

.net

html

cSS

reactJS

Java Script

angular JS

devopS

expreSS JS

mongodb

Sql

mySql

oracle

c#

node JS

rubi

linux

13. sort Command

The sort command is used to sort files in alphabetical order.

Input : ~/it$ sort languages.txt

Output:

mongodb

.net

angular js

c

c#

c++

css

devops

express js

html

java

java script

linux

mysql

node js

oracle

python

reactjs

rubi

sql

14. sleep Command

The sleep command is used to hold the terminal by the specified amount of time. By default, it takes time in seconds.

Input: ~/it$ sleep 5

Output: ~/it$

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-02

AIM: Implement the following file management tasks in Hadoop:

i. Adding files and Directories

ii. Retrieving files

iii. Deleting files

iv. Copying files from local filesystem to HDFS and vice versa.

v. Moving files

Description:

1) Install Oracle VM VirtualBox

2) Install Cloudera

3) Open Oracle VM VirtualBox->Export->Browse the cloudera

4) Click on start

PROGRAM:

i) Adding files and Directories

-mkdir: Creating a directory

```
[cloudera@quickstart ~]$ hadoop fs -mkdir it
[cloudera@quickstart ~]$
```

-put: Upload a file in HDFS

```
[cloudera@quickstart ~]$ cat > courses.txt
java
python
bda
c
c++
^Z
[1]+  Stopped                 cat > courses.txt
[cloudera@quickstart ~]$ hadoop fs -put courses.txt
[cloudera@quickstart ~]$
```

-touchz: creates a zero byte file. This is similar to the touch command in unix.

```
[cloudera@quickstart ~]$ hadoop fs -touchz gec.txt
[cloudera@quickstart ~]$
```

ii. Retrieving files

-ls: List the files in hadoop

```
[cloudera@quickstart ~]$ hadoop fs -ls
Found 7 items
-rw-r--r--   1 cloudera cloudera         46 2023-08-24 07:55 WCFile.txt
-rw-r--r--   1 cloudera cloudera         43 2023-08-24 08:10 WCFile1.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 07:57 WCWord
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 08:13 WCWord1
-rw-r--r--   1 cloudera cloudera         22 2023-08-25 04:51 courses.txt
-rw-r--r--   1 cloudera cloudera          0 2023-08-25 05:05 gec.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-25 04:47 it
```

-get: Copies files to the local file system

```
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    enterprise-deployment.json  Music      Templates
cm_api.py         Documents  express-deployment.json     parcels    Videos
courses.txt       Downloads  kerberos                    Pictures   workspace
dept.txt          eclipse    lib                         Public
[cloudera@quickstart ~]$ hadoop fs -get gec.txt
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    enterprise-deployment.json  lib        Public
cm_api.py         Documents  express-deployment.json     Music      Templates
courses.txt       Downloads  gec.txt                     parcels    Videos
dept.txt          eclipse    kerberos                    Pictures   workspace
[cloudera@quickstart ~]$
```

-cat: Displays the content in the file.

```
[cloudera@quickstart ~]$ hadoop fs -cat courses.txt
java
python
bda
c
c++
```

-head: Displays the first 10 lines in the file.

```
[cloudera@quickstart ~]$ hadoop fs -cat no.txt | head
1
2
3
4
5
6
7
8
9
10
```

-tail: Displays the last 10 lines in the file.

```
[cloudera@quickstart ~]$ hadoop fs -cat no.txt | tail
3
4
5
6
7
8
9
10
11
12
```

iii. Deleting files

-rmdir: removes a directory

```
[cloudera@quickstart ~]$ hadoop fs -rmdir it
[cloudera@quickstart ~]$ hadoop fs -ls
Found 6 items
-rw-r--r--   1 cloudera cloudera         46 2023-08-24 07:55 WCFile.txt
-rw-r--r--   1 cloudera cloudera         43 2023-08-24 08:10 WCFile1.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 07:57 WCWord
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 08:13 WCWord1
-rw-r--r--   1 cloudera cloudera         22 2023-08-25 04:51 courses.txt
-rw-r--r--   1 cloudera cloudera          0 2023-08-25 05:05 gec.txt
[cloudera@quickstart ~]$
```

-rm: removes a file

```
[cloudera@quickstart ~]$ hadoop fs -rm gec.txt
Deleted gec.txt
[cloudera@quickstart ~]$ hadoop fs -ls
Found 5 items
-rw-r--r--   1 cloudera cloudera         46 2023-08-24 07:55 WCFile.txt
-rw-r--r--   1 cloudera cloudera         43 2023-08-24 08:10 WCFile1.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 07:57 WCWord
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 08:13 WCWord1
-rw-r--r--   1 cloudera cloudera         22 2023-08-25 04:51 courses.txt
[cloudera@quickstart ~]$
```

iv. Copying files from local filesystem to HDFS and vice versa.

-copyFromLocal: Copies files from the local file system

```
[cloudera@quickstart ~]$ cat > fruits.txt
orange
apple
grapes
^Z
[4]+  Stopped                 cat > fruits.txt
[cloudera@quickstart ~]$ hadoop fs -copyFromLocal fruits.txt
[cloudera@quickstart ~]$ hadoop fs -ls
Found 6 items
-rw-r--r--   1 cloudera cloudera         46 2023-08-24 07:55 WCFile.txt
-rw-r--r--   1 cloudera cloudera         43 2023-08-24 08:10 WCFile1.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 07:57 WCWord
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 08:13 WCWord1
-rw-r--r--   1 cloudera cloudera         22 2023-08-25 04:51 courses.txt
-rw-r--r--   1 cloudera cloudera         20 2023-08-25 05:18 fruits.txt
[cloudera@quickstart ~]$
```

-copyToLocal: Copies files to the local file system

```
[cloudera@quickstart ~]$ hadoop fs -touchz file.txt
[cloudera@quickstart ~]$ hadoop fs -copyToLocal file.txt
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    enterprise-deployment.json  gec.txt   parcels    Videos
cm_api.py         Documents  express-deployment.json     kerberos  Pictures   workspace
courses.txt       Downloads  file.txt                    lib       Public
dept.txt          eclipse    fruits.txt                  Music     Templates
[cloudera@quickstart ~]$
```

v. Moving files

-moveFromLocal: files from local file system to the Hadoop distributed file system

```
[cloudera@quickstart ~]$ cat > games.txt
chess
carroms
ludo
^Z
[5]+  Stopped                 cat > games.txt
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    enterprise-deployment.json  games.txt  Music     Templates
cm_api.py         Documents  express-deployment.json     gec.txt    parcels   Videos
courses.txt       Downloads  file.txt                    kerberos   Pictures  workspace
dept.txt          eclipse    fruits.txt                  lib        Public
[cloudera@quickstart ~]$ hadoop fs -moveFromLocal games.txt
[cloudera@quickstart ~]$ hadoop fs -ls
Found 8 items
-rw-r--r--   1 cloudera cloudera         46 2023-08-24 07:55 WCFile.txt
-rw-r--r--   1 cloudera cloudera         43 2023-08-24 08:10 WCFile1.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 07:57 WCWord
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 08:13 WCWord1
-rw-r--r--   1 cloudera cloudera         22 2023-08-25 04:51 courses.txt
-rw-r--r--   1 cloudera cloudera          0 2023-08-25 05:22 file.txt
-rw-r--r--   1 cloudera cloudera         20 2023-08-25 05:18 fruits.txt
-rw-r--r--   1 cloudera cloudera         19 2023-08-25 05:26 games.txt
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    enterprise-deployment.json  gec.txt   parcels    Videos
cm_api.py         Documents  express-deployment.json     kerberos  Pictures   workspace
courses.txt       Downloads  file.txt                    lib       Public
dept.txt          eclipse    fruits.txt                  Music     Templates
[cloudera@quickstart ~]$
```

-mv: Move file from source to destination.

```
[cloudera@quickstart ~]$ hadoop fs -ls
Found 9 items
-rw-r--r--   1 cloudera cloudera         46 2023-08-24 07:55 WCFile.txt
-rw-r--r--   1 cloudera cloudera         43 2023-08-24 08:10 WCFile1.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 07:57 WCWord
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 08:13 WCWord1
drwxr-xr-x   - cloudera cloudera          0 2023-08-25 06:06 bda
-rw-r--r--   1 cloudera cloudera         22 2023-08-25 04:51 courses.txt
-rw-r--r--   1 cloudera cloudera          0 2023-08-25 05:22 file.txt
-rw-r--r--   1 cloudera cloudera         20 2023-08-25 05:18 fruits.txt
-rw-r--r--   1 cloudera cloudera         19 2023-08-25 05:26 games.txt
[cloudera@quickstart ~]$ hadoop fs -mv fruits.txt bda
[cloudera@quickstart ~]$ hadoop fs -ls
Found 8 items
-rw-r--r--   1 cloudera cloudera         46 2023-08-24 07:55 WCFile.txt
-rw-r--r--   1 cloudera cloudera         43 2023-08-24 08:10 WCFile1.txt
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 07:57 WCWord
drwxr-xr-x   - cloudera cloudera          0 2023-08-24 08:13 WCWord1
drwxr-xr-x   - cloudera cloudera          0 2023-08-25 06:11 bda
-rw-r--r--   1 cloudera cloudera         22 2023-08-25 04:51 courses.txt
-rw-r--r--   1 cloudera cloudera          0 2023-08-25 05:22 file.txt
-rw-r--r--   1 cloudera cloudera         19 2023-08-25 05:26 games.txt
[cloudera@quickstart ~]$
```

du: Shows disk usage, in bytes, for all the files which match path; filenames are reported with the full HDFS protocol prefix.

```
[cloudera@quickstart ~]$ hadoop fs -du /user/cloudera
46  46  /user/cloudera/WCFile.txt
43  43  /user/cloudera/WCFile1.txt
25  25  /user/cloudera/WCWord
46  46  /user/cloudera/WCWord1
20  20  /user/cloudera/bda
22  22  /user/cloudera/courses.txt
0   0   /user/cloudera/file.txt
19  19  /user/cloudera/games.txt
27  27  /user/cloudera/no.txt
[cloudera@quickstart ~]$
```

getmerge: concatenates the files in the source directory into the destination file.

```
[cloudera@quickstart ~]$ hadoop fs -getmerge courses.txt no.txt mergfile
[cloudera@quickstart ~]$
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-03

AIM: Write driver code, mapper code, reducer code to count number of words in a given file. (Hint: WordCount Map- Reduce Program)

Description:

1) Open Oracle VM VirtualBox->export cloudera->start

2) Cloudera->settings->system->set processors to “2”, by default it is “1”.

3) To launch “cloudera Express”

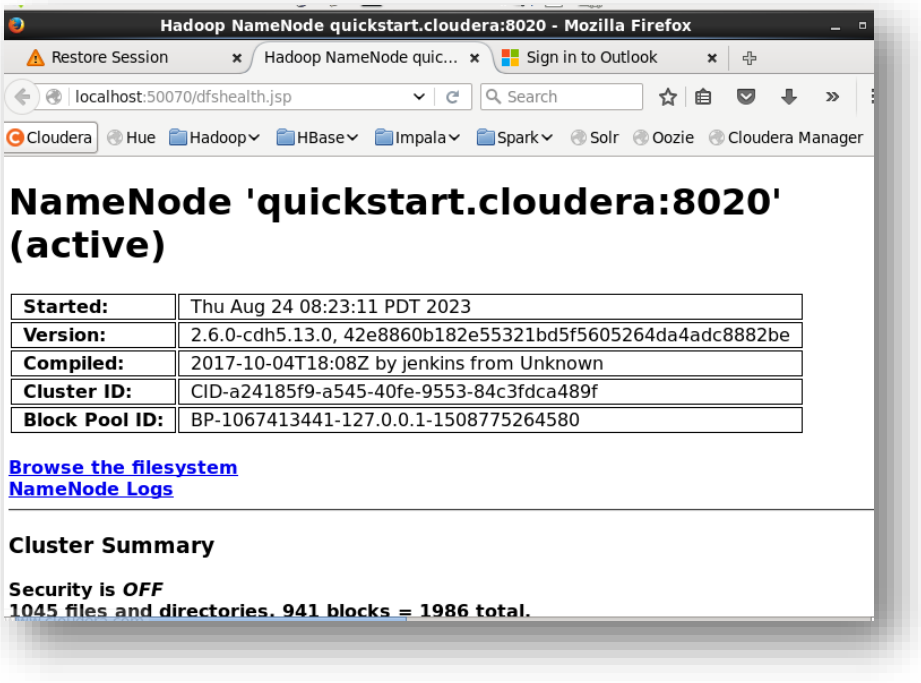
(i) Open terminal in cloudera and start the server by using

“sudo service cloudera-sdh-server start”

(ii) After successful completion click on Cloudera Express->Cloud Manager

(iii) Both username and password is “cloudera” and then click on Login

4) In browser type “localhost:50070/dfshealth.jsp”



5) In eclipse->File->New->Java project->Project name “WordCount”->Finish

6) Create three classes.

Right click on WordCount -> New->class->Name “WCDriver”

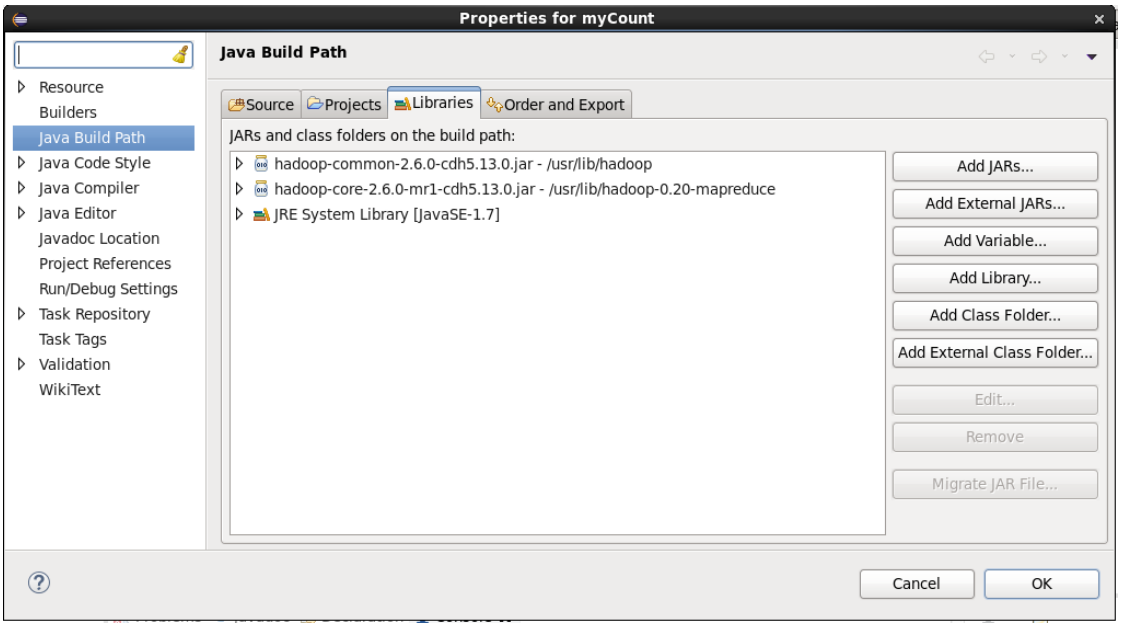
Right click on WordCount -> New->class->Name “WCMapper”

Right click on WordCount -> New->class->Name “WCReducer”

7) Add Hadoop libraries.

Right click on WordCount ->Build path->Configure Build path->Add external JARS. (usr\lib\hadoop\hadoop-common-2.6.0-cdh 5.13.0 jar,

Usr\lib\hadoop\hadoop-core-2.6.0-cdh 5.13.0 jar)



PROGRAM:

WCMapper.java

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
```

```
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper<LongWritable,
                                                    Text, Text, IntWritable> {

    // Map function
    public void map(LongWritable key, Text value, OutputCollector<Text,
                 IntWritable> output, Reporter rep) throws IOException
    {

        String line = value.toString();

        // Splitting the line on spaces
        for (String word : line.split(" "))
        {
            if (word.length() > 0)
            {
                output.collect(new Text(word), new IntWritable(1));
            }
        }
    }
}
```

WCReducer.java

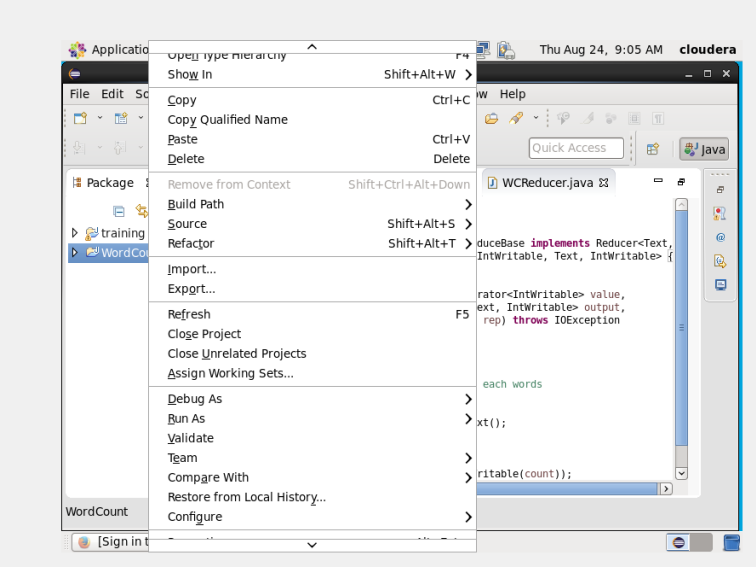
```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer<Text,
                                    IntWritable, Text, IntWritable> {

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> value,
                   OutputCollector<Text, IntWritable> output,
                            Reporter rep) throws IOException
    {

        int count = 0;
```

```
        // Counting the frequency of each words
        while (value.hasNext())
        {
            IntWritable i = value.next();
            count += i.get();
        }

        output.collect(key, new IntWritable(count));
    }
}
```

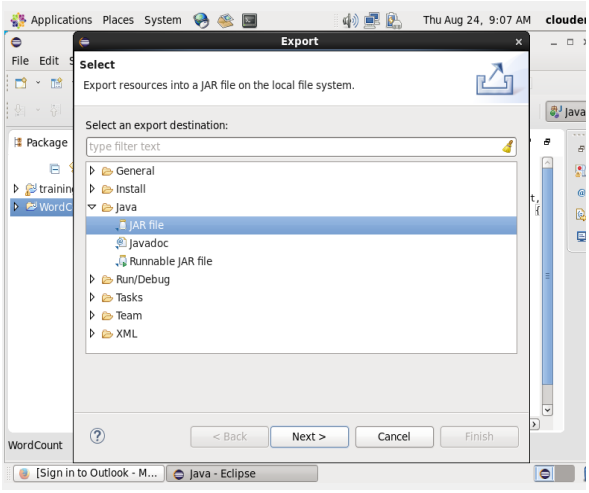
WCDriver.java

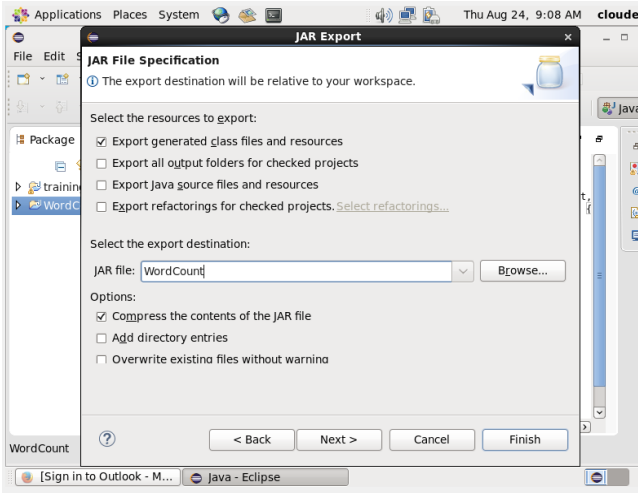
```
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

    public int run(String args[]) throws IOException
    {
        if (args.length < 2)
        {
            System.out.println("Please give valid inputs");
            return -1;
        }

        JobConf conf = new JobConf(WCDriver.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
```

```
        JobClient.runJob(conf);
        return 0;
    }

    // Main Method
    public static void main(String args[]) throws Exception
    {
        int exitCode = ToolRunner.run(new WCDriver(), args);
        System.out.println(exitCode);
    }
}
```



Right click on WordCount->Export->java->jar file->JAR file:" WordCount"->Finish





OUTPUT:

In Terminal

```
[cloudera@quickstart ~]$ cd workspace
[cloudera@quickstart workspace]$ cat > WCFile.txt
hello cse
hello it
hello gec cse
hello gec it
^Z
[1]+  Stopped                 cat > WCFile.txt
```

```
[cloudera@quickstart workspace]$ hadoop fs -put WCFile.txt WCFile.txt
```

```
[cloudera@quickstart workspace]$ hadoop jar WordCount.jar WCDriver
WCFile.txt WCWord
```

```
[cloudera@quickstart workspace]$ hadoop fs -cat WCWord/part-00000
cse     2
gec     2
hello   4
it      2
```

BIG DATA ANALYTICS LAB

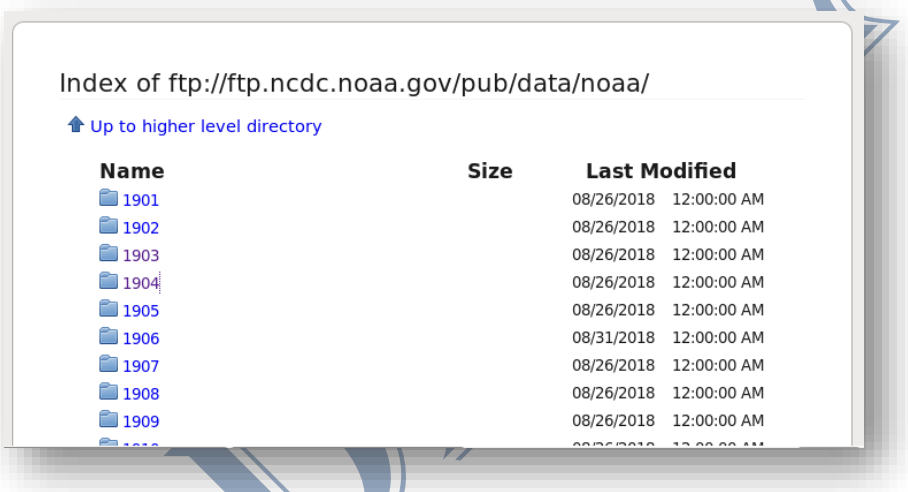
EXPERIMENT_NO-04

AIM: Write a Map Reduce program that mines weather data. Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with MapReduce, since it is semi structured and record-oriented.

Description:

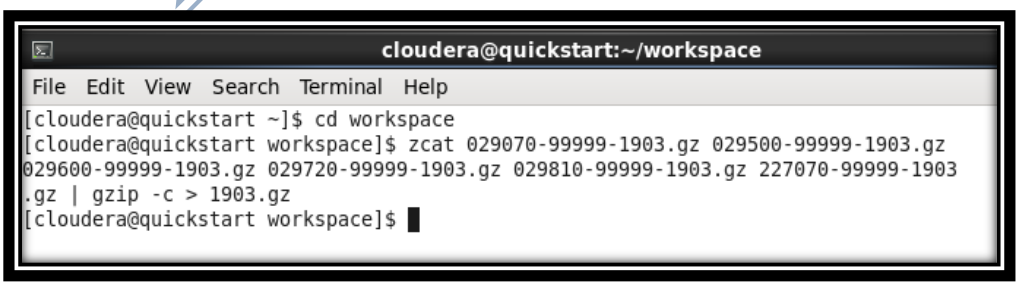
1) Open Oracle VM VirtualBox->export cloudera->start

2) Open browser and type "**ftp://ftp.ncdc.noaa.gov/pub/data/noaa/**".



3) Download any 3 folders to workspace.

4) There are multiple files in a folder, concatenate those files into a folder as follows.



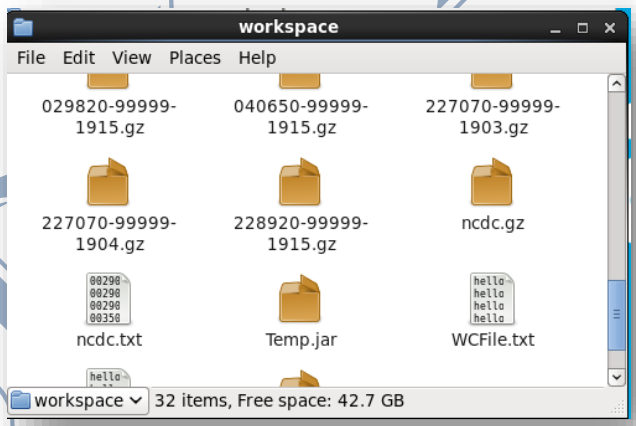
```
[cloudera@quickstart workspace]$ zcat 029070-99999-1904.gz 029500-99999-1904.gz 029600-99999-1904
.gz 029720-99999-1904.gz 029810-99999-1904.gz 227070-99999-1904.gz | gzip -c > 1904.gz
[cloudera@quickstart workspace]$
```

```
[cloudera@quickstart workspace]$ zcat 028060-99999-1915.gz 028690-99999
-1915.gz 028750-99999-1915.gz 029170-99999-1915.gz 029440-99999-1915.gz
  029820-99999-1915.gz 040650-99999-1915.gz 228920-99999-1915.gz | gzip
 -c > 1915.gz
[cloudera@quickstart workspace]$
```

5) Concatenate these 3 folders into a single folder as follows.

```
[cloudera@quickstart workspace]$ zcat 1903.gz 1904.gz 1915.gz | gzip -c > ncdc.gz
[cloudera@quickstart workspace]$
```

6) Right click on ncdc.gz ->Extract here->rename->ncdc.txt



7) In eclipse->File->New->Java project->Project name “Weather”->Finish

8) Create three classes.

Right click on Weather-> New->class->Name “MaxTemperature”

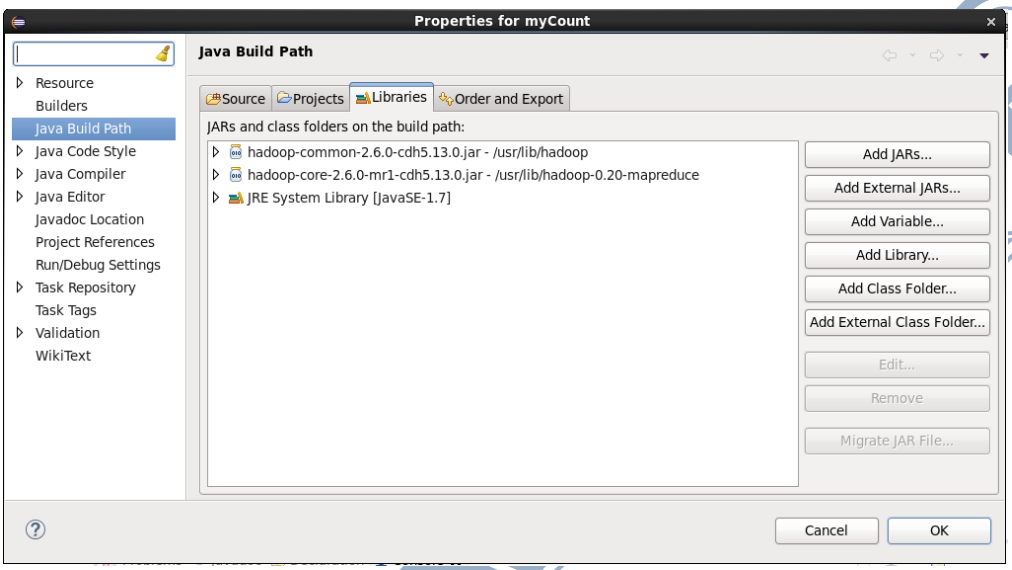
Right click on Weather-> New->class->Name “MaxTemperatureMapper”

Right click on Weather -> New->class->Name “MaxTemperatureReducer”

9) Add Hadoop libraries.

Right click on Weather->Build path->Configure Build path->Add external JARS. (usr\lib\hadoop\hadoop-common-2.6.0-cdh 5.13.0 jar,

usr\lib\hadoop\hadoop-core-2.6.0-cdh 5.13.0 jar)



PROGRAM:

MaxTemperature.java

```
import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MaxTemperature {

public static void main(String[] args)throws Exception{
```

```
    if(args.length!=2)
    {
       System.err.println("error");
       System.exit(-1);
    }
    Job job=new Job();
    job.setJarByClass(MaxTemperature.class);
    job.setJobName("Max Temperature");
    FileInputFormat.addInputPath(job,new Path(args[0]));
    FileOutputFormat.setOutputPath(job,new Path(args[1]));
    job.setMapperClass(MaxTemperatureMapper.class);
    job.setReducerClass(MaxTemperatureReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    System.exit(job.waitForCompletion(true)?0:1);
}
}
```

MaxTemperatureMapper.java

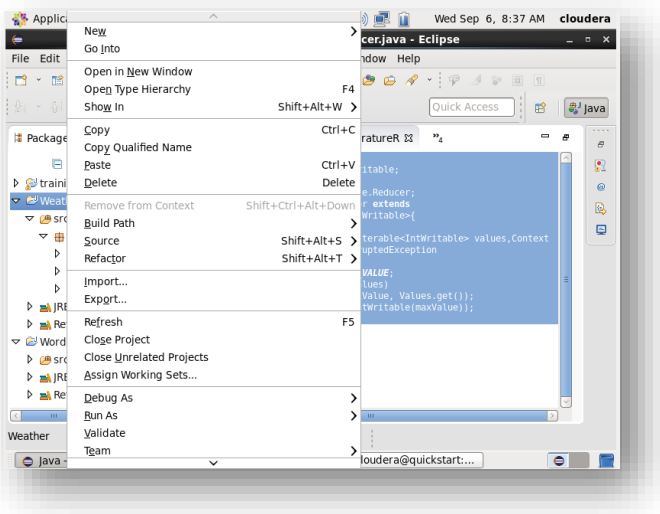
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
```

```
public class MaxTemperatureMapper extends
Mapper<LongWritable,Text,Text,IntWritable>
{
   private static final int MISSING=9999;
   @Override
   public void map(LongWritable key,Text value,Context context)
throws IOException,InterruptedException{
       String line=value.toString();
       String year=line.substring(15,19);
       int airTemperature;
       if(line.charAt(87)=='+')
          airTemperature=Integer.parseInt(line.substring(88,92));
       else
          airTemperature=Integer.parseInt(line.substring(87,92));
       String quality=line.substring(92,93);
       if(airTemperature!=MISSING && quality.matches("[01459]"))
          context.write(new Text(year),new IntWritable(airTemperature));
   }
}
```

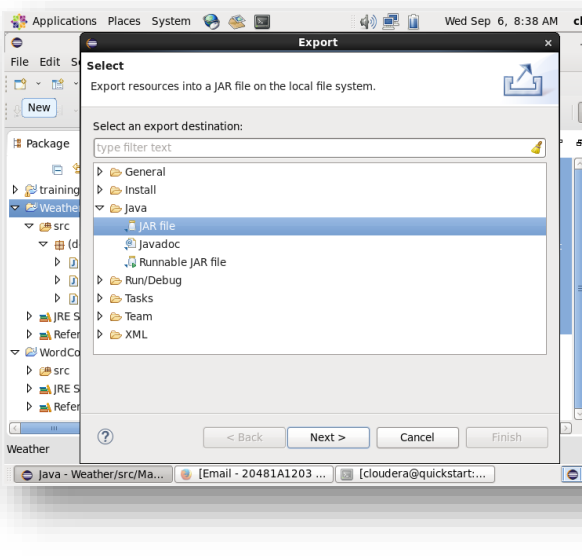
MaxTemperatureReducer.java

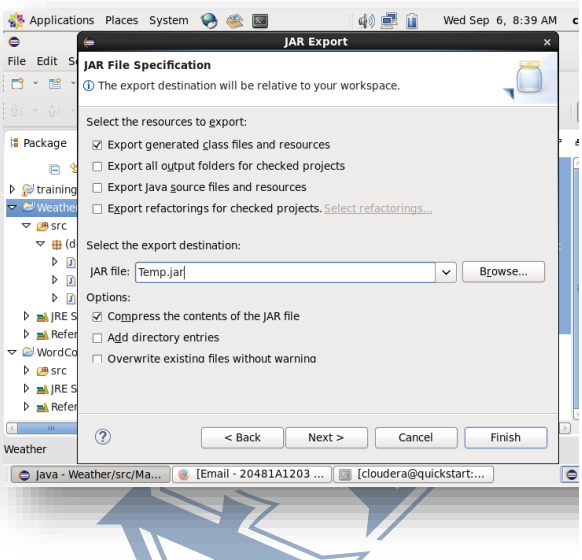
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Reducer;
public class MaxTemperatureReducer extends
Reducer<Text,IntWritable,Text,IntWritable>{
   @Override
   public void reduce(Text key,Iterable<IntWritable> values,Context
context)throws IOException,InterruptedException
   {
      int maxValue=Integer.MIN_VALUE;
      for(IntWritable Values:values)
         maxValue=Math.max(maxValue, Values.get());
      context.write(key, new IntWritable(maxValue));
   }
}
```



Right click on Weather->Export->java->jar file->JAR file:" **Temp**"->Finish





OUTPUT:

In Terminal

```
[cloudera@quickstart workspace]$ hadoop fs -put ncdc.txt ncdc.txt
[cloudera@quickstart workspace]$
```

```
[cloudera@quickstart workspace]$ hadoop jar Temp.jar MaxTemperature ncdc.txt out
```

```
[cloudera@quickstart workspace]$ hadoop fs -cat out/part-r-00000
1903    289
1904    256
1915    294
[cloudera@quickstart workspace]$
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-05

AIM: Implement Matrix Multiplication with Hadoop Map Reduce.

Description:

1) Open Oracle VM VirtualBox->export cloudera->start

2) In eclipse->File->New->Java project->Project name "MatrixMultiplication"
->Finish

3) Create three classes.

Right click on MatrixMultiplication-> New->class->Name "MatrixDriver"

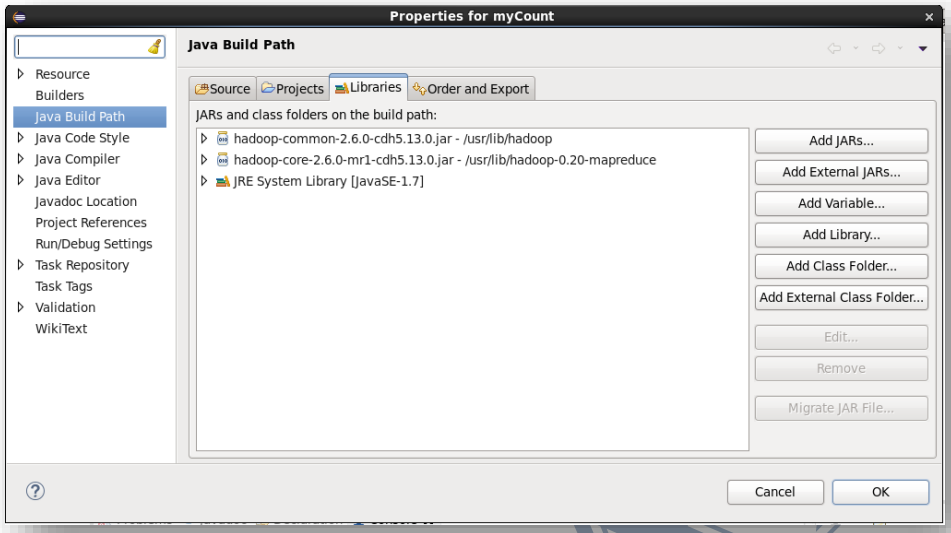
Right click on MatrixMultiplication-> New->class->Name "MatrixMapper"

Right click on MatrixMultiplication -> New->class->Name "MatrixReducer"

4) Add Hadoop libraries.

Right click on MatrixMultiplication->Build path->Configure Build path->Add external JARS. (usr\lib\hadoop\hadoop-common-2.6.0-cdh 5.13.0 jar,

usr\lib\hadoop\hadoop-core-2.6.0-cdh 5.13.0 jar)



PROGRAM:

MatrixDriver.java

```
import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapreduce.*;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class MatrixDriver

{

    public static void main(String[] args) throws Exception

    {
```

```
        Configuration conf = new Configuration();
        // M is an m-by-n matrix; N is an n-by-p matrix.
        conf.set("m", "2");
        conf.set("n", "2");
        conf.set("p", "2");
        Job job = Job.getInstance(conf, "MatrixMultiplication");
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        job.setJarByClass(MatrixDriver.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);
        job.setMapperClass(MatrixMapper.class);
        job.setReducerClass(MatrixReducer.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        System.exit(job.waitForCompletion(true)?0:1);

}
}
```

MatrixMapper.java

```
import java.io.IOException;

import org.apache.hadoop.conf.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapreduce.*;

public class MatrixMapper extends Mapper <LongWritable, Text, Text, Text>

{

public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException

{

Configuration conf = context.getConfiguration();

int m = Integer.parseInt(conf.get("m"));

int p = Integer.parseInt(conf.get("p"));

String line = value.toString();

String[] indicesAndValue = line.split(",");

Text outputKey = new Text();

Text outputValue = new Text();

if (indicesAndValue[0].equals("M"))

{

    for (int k = 0; k < p; k++)

{

outputKey.set(indicesAndValue[1] + "," + k);

outputValue.set("M," + indicesAndValue[2] + "," + indicesAndValue[3]);
```

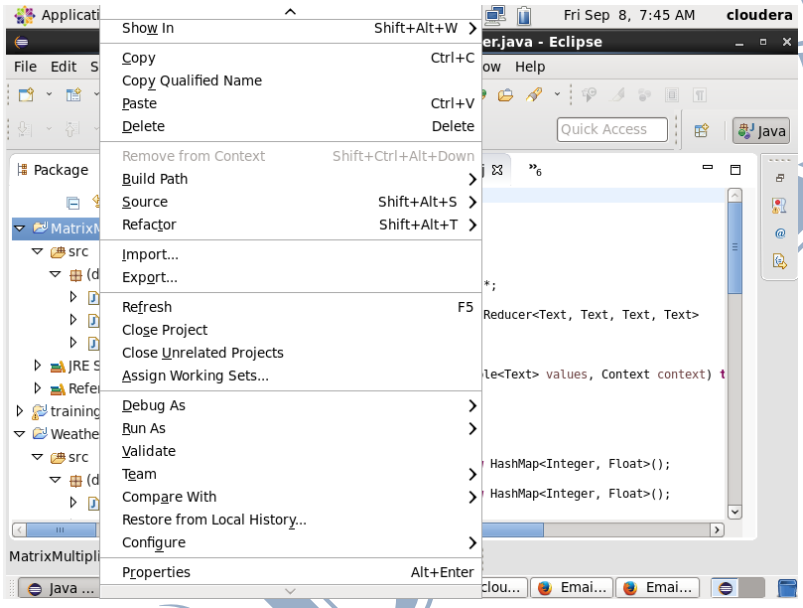
```
context.write(outputKey, outputValue);

}

}

else

{

for (int i = 0; i < m; i++)

{

outputKey.set(i + "," + indicesAndValue[2]);

outputValue.set("N," + indicesAndValue[1] + "," + indicesAndValue[3]);

context.write(outputKey, outputValue);

}

}

}

}
```

MatrixReducer.java

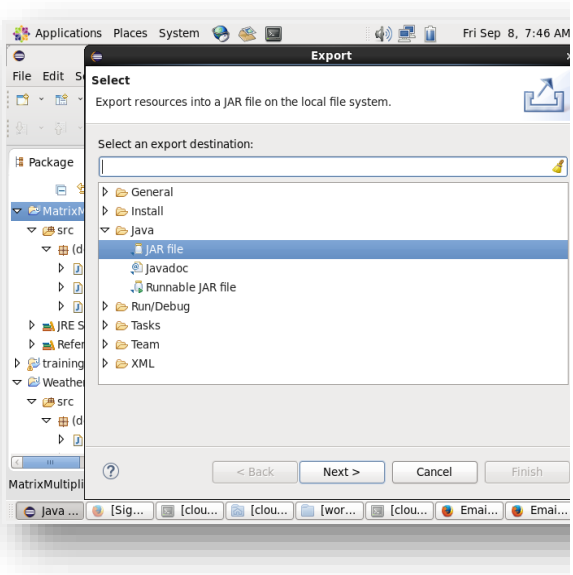
```
import java.io.IOException;

import java.util.*;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapreduce.*;

public class MatrixReducer extends Reducer<Text, Text, Text, Text>

{

public void reduce(Text key, Iterable<Text> values, Context
context) throws IOException, InterruptedException
```

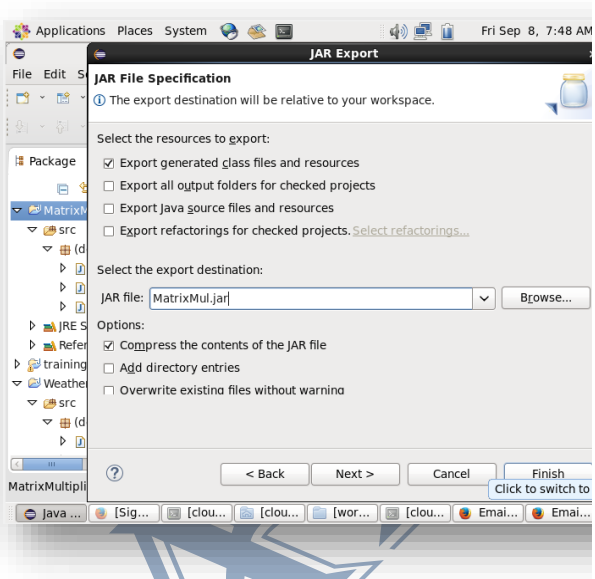
```
{
String[] value;

HashMap<Integer, Float> hashA = new HashMap<Integer, Float>();

HashMap<Integer, Float> hashB = new HashMap<Integer, Float>();

for (Text val : values)

{

value = val.toString().split(",");

if (value[0].equals("M"))

{

hashA.put(Integer.parseInt(value[1]), Float.parseFloat(value[2]));

}

else

{

hashB.put(Integer.parseInt(value[1]), Float.parseFloat(value[2]));

}

}

int n = Integer.parseInt(context.getConfiguration().get("n")); float result
= 0.0f;

float a_ij; float b_jk;

for (int j = 0; j < n; j++)

{

a_ij = hashA.containsKey(j) ? hashA.get(j) : 0.0f;

b_jk = hashB.containsKey(j) ? hashB.get(j) : 0.0f;

result += a_ij * b_jk;
```

```
}
if (result != 0.0f)
{
context.write(null, new Text(key.toString() + "," + Float.toString(result)));
}
}
}
```



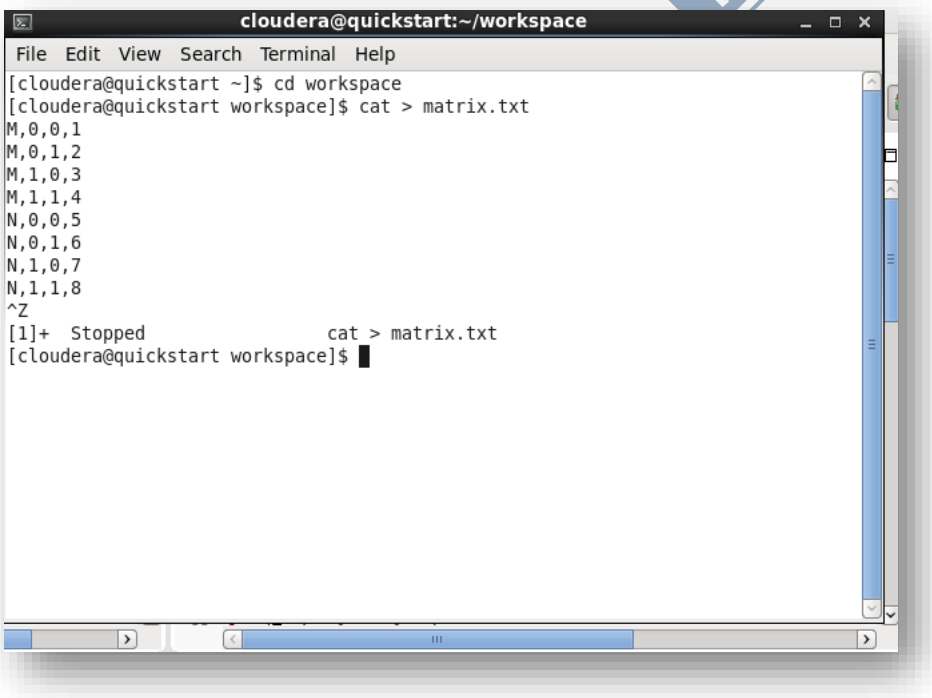
Right click on MatrixMultiplication->Export->java->jar file->JAR file:"**MatrixMul**"->Finish





OUTPUT:

In Terminal



```
[cloudera@quickstart workspace]$ hadoop fs -put matrix.txt
[cloudera@quickstart workspace]$
```

```
[cloudera@quickstart workspace]$ hadoop jar MatrixMul.jar MatrixDriver matrix.txt MatrixOutput
```

```
[cloudera@quickstart workspace]$ hadoop fs -cat MatrixOutput/part-r-00000
0,0,19.0
0,1,22.0
1,0,43.0
1,1,50.0
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-06

AIM: Install Pig and write Pig latin scripts to Load, Store and Filter data.

PROGRAM:

Installation procedure:

- Download and extract pig-0.13.0.
  Command: wget https://archive.apache.org/dist/pig/pig-0.13.0/pig-0.13.0.tar.gz
  Command: tar xvf pig-0.13.0.tar.gz
  Command: sudo mv pig-0.13.0 /usr/lib/pig

- Set Path for pig
  Command: sudo gedit
  $HOME/.bashrc
  export
  PIG_HOME=/usr/lib/
  pig
  export PATH=$PATH:$PIG_HOME/bin
  export PIG_CLASSPATH=$HADOOP_COMMON_HOME/conf

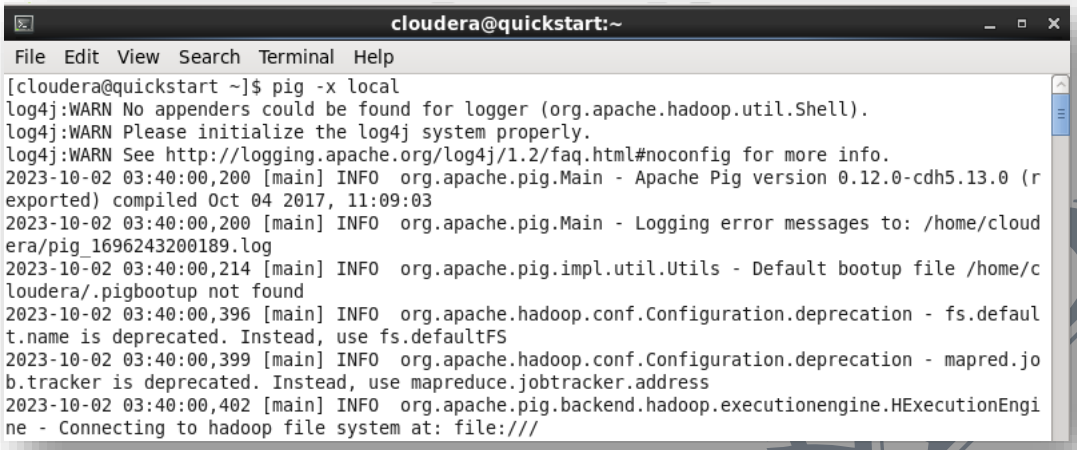
- **pig.properties file**
  In the conf folder of Pig, we have a file named pig.properties. In the pig.properties file, you can set various parameters as given below.
  pig -h properties

- **Verifying the Installation**
  Verify the installation of Apache Pig by typing the version command. If the installation is successful, you will get the version of Apache Pig as shown below.
  Command: pig -version

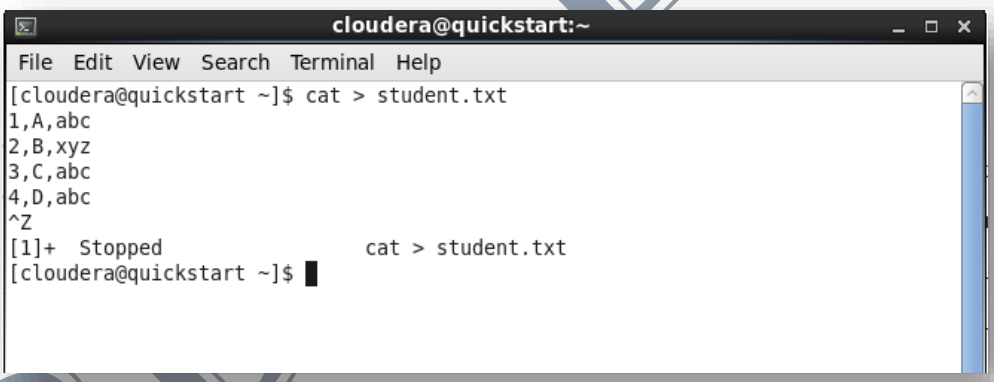
```
pcetcse@pcetcse1: ~
pcetcse@pcetcse1:~$ pig -version
Apache Pig version 0.13.0 (r1606446)
compiled Jun 29 2014, 02:29:34
pcetcse@pcetcse1:~$
```

1) Local mode:

To enter into local mode use **pig -x local**



(i) Create a file:



(ii) Load:

```
grunt> st = Load 'student.txt' using PigStorage(',') as (no:int,name:chararray,address:chararray)
;
```

```
grunt> dump st;
2023-10-02 03:42:55,016 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: UNKNOWN
2023-10-02 03:42:55,063 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimize
r - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstPar
allelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEa
ch, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCas
tInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-02 03:42:55,150 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-10-02 03:42:55,170 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MultiQueryOptimizer - MR plan size before optimization: 1
2023-10-02 03:42:55,170 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MultiQueryOptimizer - MR plan size after optimization: 1
2023-10-02 03:42:55,187 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - session.i
d is deprecated. Instead, use dfs.metrics.session-id
2023-10-02 03:42:55,190 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Initializing JVM
Metrics with processName=JobTracker, sessionId=
2023-10-02 03:42:55,208 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig script setti
ngs are added to the job
2023-10-02 03:42:55,266 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.jo
b.reduce.markreset.buffer.percent is deprecated. Instead, use mapreduce.reduce.markreset.buffer.p
ercent
2023-10-02 03:42:55,266 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.JobControlCompiler - mapred.job.reduce.markreset.buffer.percent is not set, set to default 0.3
2023-10-02 03:42:55,266 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.ou
tput.compress is deprecated. Instead, use mapreduce.output.fileoutputformat.compress
2023-10-02 03:42:55,299 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
```

Output:

```
(1,A,abc)
(2,B,xyz)
(3,C,abc)
(4,D,abc)
```

(iii) Filter:

```
grunt> r = Filter st by address=='abc';
```

```
grunt> dump r;
2023-10-02 03:44:18,891 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: FILTER
2023-10-02 03:44:18,892 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimize
r - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstPar
allelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEa
ch, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCas
tInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-02 03:44:18,904 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-10-02 03:44:18,906 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MultiQueryOptimizer - MR plan size before optimization: 1
2023-10-02 03:44:18,906 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
```

Output:

```
(1,A,abc)
(3,C,abc)
(4,D,abc)
```

2) Mapreduce mode:

To enter into a mapreduce mode use **pig**

```
cloudera@quickstart:~
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart ~]$ hadoop fs -put student.txt
[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ pig
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
2023-10-02 04:13:58,112 [main] INFO  org.apache.pig.Main - Apache Pig version 0.12.0-cdh5.13.0 (r
exported) compiled Oct 04 2017, 11:09:03
2023-10-02 04:13:58,112 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/cloud
era/pig_1696245238098.log
2023-10-02 04:13:58,122 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/c
loudera/.pigbootup not found
2023-10-02 04:13:58,669 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.jo
b.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2023-10-02 04:13:58,669 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.defaul
t.name is deprecated. Instead, use fs.defaultFS
2023-10-02 04:13:58,669 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngi
ne - Connecting to hadoop file system at: hdfs://quickstart.cloudera:8020
```

(i) Load:

```
grunt> st = Load 'hdfs://quickstart.cloudera:8020/user/cloudera/student.txt' using PigStorage(',')
) as (c1:Int,c2:chararray,c3:chararray);
grunt>
```

```
grunt> dump st;
2023-10-02 04:18:43,195 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2023-10-02 04:18:43,227 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-02 04:18:43,329 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-10-02 04:18:43,353 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size before optimization: 1
2023-10-02 04:18:43,353 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size after optimization: 1
2023-10-02 04:18:43,512 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
```

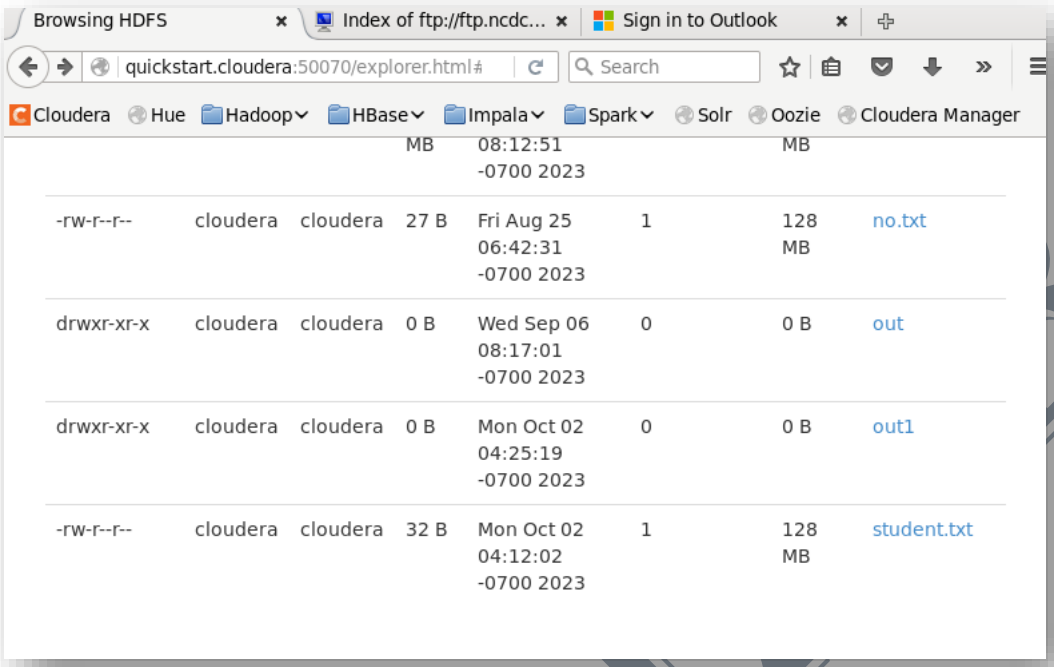
Output:

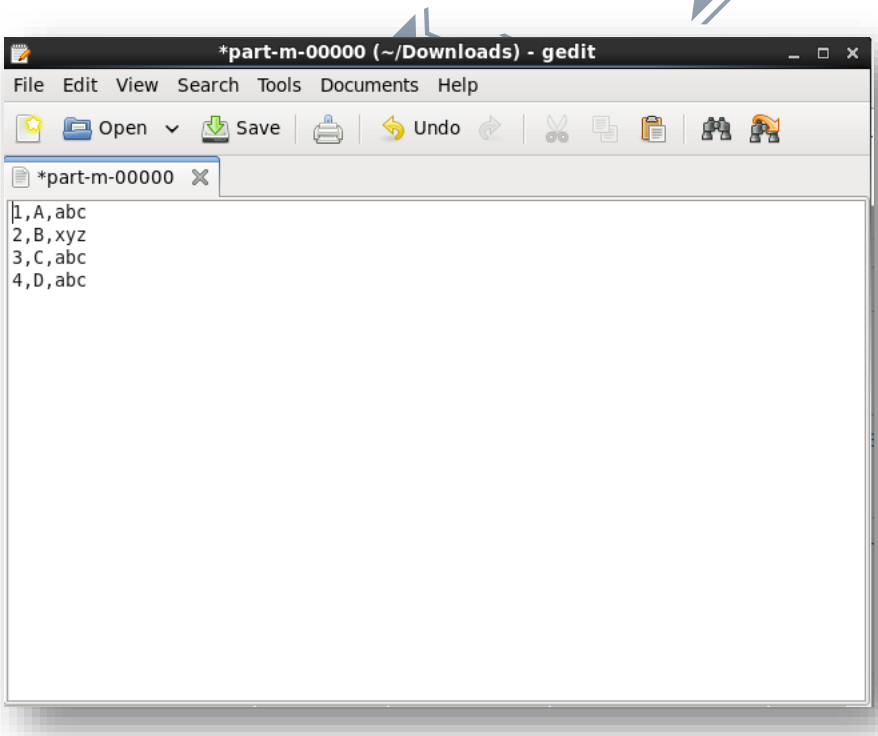
```
(1,A,abc)
(2,B,xyz)
(3,C,abc)
(4,D,abc)
grunt>
```

(ii) store:

```
grunt> store st into 'hdfs://quickstart.cloudera:8020/user/cloudera/out1/' using PigStorage(',');

2023-10-02 04:25:05,402 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2023-10-02 04:25:05,402 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-02 04:25:05,404 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.textoutputformat.separator is deprecated. Instead, use mapreduce.output.textoutputformat.separator
2023-10-02 04:25:05,410 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
```

Output:


Browsing HDFS
Index of ftp://ftp.ncdc...
Sign in to Outlook
quickstart.cloudera:50070/explorer.html#
Search
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
Oozie
Cloudera Manager
MB 08:12:51 -0700 2023 MB
-rw-r--r-- cloudera cloudera 27 B Fri Aug 25 06:42:31 -0700 2023 1 128 MB no.txt
drwxr-xr-x cloudera cloudera 0 B Wed Sep 06 08:17:01 -0700 2023 0 0 B out
drwxr-xr-x cloudera cloudera 0 B Mon Oct 02 04:25:19 -0700 2023 0 0 B out1
-rw-r--r-- cloudera cloudera 32 B Mon Oct 02 04:12:02 -0700 2023 1 128 MB student.txt


*part-m-00000 (~/Downloads) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
*part-m-00000
1,A,abc
2,B,xyz
3,C,abc
4,D,abc

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-07

AIM: Write Pig Latin scripts to perform data processing operations.

i. Grouping and joining data
ii. Sorting data
iii. Combining and Splitting data

PROGRAM:

- Create two files.

```
[cloudera@quickstart ~]$ cat > data1.txt
1,A,21,US
2,B,22,UP
3,B,22,UP
4,C,23,JP
5,D,23,IN
6,E,21,IN
^Z
[2]+  Stopped                 cat > data1.txt
[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ cat > data2.txt
1,P,21,UK
3,Q,21,UK
2,R,22,US
5,S,23,JP
9,T,21,EN
11,U,21,JP
^Z
[3]+  Stopped                 cat > data2.txt
[cloudera@quickstart ~]$
```

- To enter into local mode use **pig -x local**

```
grunt> d1 = Load 'data1.txt' using PigStorage(',') as (no:int,name:chararray,age
:int,address:chararray);
grunt>
```

```
grunt> dump d1;
2023-10-04 08:30:45,472 [main] INFO  org.apache.pig.tools.pigstats.ScriptState -
 Pig features used in the script: UNKNOWN
2023-10-04 08:30:45,546 [main] INFO  org.apache.pig.newplan.logical.optimizer.Lo
gicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateFor
EachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptim
izer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimiz
er, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter],
RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-04 08:30:45,761 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? fal
se
```

Output:

```
(1,A,21,US)
(2,B,22,UP)
(3,B,22,UP)
(4,C,23,JP)
(5,D,23,IN)
(6,E,21,IN)
```

(i) Grouping and joining data

Grouping:

- **group of single column:**

```
grunt> r1 = group d1 by age;
grunt>
```

```
grunt> dump r1;
2023-10-04 08:51:53,006 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: GROUP_BY
2023-10-04 08:51:53,012 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimize
r - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstPar
allelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEa
ch, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCas
tInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-04 08:51:53,042 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-10-04 08:51:53,055 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MultiQueryOptimizer - MR plan size before optimization: 1
```

Output:

```
(21,{(6,E,21,IN),(1,A,21,US)})
(22,{(3,B,22,UP),(2,B,22,UP)})
(23,{(5,D,23,IN),(4,C,23,JP)})
```

- **group of multiple columns:**

```
grunt> r2 = group d1 by (age,address);
grunt>
```

```
grunt> dump r2;
2023-10-04 08:55:32,914 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: GROUP_BY
2023-10-04 08:55:32,921 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimize
r - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstPar
allelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEa
ch, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCas
tInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-04 08:55:32,929 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-10-04 08:55:32,929 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MultiQueryOptimizer - MR plan size before optimization: 1
```

Output:

```
((21,IN),{(6,E,21,IN)})
((21,US),{(1,A,21,US)})
((22,UP),{(3,B,22,UP),(2,B,22,UP)})
((23,IN),{(5,D,23,IN)})
((23,JP),{(4,C,23,JP)})
```

- **group of all data:**

```
grunt> r3 = group d1 all;
grunt>
```

```
grunt> dump r3;
2023-10-04 08:59:20,303 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: GROUP_BY
2023-10-04 08:59:20,304 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimize
r - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstPar
allelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEa
ch, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCas
tInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-04 08:59:20,317 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-10-04 08:59:20,318 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MultiQueryOptimizer - MR plan size before optimization: 1
```

Output:

```
(all,{(6,E,21,IN),(5,D,23,IN),(4,C,23,JP),(3,B,22,UP),(2,B,22,UP),(1,A,21,US)})
```

- **cogroup:**

```
grunt> d2 = Load 'data2.txt' using PigStorage(',') as (eno:int,ename:chararray,eage:int,eaddress:
chararray);
```

```
grunt> c1 = cogroup d1 by age,d2 by eage;
```

```
grunt> dump c1;
2023-10-04 09:03:43,883 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features use
d in the script: COGROUP
2023-10-04 09:03:43,884 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimize
r - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstPar
allelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEa
ch, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCas
tInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-04 09:03:43,895 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MRCompiler - File concatenation threshold: 100 optimistic? false
2023-10-04 09:03:43,897 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer
.MultiQueryOptimizer - MR plan size before optimization: 1
```

Output:

```
(21,{(6,E,21,IN),(1,A,21,US)},{(11,U,21,JP),(9,T,21,EN),(3,Q,21,UK),(1,P,21,UK)})
(22,{(3,B,22,UP),(2,B,22,UP)},{(2,R,22,US)})
(23,{(5,D,23,IN),(4,C,23,JP)},{(5,S,23,JP)})
```

Joining:

- **self join:**

```
grunt> d3 = Load 'data1.txt' using PigStorage(',') as (no:int,name:chararray,age:int,address:chararray);
grunt>
```

```
grunt> j1 = join d1 by no,d3 by no;
grunt>
```

```
grunt> dump j1;
2023-10-05 02:55:25,349 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN
2023-10-05 02:55:25,602 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
orEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, Merge
imizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplif
}
2023-10-05 02:55:25,778 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
alse
2023-10-05 02:55:26,022 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler$LastInputStreamingO
>POForEach to POJoinPackage
2023-10-05 02:55:26,023 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
2023-10-05 02:55:26,023 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
2023-10-05 02:55:26,095 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=J
y initialized
```

Output:

```
(1,A,21,US,1,A,21,US)
(2,B,22,UP,2,B,22,UP)
(3,B,22,UP,3,B,22,UP)
(4,C,23,JP,4,C,23,JP)
(5,D,23,IN,5,D,23,IN)
(6,E,21,IN,6,E,21,IN)
```

- **inner join:**

```
grunt> j2 = join d1 by no,d2 by eno;
grunt>
```

```
grunt> dump j2;
2023-10-05 03:05:13,899 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN
2023-10-05 03:05:13,963 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
timizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, Spl
}
2023-10-05 03:05:14,226 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
2023-10-05 03:05:14,242 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler$LastInputStreamingO
2023-10-05 03:05:14,245 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
2023-10-05 03:05:14,245 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
2023-10-05 03:05:14,270 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=J
2023-10-05 03:05:14,279 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig script settings are added to the job
```

Output:

```
(1,A,21,US,1,P,21,UK)
(2,B,22,UP,2,R,22,US)
(3,B,22,UP,3,Q,21,UK)
(5,D,23,IN,5,S,23,JP)
```

- **left outer join:**

```
grunt> j3 = join d1 by no left outer,d2 by eno;
grunt>
```

```
grunt> dump j3;
2023-10-05 03:09:04,669 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN
2023-10-05 03:09:04,714 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:09:05,880 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:09:05,901 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 1
```

Output:

```
(1,A,21,US,1,P,21,UK)
(2,B,22,UP,2,R,22,US)
(3,B,22,UP,3,Q,21,UK)
(4,C,23,JP,,,,)
(5,D,23,IN,5,S,23,JP)
(6,E,21,IN,,,,)
```

- **right outer join:**

```
grunt> j4 = join d1 by no right outer,d2 by eno;
grunt>
```

```
grunt> dump j4;
2023-10-05 03:11:33,285 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN
2023-10-05 03:11:33,285 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:11:33,294 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:11:33,295 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 1
2023-10-05 03:11:33,295 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size after optimization: 1
```

Output:

```
(1,A,21,US,1,P,21,UK)
(2,B,22,UP,2,R,22,US)
(3,B,22,UP,3,Q,21,UK)
(5,D,23,IN,5,S,23,JP)
(,,,,9,T,21,EN)
(,,,,11,U,21,JP)
```

- **full outer join:**

```
grunt> j5 = join d1 by no full,d2 by eno;
grunt>
```

```
grunt> dump j5;
2023-10-05 03:18:38,390 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN
2023-10-05 03:18:38,391 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:18:38,400 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:18:38,402 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 1
2023-10-05 03:18:38,402 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size after optimization: 1
```

Output:

```
(1,A,21,US,1,P,21,UK)
(2,B,22,UP,2,R,22,US)
(3,B,22,UP,3,Q,21,UK)
(4,C,23,JP,,,,)
(5,D,23,IN,5,S,23,JP)
(6,E,21,IN,,,,)
(,,,,9,T,21,EN)
(,,,,11,U,21,JP)
```

- **cross product:**

```
grunt> c1 = cross d1,d2;
grunt>
```

```
grunt> dump c1;
2023-10-05 03:24:33,765 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: CROSS
2023-10-05 03:24:33,766 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:24:33,811 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:24:33,825 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler$LastInputStreamingO
ptimizer - Rewrite: POPackage->POForEach to POJoinPackage
2023-10-05 03:24:33,825 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 1
```

Output:

```
(6,E,21,IN,11,U,21,JP)
(6,E,21,IN,9,T,21,EN)
(6,E,21,IN,5,S,23,JP)
(6,E,21,IN,2,R,22,US)
(6,E,21,IN,3,Q,21,UK)
(6,E,21,IN,1,P,21,UK)
(5,D,23,IN,11,U,21,JP)
(5,D,23,IN,9,T,21,EN)
(5,D,23,IN,5,S,23,JP)
(5,D,23,IN,2,R,22,US)
(5,D,23,IN,3,Q,21,UK)
(5,D,23,IN,1,P,21,UK)
(4,C,23,JP,11,U,21,JP)
(4,C,23,JP,9,T,21,EN)
(4,C,23,JP,5,S,23,JP)
(4,C,23,JP,2,R,22,US)
(4,C,23,JP,3,Q,21,UK)
(4,C,23,JP,1,P,21,UK)
(3,B,22,UP,11,U,21,JP)
(3,B,22,UP,9,T,21,EN)
(3,B,22,UP,5,S,23,JP)
(3,B,22,UP,2,R,22,US)
(3,B,22,UP,3,Q,21,UK)
(3,B,22,UP,1,P,21,UK)
(2,B,22,UP,11,U,21,JP)
(2,B,22,UP,9,T,21,EN)
(2,B,22,UP,5,S,23,JP)
(2,B,22,UP,2,R,22,US)
(2,B,22,UP,3,Q,21,UK)
(2,B,22,UP,1,P,21,UK)
(1,A,21,US,11,U,21,JP)
(1,A,21,US,9,T,21,EN)
(1,A,21,US,5,S,23,JP)
(1,A,21,US,2,R,22,US)
(1,A,21,US,3,Q,21,UK)
(1,A,21,US,1,P,21,UK)
```


ii. Sorting data

- **Descending order:**

```
grunt> s = order d1 by age desc;
grunt>
```

```
grunt> dump s;
2023-10-05 03:27:58,377 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: ORDER_BY
2023-10-05 03:27:58,377 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
timizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, Spl
}
2023-10-05 03:27:58,393 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
2023-10-05 03:27:58,423 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
2023-10-05 03:27:58,423 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
2023-10-05 03:27:58,424 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=J
2023-10-05 03:27:58,425 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig script settings are added to the job
```

Output:

```
(5,D,23,IN)
(4,C,23,JP)
(3,B,22,UP)
(2,B,22,UP)
(6,E,21,IN)
(1,A,21,US)
```

- **Ascending order:**

```
grunt> s1 = order d1 by age;
grunt>
```

```
grunt> dump s1;
2023-10-05 03:30:46,573 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: ORDER_BY
2023-10-05 03:30:46,575 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:30:46,581 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:30:46,583 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 3
2023-10-05 03:30:46,583 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size after optimization: 3
2023-10-05 03:30:46,584 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=J
obTracker, sessionId= - already initialized
```

Output:

```
(6,E,21,IN)
(1,A,21,US)
(3,B,22,UP)
(2,B,22,UP)
(5,D,23,IN)
(4,C,23,JP)
```

- **limit:**

```
grunt> l = limit d1 3;
grunt>
```

```
grunt> dump l;
2023-10-05 03:34:47,991 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: LIMIT
2023-10-05 03:34:47,991 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:34:48,025 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:34:48,035 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 2
2023-10-05 03:34:48,035 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size after optimization: 2
```

Output:

```
(1,A,21,US)
(2,B,22,UP)
(3,B,22,UP)
```

iii. Combining and Splitting data

Combining:

```
grunt> u = union d1,d2;
grunt>
```

```
grunt> dump u;
2023-10-05 03:38:26,022 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNION
2023-10-05 03:38:26,022 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:38:26,029 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:38:26,030 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 1
2023-10-05 03:38:26,031 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size after optimization: 1
```

Output:

```
(1,A,21,US)
(2,B,22,UP)
(3,B,22,UP)
(4,C,23,JP)
(5,D,23,IN)
(6,E,21,IN)
(1,P,21,UK)
(3,Q,21,UK)
(2,R,22,US)
(5,S,23,JP)
(9,T,21,EN)
(11,U,21,JP)
```

Splitting:

```
grunt> split d1 into f1 if no>4,f2 if age>22;
grunt>
```

```
grunt> dump f1;
2023-10-05 03:42:23,346 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2023-10-05 03:42:23,346 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:42:23,351 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:42:23,352 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 2
2023-10-05 03:42:23,352 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - Merged t
he only map-only splittee.
2023-10-05 03:42:23,352 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size after optimization: 1
2023-10-05 03:42:23,353 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=J
obTracker, sessionId= - already initialized
```

Output:

```
(5,D,23,IN)
(6,E,21,IN)
```

```
grunt> dump f2;
2023-10-05 03:43:43,874 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNKNOWN
2023-10-05 03:43:43,875 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach,
 ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCa
stInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTyp
eCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2023-10-05 03:43:43,881 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenatio
n threshold: 100 optimistic? false
2023-10-05 03:43:43,882 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size before optimization: 2
2023-10-05 03:43:43,882 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - Merged t
he only map-only splittee.
2023-10-05 03:43:43,882 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan
size after optimization: 1
2023-10-05 03:43:43,883 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=J
obTracker, sessionId= - already initialized
```

Output:

```
(4,C,23,JP)
(5,D,23,IN)
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-08

AIM: Implement user defined functions in PIG.

Description:

1) In terminal, create a dataset with some fields.

```
[cloudera@quickstart ~]$ cat > student.txt
1,James,Gosling,Mar,1920
2,Dennis,Ritche,Sep,1930
9,Bjarne,Stroustrup,Jan,1912
11,Guido,Van Rossum,Apr,1915
15,Berners,Lee,Jun,1923
17,Andreas,Rumpf,Aug,1925
^Z
[1]+  Stopped                 cat > student.txt
[cloudera@quickstart ~]$
```

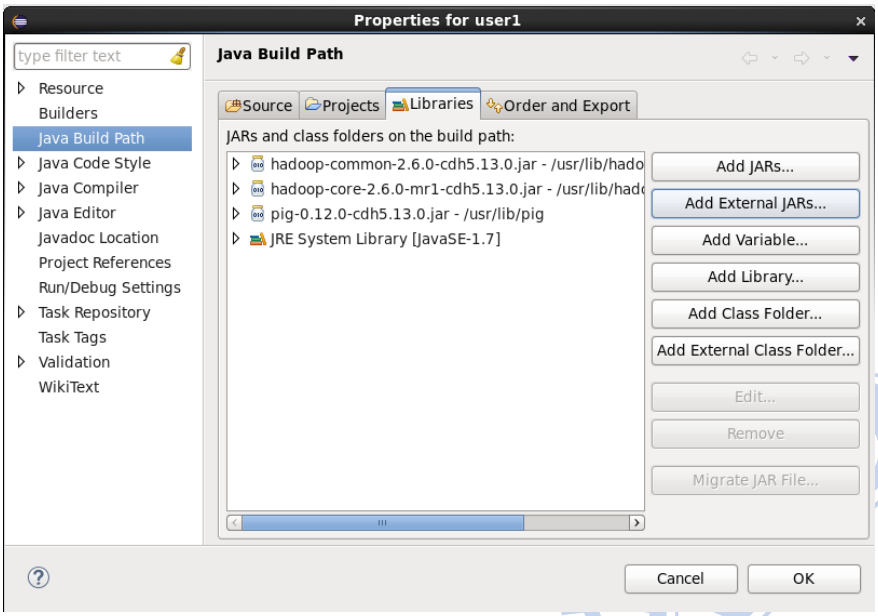
2) In eclipse->File->New->Java project->Project name “user1”->Finish.
3) Create new package (user1->New->Package->Name “pack1”->Finish).
4) Create a class (pack1->New->Class->Name “User”->Finish).
5) Add external jar files.

Right click on user1->Build path->Configure Build path->Add external JARS.

(usr\lib\hadoop\hadoop-common-2.6.0-cdh 5.13.0 jar,

usr\lib\hadoop-0.20-mapreduce\hadoop-core-2.6.0-cdh 5.13.0 jar,

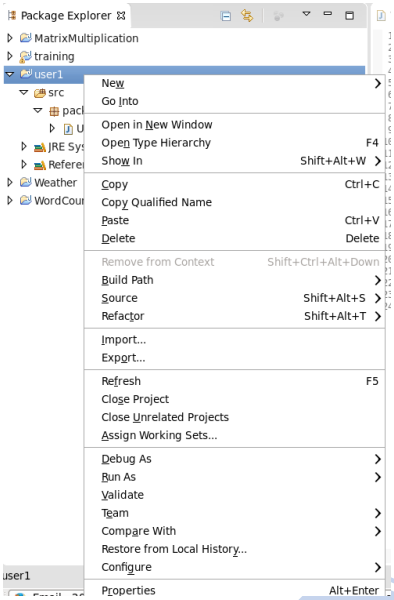
usr\lib\pig\pig-0.12.0-cdh5.13.0.jar)



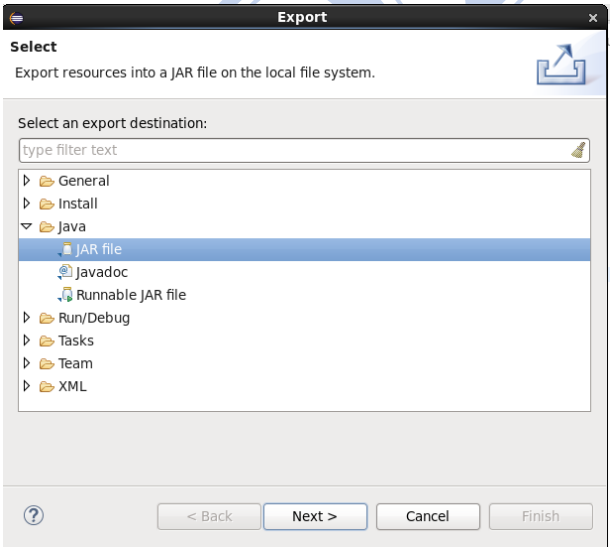
Program:

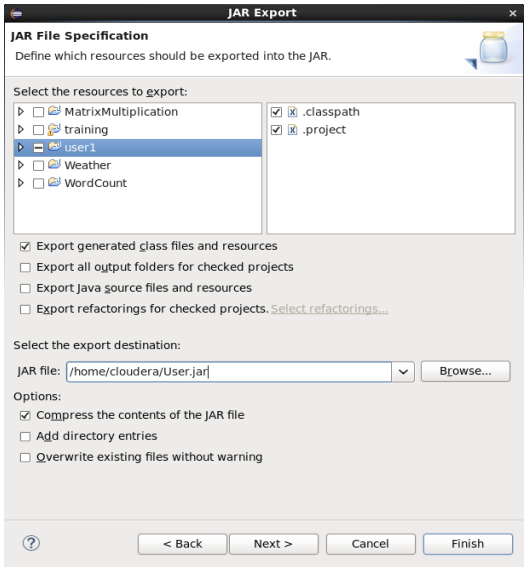
```
package pack1;
import java.io.IOException;
import org.apache.pig.EvalFunc;
import org.apache.pig.data.Tuple;

public class User extends EvalFunc <String>

 {
   public String exec(Tuple t) throws IOException

        {
              if(t.size()==0 || t==null)
                    return null;
              String fname=(String)t.get(0);
              String id1=fname.substring(0,2);
              String lname=(String)t.get(1);
              String id2=lname.substring(0,3);
              String month=(String)t.get(2);
              String id3=month.substring(0,1);
              String dob=(String)t.get(3);
              String id4=dob.substring(0,2);
```

```
        String id=id1+id2+id3+id4;

        return "UserID is: "+id;
    }

}
```



- Right click on **user1**->Export->java->jar file->JAR file: Browse (/home/cloudera/User.jar)->Finish

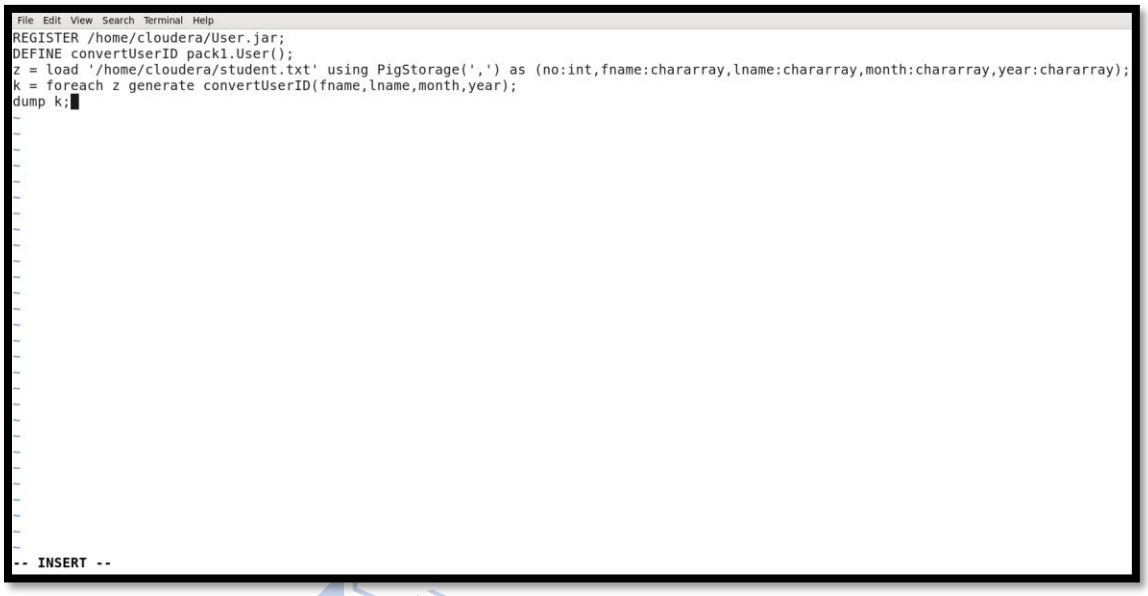




- In terminal

```
[cloudera@quickstart ~]$ vi UserId.pig
```

then automatically vi editor will open and press **"Insert"**



- Press **Esc** and **shift + zz ,** then enters into cloudera environment.

Output:

```
[cloudera@quickstart ~]$ pig -x local UserId.pig
```

```
(UserID is: JaGosM19)
(UserID is: DeRitS19)
(UserID is: BjStrJ19)
(UserID is: GuVanA19)
(UserID is: BeLeeJ19)
(UserID is: AnRumA19)
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-09

AIM: Install Hive and use Hive to create databases and tables

i. Create and drop databases

ii. Create, alter, and drop tables

iii. Insert, Update and delete records

PROGRAM:

Installation procedure:

- **Download and extract Hive:**

Command: wget https://archive.apache.org/dist/hive/hive-0.14.0/apache-hive-0.14.0- bin.tar.gz
Command: tar zxvf apache-hive-0.14.0-bin.tar.gz
Command: sudo mv apache-hive-0.13.1-bin /usr/lib/hive
Command: sudo gedit

$HOME/.bashrc
export HIVE_HOME=/usr/lib/hive
export PATH=$PATH:$HIVE_HOME/bin
export CLASSPATH=$CLASSPATH:/usr/lib/hadoop/lib/*.jar
export CLASSPATH=$CLASSPATH:/usr/lib/hive/lib/*.jar
Command: sudo cd $HIVE_HOME/conf
Command: sudo cp hive-env.sh.template hive-env.sh export HADOOP_HOME=/usr/lib/Hadoop

To enter into hive mode use "**hive**"

```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
```

i. Create and drop databases

```
hive> create database gecit;
OK
Time taken: 0.3 seconds
hive> drop database gecit;
OK
Time taken: 0.914 seconds
```

ii. Create, alter, and drop tables

Table Creation

- Create a database

```
hive> create database it;
OK
Time taken: 13.687 seconds
```

- use the database

```
hive> use it;
OK
Time taken: 0.354 seconds
```

- Create a dataset with some fields.

```
[cloudera@quickstart ~]$ cat > student.txt
1200 John
1201 Dennis
1202 James
1203 Guido
1204 Thomas
^Z
[1]+  Stopped                 cat > student.txt
[cloudera@quickstart ~]$
```

- Create a table and load the dataset into that table

```
hive> create table student(rno int,name string) comment 'It student' row format delimited fields terminated by ' ' lines terminated by
'\n';
OK
Time taken: 3.163 seconds
hive>
```

```
hive> load data local Inpath '/home/cloudera/students.txt' into table student;
Loading data to table it.student
Table it.student stats: [numFiles=1, totalSize=56]
OK
Time taken: 18.378 seconds
```

```
hive> select * from student;
OK
1200    John
1201    Dennis
1202    James
1203    Guido
1204    Thomas
Time taken: 2.523 seconds, Fetched: 5 row(s)
```

Alter Table

Renaming Table Name

```
hive> alter table student rename to students;
OK
Time taken: 8.58 seconds
```

Output

```
hive> select * from students;
OK
1200    John
1201    Dennis
1202    James
1203    Guido
1204    Thomas
Time taken: 2.67 seconds, Fetched: 5 row(s)
```

CHANGE Column

```
hive> alter table students change rno sno int;
OK
Time taken: 2.87 seconds
```

ADD Column

```
hive> alter table students add columns(addr string,mail string);
OK
Time taken: 4.224 seconds
```

Output

```
hive> select * from students;
OK
1200    John    NULL    NULL
1201    Dennis  NULL    NULL
1202    James   NULL    NULL
1203    Guido   NULL    NULL
1204    Thomas  NULL    NULL
Time taken: 0.638 seconds, Fetched: 5 row(s)
```

REPLACE Column

```
hive> alter table students replace columns(sub string,marks double)
OK
Time taken: 2.047 seconds
```

Output

```
hive> describe students;
OK
sub                     string
marks                   double
Time taken: 5.152 seconds, Fetched: 2 row(s)
```

DROP Table

```
hive> drop table students;
OK
Time taken: 7.889 seconds
```

Output

```
hive> describe students;
FAILED: SemanticException [Error 10001]: Table not found students
```

iii. Insert, Update and delete records

```
hive> create table employees(eno int,ename string,esal int);
OK
Time taken: 5.53 seconds
```

Insert

```
hive> insert into table employees values(1200,'James',10000);
```

```
hive> insert into table employees values(1206,'Dennis',15000);
```

```
hive> insert into table employees values(1203,'Gosling',13000);
```

```
hive> insert into table employees values(1201,'Ritche',18000);
```

```
hive> insert into table employees values(1208,'Guido',20000);
```

Output

```
hive> select * from employees;
OK
1200    James   10000
1206    Dennis  15000
1203    Gosling 13000
1201    Ritche  18000
1208    Guido   20000
Time taken: 4.377 seconds, Fetched: 5 row(s)
```

Update

```
hive> update employees set esal=25000 where eno=1206;
```

Delete

```
hive> delete from employees where rno=1201;
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-10

AIM: Perform data processing operations using Hive

i. Sort and Aggregation of data

ii. Joins

PROGRAM:

- Create two datasets with some fields

```
[cloudera@localhost ~]$ cat > students.txt
1,John,12000,18
2,James,15000,20
3,Dennis,20000,25
4,Gosling,14000,23
5,Thomas,10000,22
^Z
[5]+  Stopped                 cat > students.txt
```

```
[cloudera@localhost ~]$ cat > stucourse.txt
BDA,1
Python,2
BDA,4
C,9
C,5
BDA,3
Java,10
^Z
[7]+  Stopped                 cat > stucourse.txt
```

- Create a table and load the dataset into that table.

```
hive> create table employees(eno int,name string,salary int,age int) comment 'It student' row format delimited fields terminated by ',' lines terminated by '\n';
OK
Time taken: 1.218 seconds
```

```
hive> load data local inpath '/home/cloudera/students.txt' into table employees;
Copying data from file:/home/cloudera/students.txt
Copying file: file:/home/cloudera/students.txt
Loading data to table it.employees
chgrp: changing ownership of '/user/hive/warehouse/it.db/employees/students.txt': User does not belong to hive
Table it.employees stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 88, raw_data_size: 0]
OK
Time taken: 11.028 seconds
```

```
hive> select * from employees;
OK
1       John    12000   18
2       James   15000   20
3       Dennis  20000   25
4       Gosling 14000   23
5       Thomas  10000   22
Time taken: 1.363 seconds
```

- Create another table and load the dataset into that table.

```
hive> create table empcourse(cname string,eno int) comment 'It student' row format delimited fields terminated by ',' lines terminated by '\n';
OK
Time taken: 0.914 seconds
```

```
hive> load data local inpath '/home/cloudera/stucourse.txt' into table empcourse;
Copying data from file:/home/cloudera/stucourse.txt
Copying file: file:/home/cloudera/stucourse.txt
Loading data to table it.empcourse
chgrp: changing ownership of '/user/hive/warehouse/it.db/empcourse/stucourse.txt': User does not belong to hive
Table it.empcourse stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 43, raw_data_size: 0]
OK
Time taken: 1.055 seconds
```

```
hive> select * from empcourse;
OK
BDA     1
Python  2
BDA     4
C       9
C       5
BDA     3
Java    10
Time taken: 0.785 seconds
```

i. Sort and Aggregation of data

Sorting of data

Using sort by

```
hive> select * from employees sort by age;
```

Output:

```
OK
1       John    12000   18
2       James   15000   20
5       Thomas  10000   22
4       Gosling 14000   23
3       Dennis  20000   25
Time taken: 69.548 seconds
```

Using order by

```
hive> select * from employees order by age;
```

Output:

```
OK
1       John    12000   18
2       James   15000   20
5       Thomas  10000   22
4       Gosling 14000   23
3       Dennis  20000   25
Time taken: 47.292 seconds
```

Using group by

```
hive> select cname,count(cname) as count_of_cname from empcourse group by cname;
```

Output:

```
OK
BDA     3
C       2
Java    1
Python  1
Time taken: 52.176 seconds
```

Aggregation of data

Using count()

```
hive> select count(*) from employees;
```

Output:

```
OK
5
Time taken: 55.474 seconds
```

Using avg()

```
hive> select avg(salary) from employees;
```

Output:

```
OK
14200.0
Time taken: 54.35 seconds
```

Using sum()

```
hive> select sum(salary) from employees;
```

Output:

```
OK
71000
Time taken: 55.772 seconds
```

Using max()

```
hive> select max(salary) from employees;
```

Output:

```
OK
20000
Time taken: 51.093 seconds
```

Using min()

```
hive> select min(salary) from employees;
```

Output:

```
OK
10000
Time taken: 51.306 seconds
```

ii. Joins

Using inner join

```
hive> select empcourse.cname,employees.name from employees inner join empcourse on employees.eno = empcourse.eno;
```

Output:

```
OK
BDA     John
Python  James
BDA     Dennis
BDA     Gosling
C       Thomas
Time taken: 101.828 seconds
```

Using left outer join

```
hive> select employees.name,empcourse.cname from employees left outer join empcourse on employees.eno = empcourse.eno;
```

Output:

```
OK
John    BDA
James   Python
Dennis  BDA
Gosling BDA
Thomas  C
Time taken: 69.388 seconds
```

Using right outer join

```
hive> select employees.name,empcourse.cname from employees right outer join empcourse on employees.eno = empcourse.eno;
```

Output:

```
OK
John    BDA
James   Python
Dennis  BDA
Gosling BDA
Thomas  C
NULL    C
NULL    Java
Time taken: 109.794 seconds
```

Using full outer join

```
hive> select employees.name,empcourse.cname from employees full outer join empcourse on employees.eno = empcourse.eno;
```

Output:

```
OK
John    BDA
James   Python
Dennis  BDA
Gosling BDA
Thomas  C
NULL    C
NULL    Java
Time taken: 68.767 seconds
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-11

AIM: Perform data processing operations using Hive

i. Views

ii. Indexes

PROGRAM:

- Create a dataset with some fields

```
[cloudera@localhost ~]$ cat > students.txt
1,John,12000,18
2,James,15000,20
3,Dennis,20000,25
4,Gosling,14000,23
5,Thomas,10000,22
^Z
[5]+  Stopped                 cat > students.txt
```

- Create a table and load the dataset into that table.

```
hive> create table employees(eno int,name string,salary int,age int) comment 'It student' row format delimited fields terminated by ',' lines terminated by '\n';
OK
Time taken: 1.218 seconds
```

```
hive> load data local inpath '/home/cloudera/students.txt' into table employees;
Copying data from file:/home/cloudera/students.txt
Copying file: file:/home/cloudera/students.txt
Loading data to table it.employees
chgrp: changing ownership of '/user/hive/warehouse/it.db/employees/students.txt': User does not belong to hive
Table it.employees stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 88, raw_data_size: 0]
OK
Time taken: 11.028 seconds
```

```
hive> select * from employees;
OK
1       John    12000   18
2       James   15000   20
3       Dennis  20000   25
4       Gosling 14000   23
5       Thomas  10000   22
Time taken: 1.363 seconds
```

i. Views

Creating a view

```
hive> create view empdetails as select * from employees where salary>12000;
OK
Time taken: 0.271 seconds
```

Dropping a view

```
hive> drop view empdetails;
OK
Time taken: 0.704 seconds
```

ii. Indexes

Creating an index

```
hive> create index index_salary on table employees(salary) as 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler' with deferred rebuild;
OK
Time taken: 1.988 seconds
```

Dropping an index

```
hive> drop index index_salary on employees;
OK
Time taken: 0.473 seconds
```

BIG DATA ANALYTICS LAB

EXPERIMENT_NO-12

AIM: Implement user defined functions in hive.

DESCRIPTION:

1) In eclipse->File->New->Java project->Project name "**user3**"->Finish.

2) Create new package (**user3**->New->Package->Name "UDF"->Finish).

3) Create a class (UDF->New->Class->Name "MyUpper"->Finish).

4) Add external jar files.

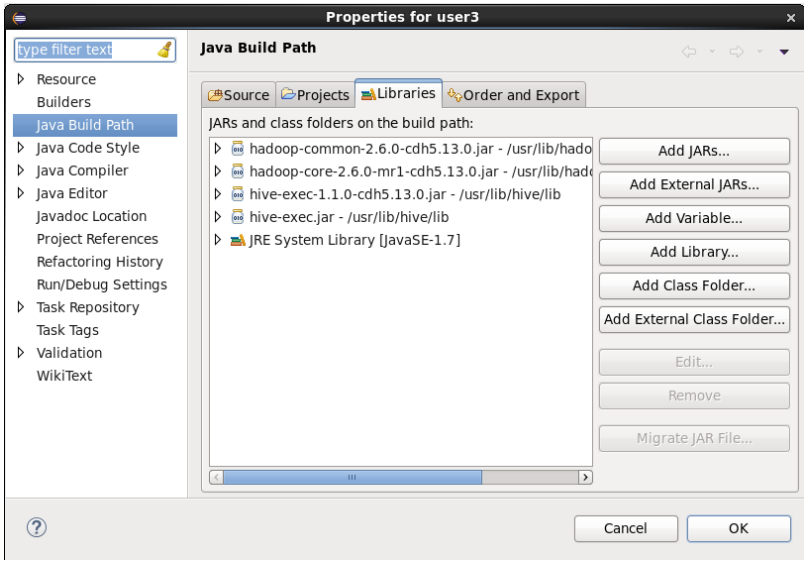
Right click on **user3**->Build path->Configure Build path->Add external JARS.

(usr\lib\hadoop\hadoop-common-2.6.0-cdh 5.13.0 jar,

usr\lib\hadoop-0.20-mapreduce\hadoop-core-2.6.0-cdh 5.13.0 jar,

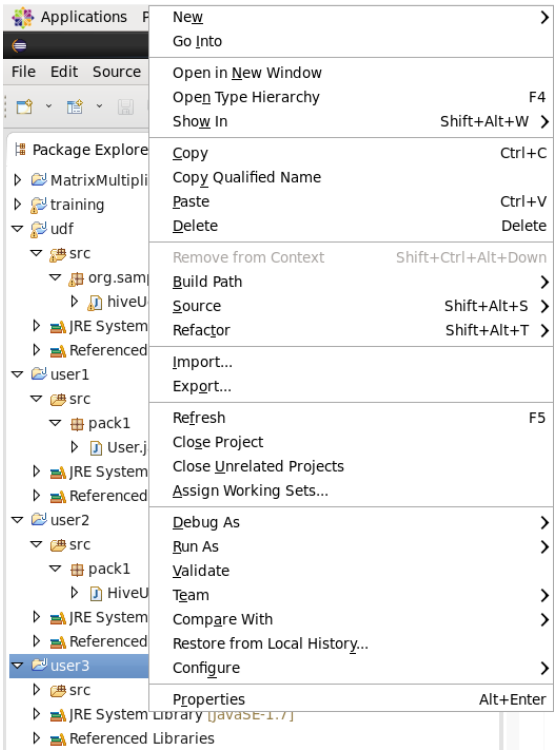
usr\lib\hive\lib\hive-exec-1.1.0-cdh5.13.0.jar,

usr\lib\hive\lib\hive-exec.jar)

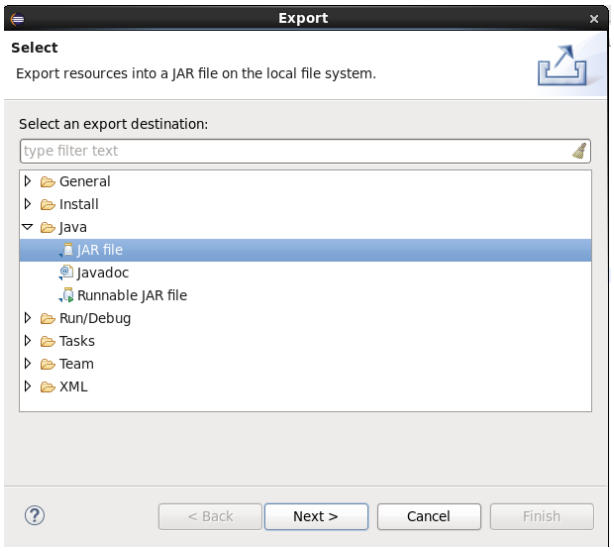


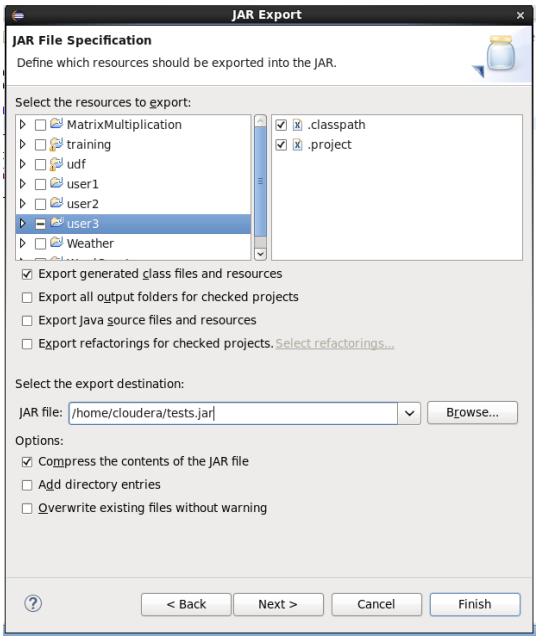
PROGRAM:

```
package UDF;
import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.Text;

@SuppressWarnings("deprecation")

public class MyUpper extends UDF {
	public Text evaluate(final Text s){
		if(s==null){
			return null;
		}
		return new Text(s.toString().toUpperCase());
	}
}
```



- Right click on **user3**->Export->java->jar file->JAR file: Browse (/home/cloudera/tests.jar)->Finish





Output:

- To enter into hive mode use "**hive**"

```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
```

```
hive> ADD JAR tests.jar;
Added [tests.jar] to class path
Added resources: [tests.jar]
```

```
hive> CREATE TEMPORARY FUNCTION myupper as 'UDF.MyUpper';
OK
Time taken: 0.427 seconds
```

```
hive> select myupper("james gosling");
OK
JAMES GOSLING
Time taken: 14.424 seconds, Fetched: 1 row(s)
```