**B.M.S. COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



Lab Record

**Big-Data Analytics**

*Submitted in partial fulfillment for the 6th Semester Laboratory*

Bachelor of Technology

in

Computer Science and Engineering

*Submitted by:*

**P Sai Deekshith**

1BM18S148

Department of Computer Science and Engineering

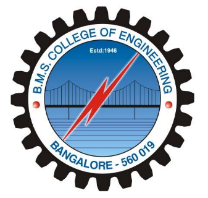
B.M.S. College of Engineering

Bull Temple Road, Basavanagudi, Bangalore 560 019

Mar-June 2021

**B.M.S. COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

***CERTIFICATE***

This is to certify that the Big-Data Analytics (20CS6PEBDA) laboratory has been carried out by P Sai Deekshith (1BM18CS148) during the 6th Semester Mar-June-2021.

Signature of the Faculty Incharge:

Bhoomika A P

Associate Professor

Department of Computer Science and Engineering

B.M.S. College of Engineering, Bangalore

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **Programs** | **Page** |
| 1 | Perform the following DB operations using Cassandra Employee. | 4 |
| 2 | Perform the following DB operations using Cassandra Library. | 6 |
| 3 | MongoDB - CRUD Demonstration. | 8 |
| 4 | Screenshot of Hadoop installed. | 10 |
| 5 | Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed) | 11 |
| 6 | Create a Map Reduce program to a) find average temperature for each year from NCDC data set. b) find the mean max temperature for every month | 19 |
| 7 | For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words. | 27 |
| 8 | Create a Map Reduce program to demonstrating join operation. | 32 |
| 9 | Screenshot of Spark Installed. | 41 |
| 10 | Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark. | 42 |

**PROGRAM - 1**

**PERFORM THE FOLLOWING DB OPERATIONS USING CASSANDRA LIBRARY.**

Create a keyspace by name Employee

cqlsh> create keyspace employee with replication = {'class': 'SimpleStrategy', 'replication\_factor': 1};

cqlsh> use employee;

Create a column family by name Employee-Info with attributes Emp\_Id Primary Key, Emp\_Name, Designation, Date\_of\_Joining, Salary, Dept\_Name

cqlsh:employee> create table employeeinfo(emp\_id int primary key, emp\_name text, designation text, doj timestamp, salary double, dept\_name text);

Insert the values into the table in batch

cqlsh:employee> begin batch

... insert into employeeinfo(emp\_id, emp\_name, designation, doj, salary, dept\_name) values (1, 'Ajay', 'Data analyst', '2018-04-16', 20000, 'Corporate');

... insert into employeeinfo(emp\_id, emp\_name, designation, doj, salary, dept\_name) values (121, 'Chaitra', 'web design', '2019-08-06', 15000, 'web\_designer');

... apply batch;

cqlsh:employee> select \* from employeeinfo;

Update Employee name and Department of Emp-Id 121

cqlsh:employee> update employeeinfo set emp\_name = 'Joy', dept\_name = 'Management' where emp\_id = 121;

cqlsh:employee> select \* from employeeinfo;

Alter the schema of the table Employee\_Info to add a column Projects which stores a set

of Projects done by the corresponding Employee.

cqlsh:employee> alter table employeeinfo add projects set<text>;

Update the altered table to add project names.

cqlsh:employee> update employeeinfo set projects = {'project1', 'project2'} where emp\_id in(1,121);

cqlsh:employee> select \* from employeeinfo;

Create a TTL of 15 seconds to display the values of Employees.

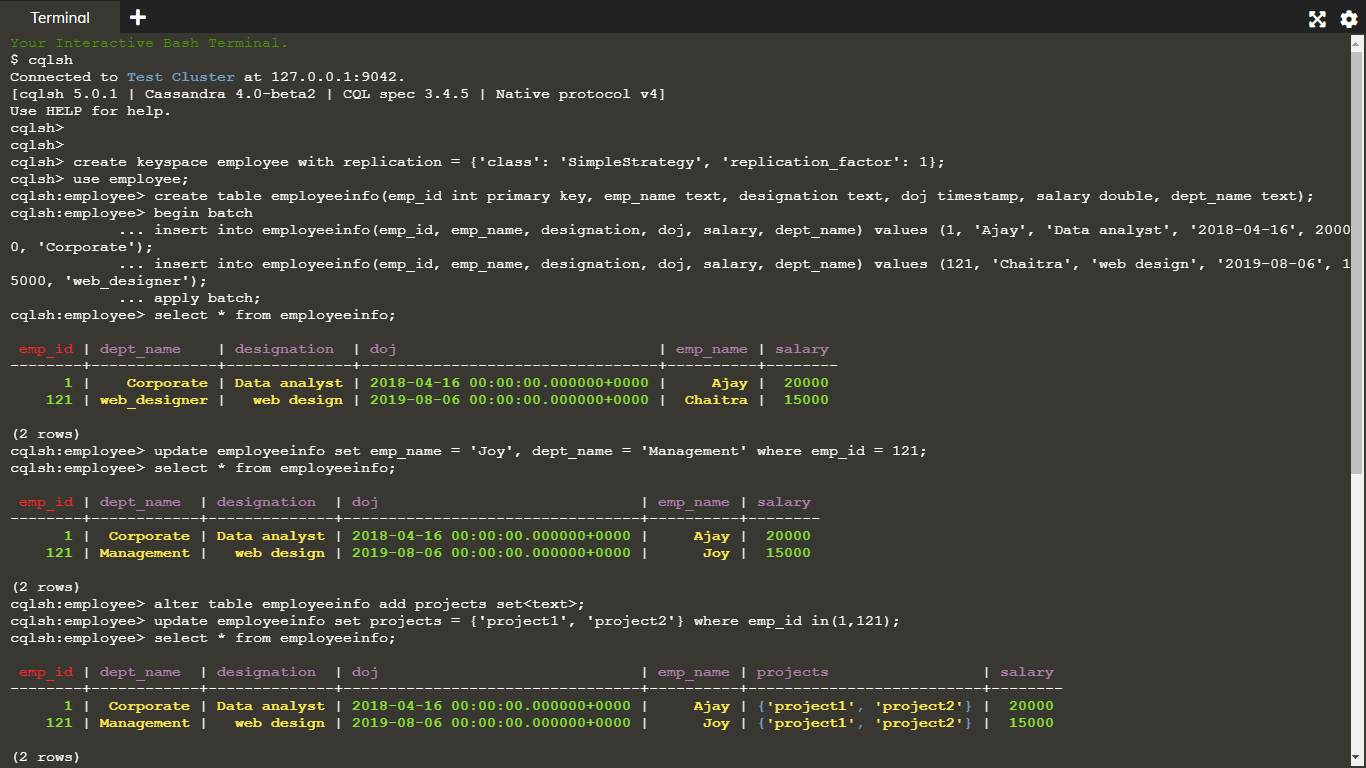
cqlsh:employee> begin batch

... insert into employeeinfo(emp\_id, emp\_name, designation, doj, salary, dept\_name) values (121, 'Boris', 'MTO', '2001-08-05', 12212, 'Corporate') using ttl 15;

... apply batch;

cqlsh:employee> select ttl(designation) from employeeinfo where emp\_id = 121;

**Output :**





**PROGRAM – 2**

**PERFORM THE FOLLOWING DB OPERATIONS USING CASSANDRA:**

Create a keyspace by name Library

cqlsh> create keyspace library with replication = { 'class' : 'SimpleStrategy','replication\_factor':1};

cqlsh> use library;

Create a column family by name Library-Info with attributes Stud\_Id Primary Key, Counter\_value of type Counter, Stud\_Name, Book-Name, Book-Id, Date\_of\_issue

cqlsh:library> create table library\_info( id int, counter\_val counter, stud\_name text, book\_name text, book\_id int, issue\_date timestamp,primary key(id,stud\_name,book\_name,book\_id,issue\_date));

Insert the values into the table in batch

cqlsh:library> update library\_info SET counter\_val = counter\_val +1 where id = 1 and stud\_name = 'Anand' and book\_name = 'CNS' and book\_id = 121 and issue\_date='2020-12-31';

cqlsh:library> update library\_info SET counter\_val = counter\_val +1 where id = 3 and stud\_name = 'Arjun' and book\_name = 'ML' and book\_id = 112 and issue\_date='2021-02-01';

cqlsh:library> update library\_info SET counter\_val = counter\_val +1 where id = 5 and stud\_name = 'Chaitra' and book\_name = 'Python' and book\_id = 114 and issue\_date='2009-08-27';

cqlsh:library> select \* from library\_info;

Display the details of the table created and increase the value of the counter

cqlsh:library> update library\_info SET counter\_val = counter\_val +1 where id = 3 and stud\_name = 'Arjun' and book\_name = 'ML' and book\_id = 112 and issue\_date='2021-02-01';

Write a query to show that a student with id 112 has taken a book “BDA” 2 times.

cqlsh:library> select \* from library\_info where counter\_val = 2 allow filtering;

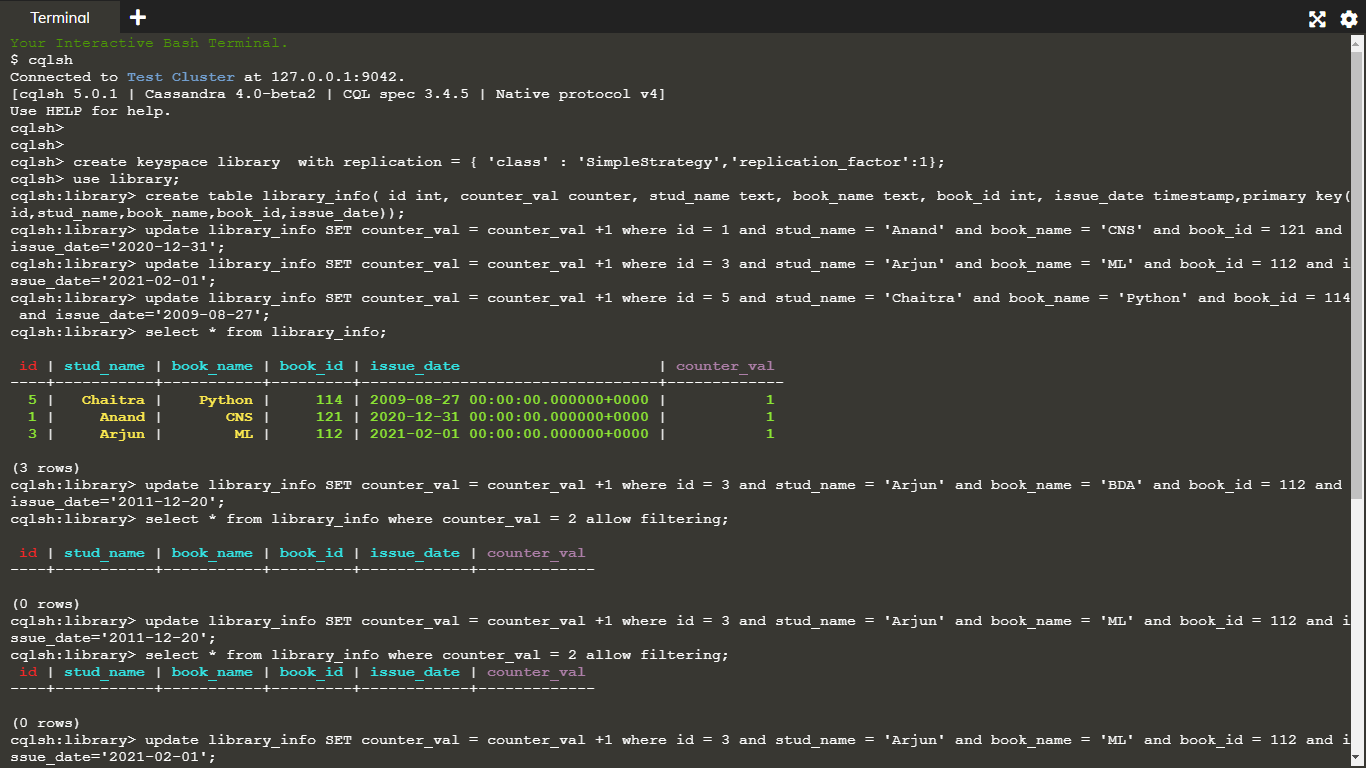
Export the created column to a csv file

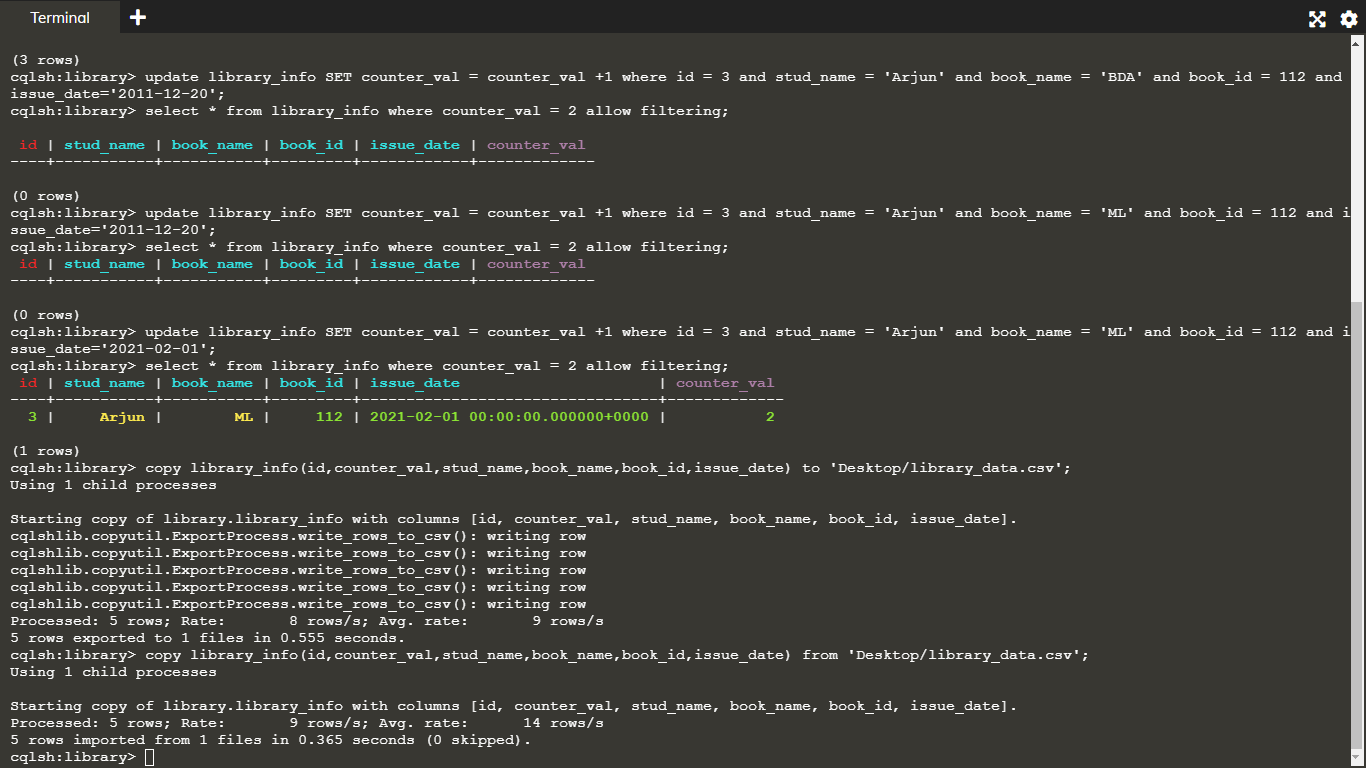
cqlsh:library> copy library\_info(id,counter\_val,stud\_name,book\_name,book\_id,issue\_date) to 'Desktop/library\_data.csv';

Import a given csv dataset from local file system into Cassandra column family

cqlsh:library> copy library\_info(id,counter\_val,stud\_name,book\_name,book\_id,issue\_date) from 'Desktop/library\_data.csv';

**Output :**





**PROGRAM – 3**

**PERFORM THE FOLLOWING  DB OPERATIONS USING MONGODB:**

Create a database “Student” with the following attributes  Rollno, Age, ContactNo, Email-Id.

use student

Insert appropriate values

db.student.insert({Roll: 10, Name: "suma", age: 21, contact: "7723112389", email: "suma@gmail.com"})

db.student.insert({Roll: 11, Name: "ABC", age: 20, contact: "9263532389", email: "abc@gmail.com"})

db.student.insert({Roll: 12, Name: "shek", age: 21, contact: "7788996655", email: "shek@gmail.com"})

db.student.insert({Roll: 13, Name: "raj", age: 20, contact: "1234123412", email: "raj@gmail.com"})

Write a query to update Email-Id of a student with rollno 10.

db.student.update({Roll:10}, {$set: {email: "suma123@gmail.com"}})

Replace the student name from “ABC” to “FEM” of rollno 11.

db.student.update({Roll:11}, {$set: {Name: "FEM"}})

Export the created table into local file system

mongoexport --db student --collection student --type csv --out D:\export.csv --fields “Roll,Name,age,contact,email”

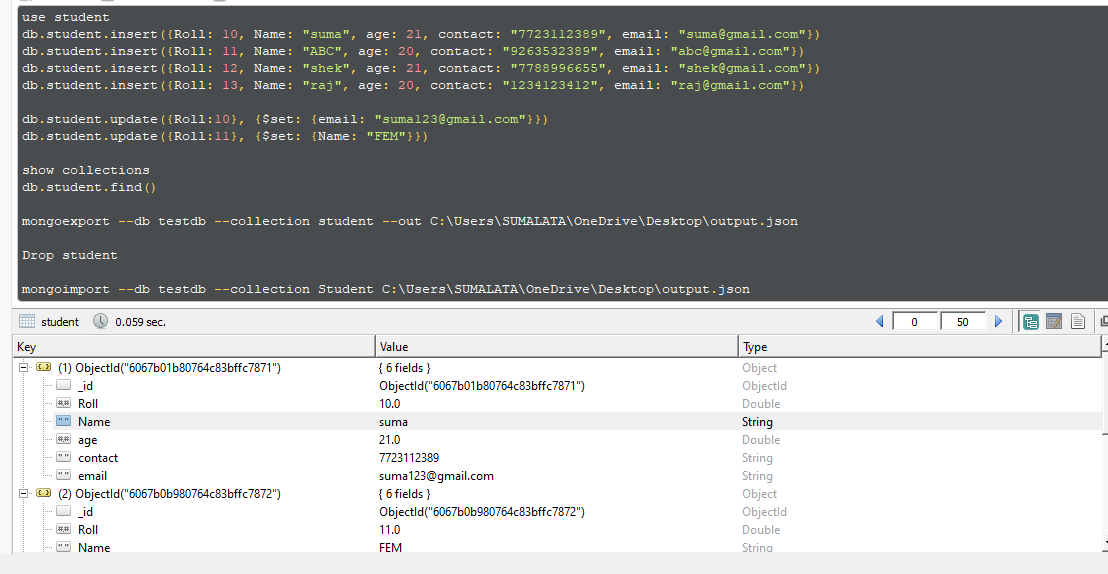
Drop the table

db.student.drop()

Import a given csv dataset from the local file system into mongodb collection.

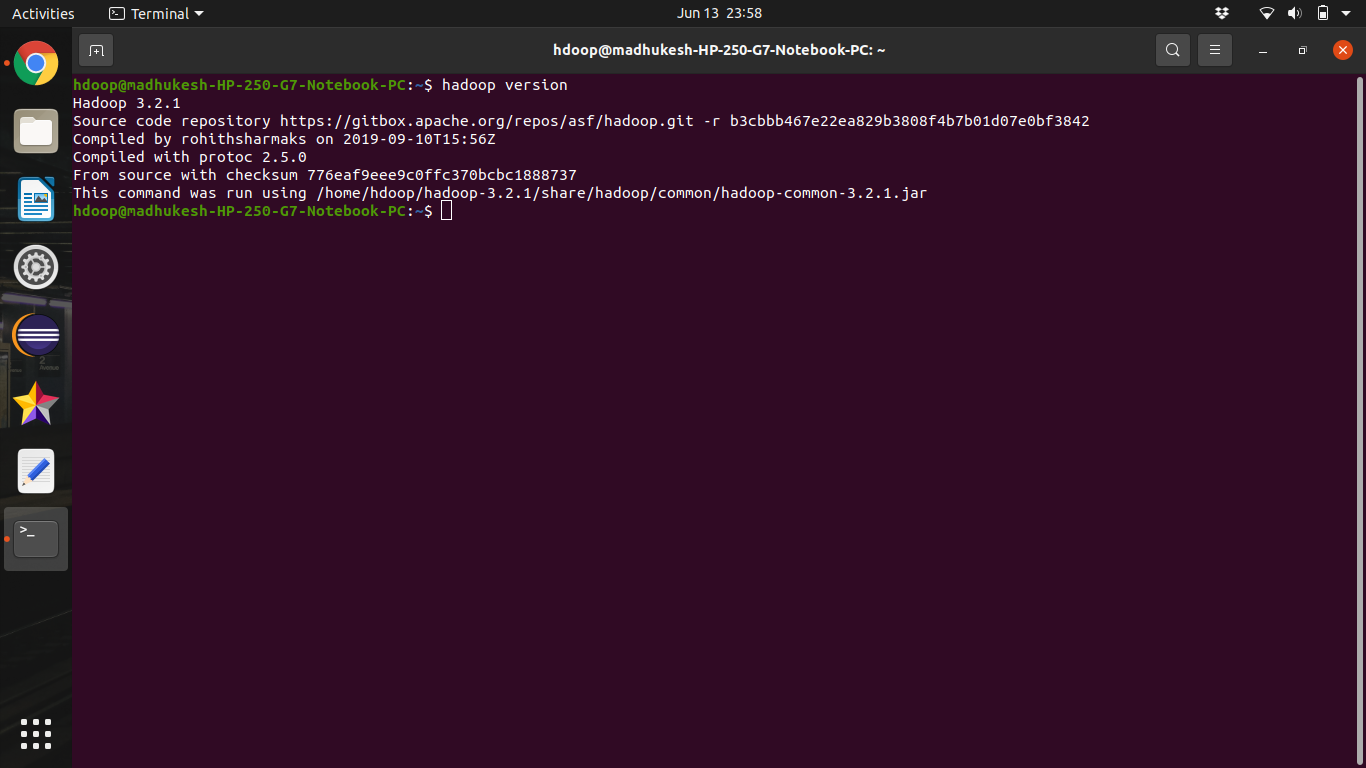
mongoimport --db student --collection student --type csv --file D:\export.csv --headerline

**Output :**



**PROGRAM – 4**

**SCREENSHOT OF HADOOP INSTALLED:**



**PROGRAM – 5**

**EXECUTION OF HDFS COMMANDS FOR INTERACTION WITH HADOOP ENVIRONMENT. (MINIMUM 10 COMMANDS TO BE EXECUTED:**

version

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop version

Hadoop 3.2.1

Source code repository https://gitbox.apache.org/repos/asf/hadoop.git -r b3cbbb467e22ea829b3808f4b7b01d07e0bf3842

Compiled by rohithsharmaks on 2019-09-10T15:56Z

Compiled with protoc 2.5.0

From source with checksum 776eaf9eee9c0ffc370bcbc1888737

This command was run using /home/hdoop/hadoop-3.2.1/share/hadoop/common/hadoop-common-3.2.1.jar

mkdir

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -mkdir /samplefile1

2021-04-20 13:37:25,376 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -mkdir /samplefile2

2021-04-20 13:37:43,271 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -mkdir /samplefile3

2021-04-20 13:38:18,887 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

ls

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls /

2021-04-20 13:38:41,762 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 3 items

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:37 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:37 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:38 /samplefile3

hdoop@madhu-HP-250-G7-Notebook-PC:~$ ls

dfsdata hadoop-3.2.1 hadoop-3.2.1.tar.gz tmpdata

hdoop@madhu-HP-250-G7-Notebook-PC:~$ mkdir lab

hdoop@madhu-HP-250-G7-Notebook-PC:~$ mkdir lab/samples/

hdoop@madhu-HP-250-G7-Notebook-PC:~$ mkdir lab/samples/file1

hdoop@madhu-HP-250-G7-Notebook-PC:~$ mkdir lab/samples/file2

hdoop@madhu-HP-250-G7-Notebook-PC:~$ mkdir lab/samples/file3

hdoop@madhu-HP-250-G7-Notebook-PC:~$ mkdir lab/samples/file4

put / copyFromLocal

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -put ~/lab/samples/file1 /

2021-04-20 13:48:24,640 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -put ~/lab/samples/file2 /samplefile1

2021-04-20 13:49:04,048 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls /

2021-04-20 13:50:32,226 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 4 items

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:48 /file1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:37 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:38 /samplefile3

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls -R /

2021-04-20 13:52:21,533 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:48 /file1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1/file2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:37 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:38 /samplefile3

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -copyFromLocal ~/lab/samples/file3 /samplefile2

2021-04-20 13:58:22,912 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -copyFromLocal ~/lab/sam'ples/file4 /samplefile3

2021-04-20 13:58:38,623 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls -R /

2021-04-20 13:58:49,088 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:48 /file1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1/file2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile2/file3

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile3

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile3/file4

get / copyToLocal

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -get /file1 ~/lab/copies

2021-04-20 19:16:54,079 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -get /samplefile1/file2 ~/lab/copies

2021-04-20 19:17:59,535 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -copyToLocal /samplefile2/file3 ~/lab/copies

2021-04-20 19:19:09,548 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -copyToLocal /samplefile3/file4 ~/lab/copies

2021-04-20 19:19:30,733 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ ls -l lab/copies

total 12

drwxr-xr-x 2 hdoop hdoop 4096 Apr 20 19:18 file2

drwxr-xr-x 2 hdoop hdoop 4096 Apr 20 19:19 file3

drwxr-xr-x 2 hdoop hdoop 4096 Apr 20 19:19 file4

hdoop@madhu-HP-250-G7-Notebook-PC:~$ ls -l lab

total 8

drwxr-xr-x 5 hdoop hdoop 4096 Apr 20 19:19 copies

drwxrwxr-x 6 hdoop hdoop 4096 Apr 20 13:47 samples

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -get /file1 ~/lab/copies

2021-04-20 19:22:17,555 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ ls -l lab/copies

total 16

drwxr-xr-x 2 hdoop hdoop 4096 Apr 20 19:22 file1

drwxr-xr-x 2 hdoop hdoop 4096 Apr 20 19:18 file2

drwxr-xr-x 2 hdoop hdoop 4096 Apr 20 19:19 file3

drwxr-xr-x 2 hdoop hdoop 4096 Apr 20 19:19 file4

hdoop@madhu-HP-250-G7-Notebook-PC:~$ mkdir lab/text

hdoop@madhu-HP-250-G7-Notebook-PC:~$ cd lab/text/

hdoop@madhu-HP-250-G7-Notebook-PC:~/lab/text$ cat > text1

Hi

I'm executing hadoop commands

hdoop@madhu-HP-250-G7-Notebook-PC:~/lab/text$ cd

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -copyFromLocal ~/lab/text/text1 /

2021-04-20 19:26:31,016 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

2021-04-20 19:26:33,108 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls /

2021-04-20 19:27:17,423 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 5 items

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:48 /file1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile3

-rw-r--r-- 1 hdoop supergroup 33 2021-04-20 19:26 /text1

cat

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -cat /text1

2021-04-20 19:28:24,990 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

2021-04-20 19:28:26,530 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

Hi

I'm executing hadoop commands

mv

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -mv /file1 /samplefile1

2021-04-20 19:31:09,926 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls /samplefile1

2021-04-20 19:31:49,316 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 2 items

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:48 /samplefile1/file1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1/file2

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls /

2021-04-20 19:32:12,458 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 4 items

drwxr-xr-x - hdoop supergroup 0 2021-04-20 19:31 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile3

-rw-r--r-- 1 hdoop supergroup 33 2021-04-20 19:26 /text1

cp

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -cp /text1 /samplefile3

2021-04-20 19:33:32,689 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

2021-04-20 19:33:34,093 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

2021-04-20 19:33:34,332 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls -R /

2021-04-20 19:33:52,862 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

drwxr-xr-x - hdoop supergroup 0 2021-04-20 19:31 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:48 /samplefile1/file1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:49 /samplefile1/file2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile2/file3

drwxr-xr-x - hdoop supergroup 0 2021-04-20 19:33 /samplefile3

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile3/file4

-rw-r--r-- 1 hdoop supergroup 33 2021-04-20 19:33 /samplefile3/text1

-rw-r--r-- 1 hdoop supergroup 33 2021-04-20 19:26 /text1

rm

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -rm /text1

2021-04-20 19:49:33,071 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Deleted /text1

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls /

2021-04-20 19:49:44,650 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 3 items

drwxr-xr-x - hdoop supergroup 0 2021-04-20 19:31 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 13:58 /samplefile2

drwxr-xr-x - hdoop supergroup 0 2021-04-20 19:33 /samplefile3

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -rm -r /samplefile2

2021-04-20 19:51:13,448 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Deleted /samplefile2

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -ls /

2021-04-20 19:51:21,573 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Found 2 items

drwxr-xr-x - hdoop supergroup 0 2021-04-20 19:31 /samplefile1

drwxr-xr-x - hdoop supergroup 0 2021-04-20 19:33 /samplefile3

du

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -du -s /samplefile3/text1

2021-04-20 19:54:16,666 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

33 33 /samplefile3/text1

df

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -df

2021-04-20 19:55:56,239 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Filesystem Size Used Available Use%

hdfs://127.0.0.1:9000 267221413888 45056 155517390848 0%

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -df -h

2021-04-20 19:56:16,756 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Filesystem Size Used Available Use%

hdfs://127.0.0.1:9000 248.9 G 44 K 144.8 G 0%

count

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -count -v -h /

2021-04-20 20:01:44,154 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

DIR\_COUNT FILE\_COUNT CONTENT\_SIZE PATHNAME

6 1 33 /

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -count -v -h -q /

2021-04-20 20:02:07,036 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

QUOTA REM\_QUOTA SPACE\_QUOTA REM\_SPACE\_QUOTA DIR\_COUNT FILE\_COUNT CONTENT\_SIZE PATHNAME

8.0 E 8.0 E none inf 6 1 33 /

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -count -v -q /

2021-04-20 20:02:22,339 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

QUOTA REM\_QUOTA SPACE\_QUOTA REM\_SPACE\_QUOTA DIR\_COUNT FILE\_COUNT CONTENT\_SIZE PATHNAME

9223372036854775807 9223372036854775800 none inf 6 1 33 /

hdoop@madhu-HP-250-G7-Notebook-PC:~$ hadoop fs -count -v -u /

2021-04-20 20:02:33,945 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicabl/e

QUOTA REM\_QUOTA SPACE\_QUOTA REM\_SPACE\_QUOTA PATHNAME

9223372036854775807 9223372036854775800 none inf /

**PROGRAM – 6**

**CREATE A MAP REDUCE PROGRAM TO**

**FIND AVERAGE TEMPERATURE FOR EACH YEAR FROM NCDC DATA SET.**

**Java Files:**

AverageReducer.java

package temperature;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

import java.io.IOException;

public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable >

{

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

{

int max\_temp = 0;

int count = 0;

for (IntWritable value : values)

{

max\_temp += value.get();

count+=1;

}

context.write(key, new IntWritable(max\_temp/count));

}

}

AverageeDriver.java

package temperature;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.fs.\*;

import org.apache.hadoop.mapreduce.\*;

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

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

public class AverageDriver

{

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

{

if (args.length != 2)

{

System.err.println("Please Enter the input and output parameters");

System.exit(-1);

}

Job job = new Job();

job.setJarByClass(AverageDriver.class);

job.setJobName("Max temperature");

FileInputFormat.addInputPath(job,new Path(args[0]));

FileOutputFormat.setOutputPath(job,new Path (args[1]));

job.setMapperClass(AverageMapper.class);

job.setReducerClass(AverageReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

System.exit(job.waitForCompletion(true)?0:1);

}

}

AverageMapper.java:

package temperature;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

import java.io.IOException;

public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>

{

public static final int MISSING = 9999;

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

{

String line = value.toString();

String year = line.substring(15,19);

int temperature;

if (line.charAt(87)=='+')

temperature = Integer.parseInt(line.substring(88, 92));

else

temperature = Integer.parseInt(line.substring(87, 92));

String quality = line.substring(92, 93);

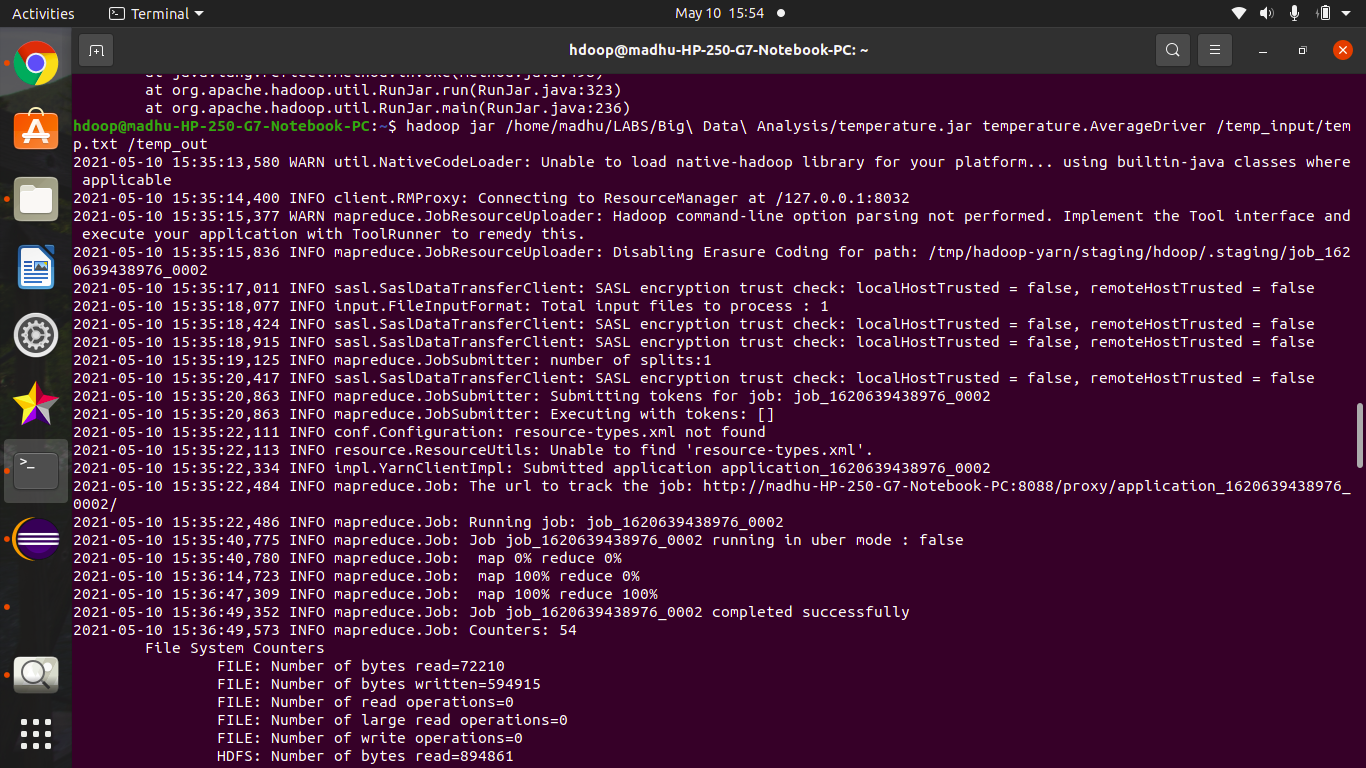
if(temperature != MISSING && quality.matches("[01459]"))

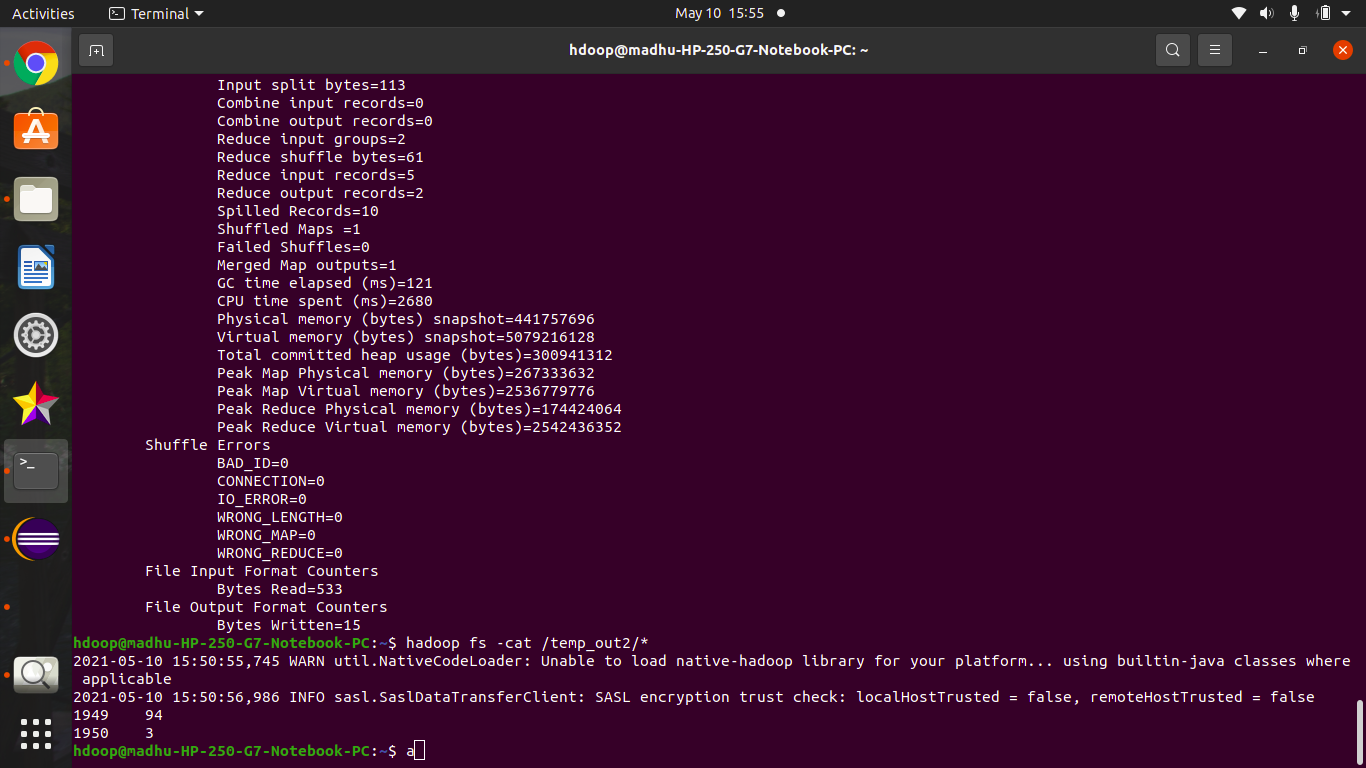
context.write(new Text(year),new IntWritable(temperature));

}

}

**Output:**





**FIND THE MEAN MAX TEMPERATURE FOR EVERY MONTH:**

**Java Files:**

MaxDriver:

package tempMax;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.fs.\*;

import org.apache.hadoop.mapreduce.\*;

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

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

public class MaxDriver

{

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

{

if (args.length != 2)

{

System.err.println("Please Enter the input and output parameters");

System.exit(-1);

}

Job job = new Job();

job.setJarByClass(MaxDriver.class);

job.setJobName("Max temperature");

FileInputFormat.addInputPath(job,new Path(args[0]));

FileOutputFormat.setOutputPath(job,new Path (args[1]));

job.setMapperClass(MaxMapper.class);

job.setReducerClass(MaxReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

System.exit(job.waitForCompletion(true)?0:1);

}

}

MaxMapper:

package tempMax;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

import java.io.IOException;

public class MaxMapper extends Mapper <LongWritable, Text, Text, IntWritable>

{

public static final int MISSING = 9999;

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

{

String line = value.toString();

String month = line.substring(19,21);

int temperature;

if (line.charAt(87)=='+')

temperature = Integer.parseInt(line.substring(88, 92));

else

temperature = Integer.parseInt(line.substring(87, 92));

String quality = line.substring(92, 93);

if(temperature != MISSING && quality.matches("[01459]"))

context.write(new Text(month),new IntWritable(temperature));

}

}

MaxReducer:

package tempMax;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapreduce.\*;

import java.io.IOException;

public class MaxReducer extends Reducer <Text, IntWritable,Text, IntWritable>

{

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

{

int max\_temp = 0;

for (IntWritable value : values)

{

if(max\_temp<value.get()) {

max\_temp = value.get();

}

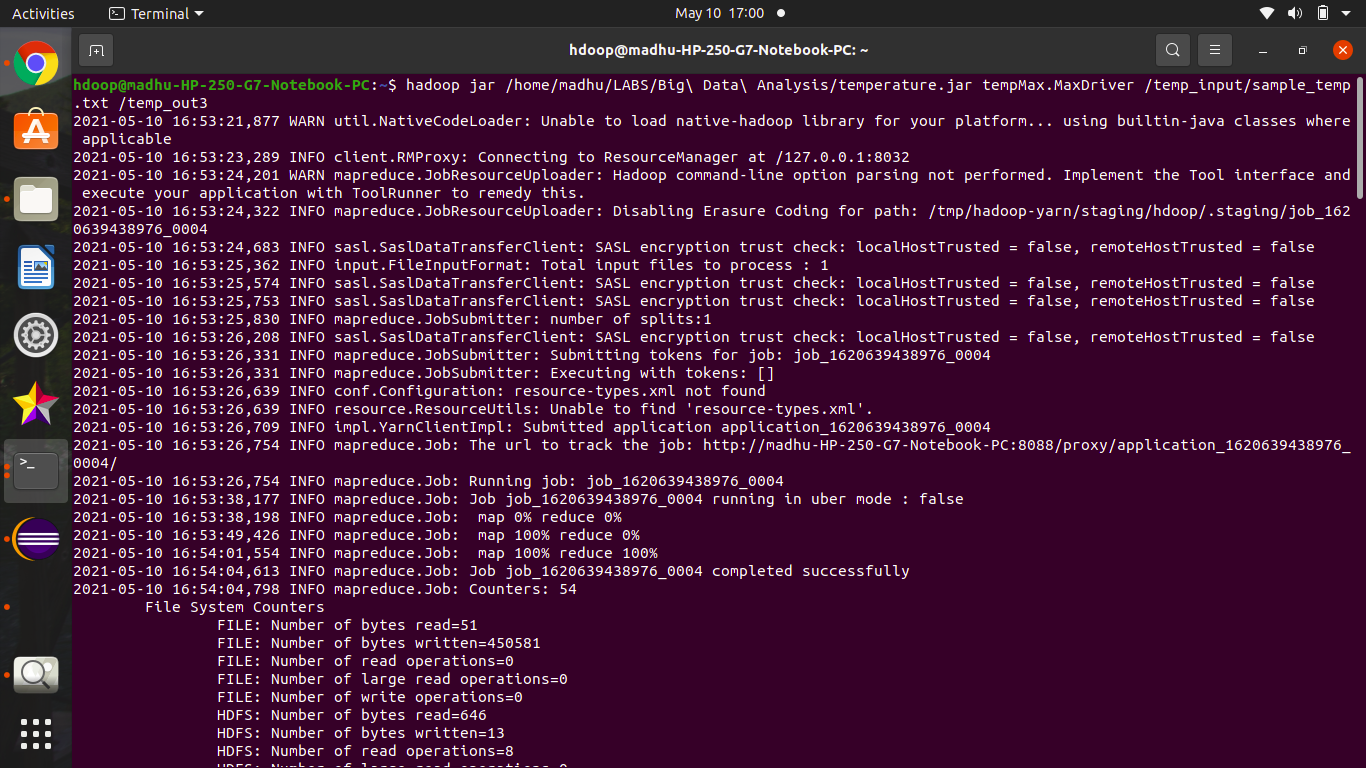
}

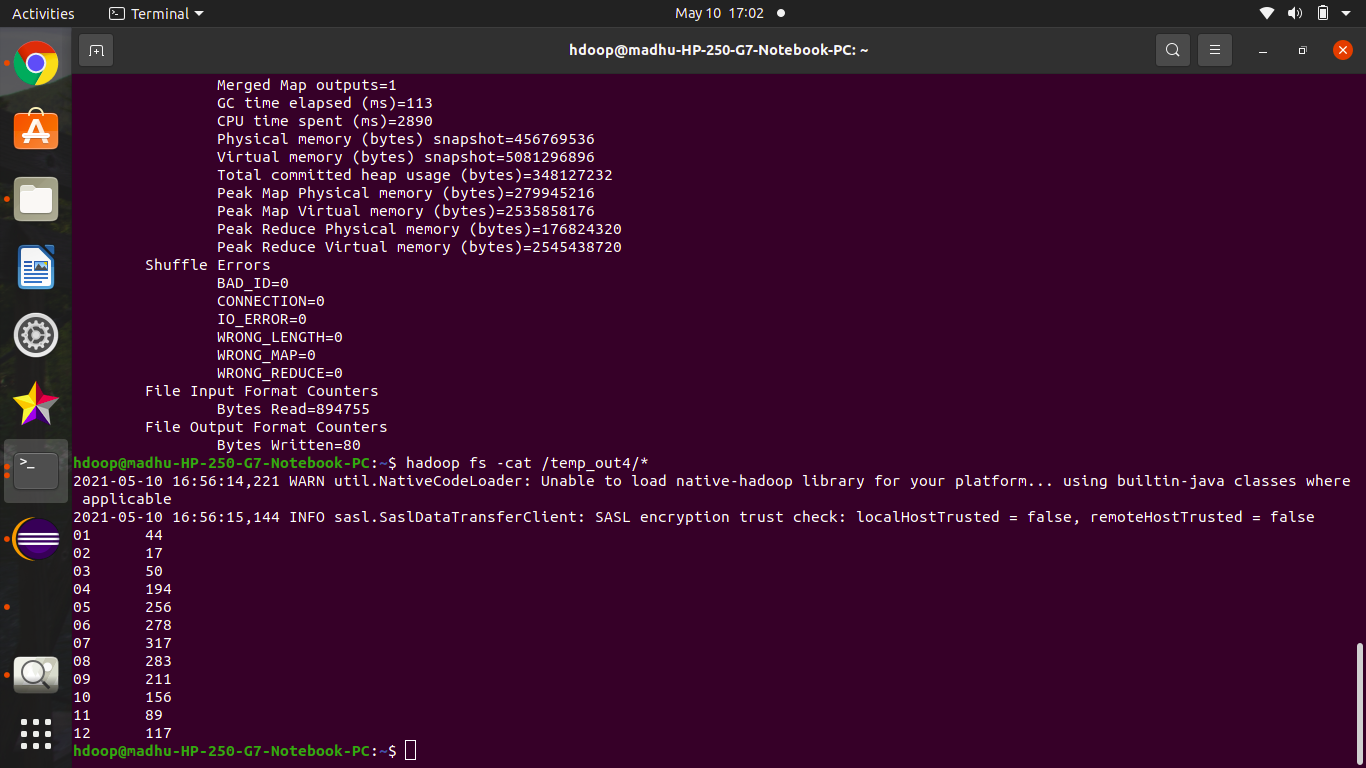
context.write(key, new IntWritable(max\_temp));

}

}

**Output:**





**PROGRAM – 7**

**FOR A GIVEN TEXT FILE, CREATE A MAP REDUCE PROGRAM TO SORT THE CONTENT IN AN ALPHABETIC ORDER LISTING ONLY TOP 10 MAXIMUM OCCURRENCES OF WORDS:**

**Java Files:**

TopN.java:

package sortwords;

import org.apache.hadoop.conf.Configuration;

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.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

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

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

import org.apache.hadoop.util.GenericOptionsParser;

import utils.MiscUtils;

import java.io.IOException;

import java.util.\*;

public class TopN {

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

Configuration conf = new Configuration();

String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();

if (otherArgs.length != 2) {

System.err.println("Usage: TopN <in> <out>");

System.exit(2);

}

Job job = Job.getInstance(conf);

job.setJobName("Top N");

job.setJarByClass(TopN.class);

job.setMapperClass(TopNMapper.class);

job.setReducerClass(TopNReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(otherArgs[0]));

FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {

private final static IntWritable one = new IntWritable(1);

private Text word = new Text();

private String tokens = "[\_|$#<>\\^=\\[\\]\\\*/\\\\,;,.\\-:()?!\"']";

@Override

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

String cleanLine = value.toString().toLowerCase().replaceAll(tokens, " ");

StringTokenizer itr = new StringTokenizer(cleanLine);

while (itr.hasMoreTokens()) {

word.set(itr.nextToken().trim());

context.write(word, one);

}

}

}

public static class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

private Map<Text, IntWritable> countMap = new HashMap<>();

@Override

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

int sum = 0;

for (IntWritable val : values) {

sum += val.get();

}

countMap.put(new Text(key), new IntWritable(sum));

}

@Override

protected void cleanup(Context context) throws IOException, InterruptedException {

Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(countMap);

int counter = 0;

for (Text key : sortedMap.keySet()) {

if (counter++ == 3) {

break;

}

context.write(key, sortedMap.get(key));

}

}

}

public static class TopNCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override

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

int sum = 0;

for (IntWritable val : values) {

sum += val.get();

}

context.write(key, new IntWritable(sum));

}

}

}

MiscUtils.java

package utils;

import java.util.\*;

public class MiscUtils {

public static <K extends Comparable, V extends Comparable> Map<K, V> sortByValues(Map<K, V> map) {

List<Map.Entry<K, V>> entries = new LinkedList<Map.Entry<K, V>>(map.entrySet());

Collections.sort(entries, new Comparator<Map.Entry<K, V>>() {

@Override

public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) {

return o2.getValue().compareTo(o1.getValue());

}

});

Map<K, V> sortedMap = new LinkedHashMap<K, V>();

for (Map.Entry<K, V> entry : entries) {

sortedMap.put(entry.getKey(), entry.getValue());

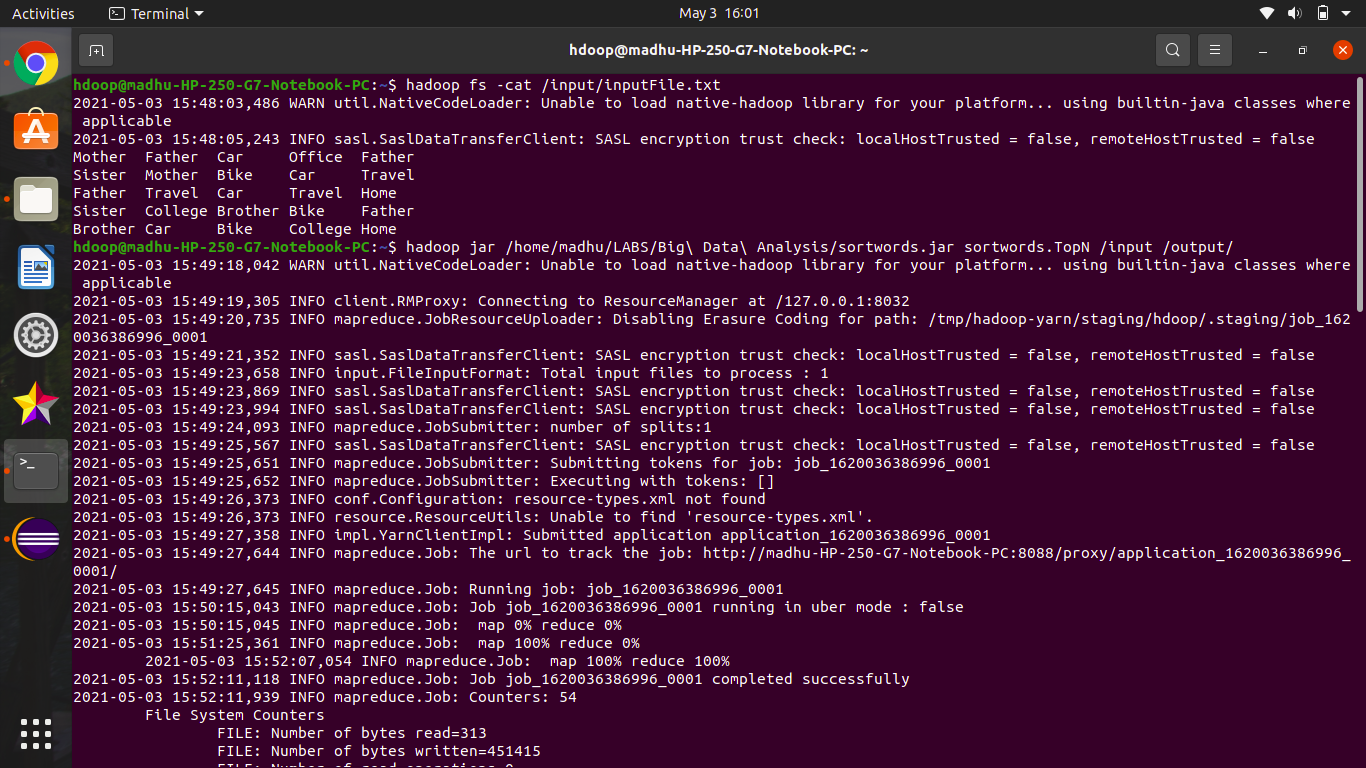
}

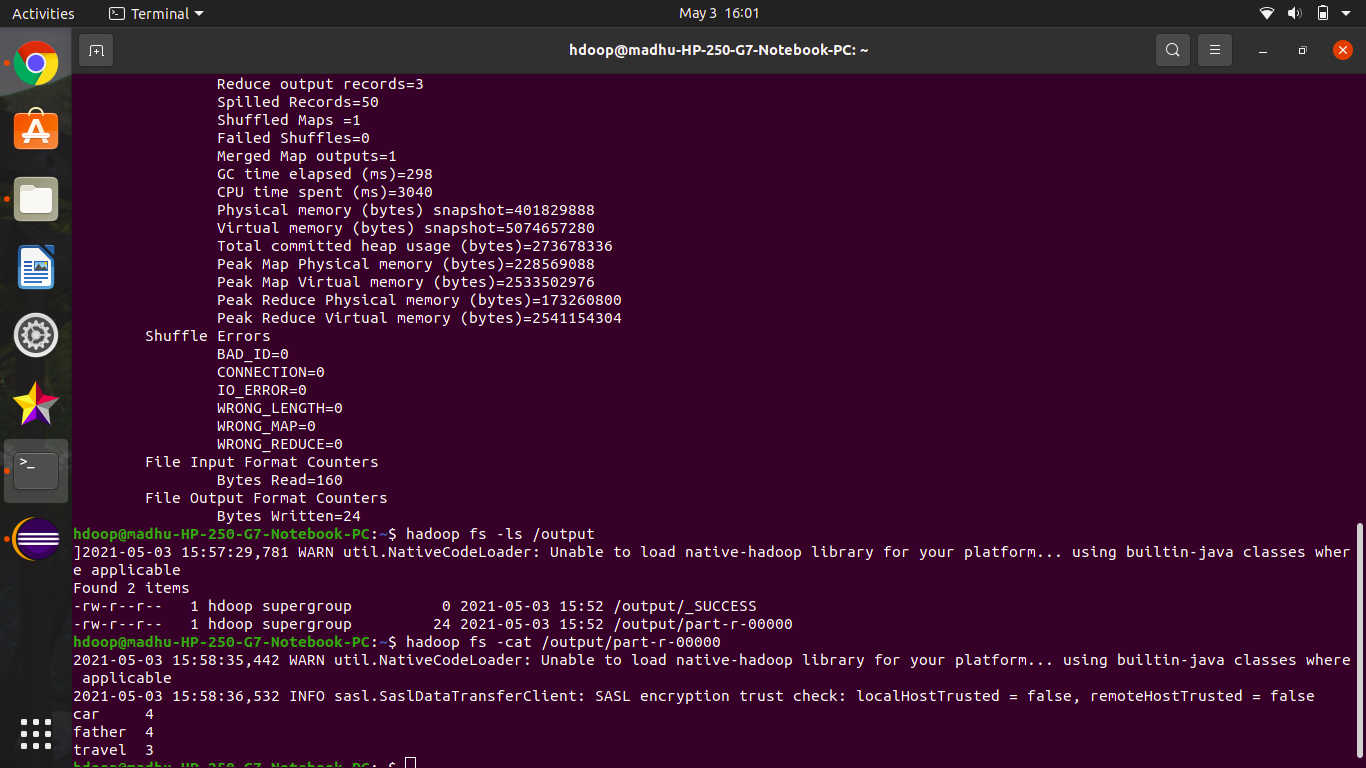
return sortedMap;

}

}

**Output:**





**PROGRAM – 8**

**CREATE A MAP REDUCE PROGRAM TO DEMONSTRATING JOIN OPERATION:**

**Java Files:**

[JoinDriver.java](https://github.com/mkgaganr/6thsem/blob/main/bdalab_1bm18cs049/join/JoinDriver.java)

package DatasetJoin;

import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.\*;

import org.apache.hadoop.mapred.lib.MultipleInputs;

import org.apache.hadoop.util.\*;

public class JoinDriver extends Configured implements Tool {

public static class KeyPartitioner implements Partitioner<TextPair, Text> {

@Override

public void configure(JobConf job) {}

@Override

public int getPartition(TextPair key, Text value, int numPartitions) {

return (key.getFirst().hashCode() & Integer.MAX\_VALUE) % numPartitions;

}

}

@Override

public int run(String[] args) throws Exception {

if (args.length != 3) {

System.out.println("Usage: <Department Emp Strength input> <Department Name input> <output>");

return -1;

}

JobConf conf = new JobConf(getConf(), getClass());

conf.setJobName("Join 'Department Emp Strength input' with 'Department Name input'");

Path AInputPath = new Path(args[0]);

Path BInputPath = new Path(args[1]);

Path outputPath = new Path(args[2]);

MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class, Posts.class);

MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class, Users.class);

FileOutputFormat.setOutputPath(conf, outputPath);

conf.setPartitionerClass(KeyPartitioner.class);

conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);

conf.setMapOutputKeyClass(TextPair.class);

conf.setReducerClass(JoinReducer.class);

conf.setOutputKeyClass(Text.class);

JobClient.runJob(conf);

return 0;

}

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

int exitCode = ToolRunner.run(new JoinDriver(), args);

System.exit(exitCode);

}

}

[JoinReducer.java](https://github.com/mkgaganr/6thsem/blob/main/bdalab_1bm18cs049/join/JoinReducer.java)

package DatasetJoin;

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.\*;

public class JoinReducer extends MapReduceBase implements Reducer<TextPair, Text, Text, Text> {

@Override

public void reduce (TextPair key, Iterator<Text> values, OutputCollector<Text, Text> output, Reporter reporter)

throws IOException

{

Text nodeId = new Text(values.next());

while (values.hasNext()) {

Text node = values.next();

Text outValue = new Text(nodeId.toString() + "\t\t" + node.toString());

output.collect(key.getFirst(), outValue);

}

}

}

[Posts.java](https://github.com/mkgaganr/6thsem/blob/main/bdalab_1bm18cs049/join/Posts.java)

package DatasetJoin;

import java.io.IOException;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapred.\*;

public class Posts extends MapReduceBase implements Mapper<LongWritable, Text, TextPair, Text> {

@Override

public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output, Reporter reporter)

throws IOException

{

String valueString = value.toString();

String[] SingleNodeData = valueString.split("\t");

output.collect(new TextPair(SingleNodeData[3], "0"), new Text(SingleNodeData[9]));

}

}

[Users.java](https://github.com/mkgaganr/6thsem/blob/main/bdalab_1bm18cs049/join/Users.java)

package DatasetJoin;

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FSDataInputStream;

import org.apache.hadoop.fs.FSDataOutputStream;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.\*;

import org.apache.hadoop.io.IntWritable;

public class Users extends MapReduceBase implements Mapper<LongWritable, Text, TextPair, Text> {

@Override

public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output, Reporter reporter)

throws IOException

{

String valueString = value.toString();

String[] SingleNodeData = valueString.split("\t");

output.collect(new TextPair(SingleNodeData[0], "1"), new Text(SingleNodeData[1]));

}

}

[TextPair.java](https://github.com/mkgaganr/6thsem/blob/main/bdalab_1bm18cs049/join/TextPair.java)

package DatasetJoin;

import java.io.\*;

import org.apache.hadoop.io.\*;

public class TextPair implements WritableComparable<TextPair> {

private Text first;

private Text second;

public TextPair() {

set(new Text(), new Text());

}

public TextPair(String first, String second) {

set(new Text(first), new Text(second));

}

public TextPair(Text first, Text second) {

set(first, second);

}

public void set(Text first, Text second) {

this.first = first;

this.second = second;

}

public Text getFirst() {

return first;

}

public Text getSecond() {

return second;

}

@Override

public void write(DataOutput out) throws IOException {

first.write(out);

second.write(out);

}

@Override

public void readFields(DataInput in) throws IOException {

first.readFields(in);

second.readFields(in);

}

@Override

public int hashCode() {

return first.hashCode() \* 163 + second.hashCode();

}

@Override

public boolean equals(Object o) {

if (o instanceof TextPair) {

TextPair tp = (TextPair) o;

return first.equals(tp.first) && second.equals(tp.second);

}

return false;

}

@Override

public String toString() {

return first + "\t" + second;

}

@Override

public int compareTo(TextPair tp) {

int cmp = first.compareTo(tp.first);

if (cmp != 0) {

return cmp;

}

return second.compareTo(tp.second);

}

public static class Comparator extends WritableComparator {

private static final Text.Comparator TEXT\_COMPARATOR = new Text.Comparator();

public Comparator() {

super(TextPair.class);

}

@Override

public int compare(byte[] b1, int s1, int l1, byte[] b2, int s2, int l2) {

try {

int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);

int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);

int cmp = TEXT\_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);

if (cmp != 0) {

return cmp;

}

return TEXT\_COMPARATOR.compare(b1, s1 + firstL1, l1 - firstL1,

b2, s2 + firstL2, l2 - firstL2);

} catch (IOException e) {

throw new IllegalArgumentException(e);

}

}

}

static {

WritableComparator.define(TextPair.class, new Comparator());

}

public static class FirstComparator extends WritableComparator {

private static final Text.Comparator TEXT\_COMPARATOR = new Text.Comparator();

public FirstComparator() {

super(TextPair.class);

}

@Override

public int compare(byte[] b1, int s1, int l1, byte[] b2, int s2, int l2) {

try {

int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);

int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);

return TEXT\_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);

} catch (IOException e) {

throw new IllegalArgumentException(e);

}

}

@Override

public int compare(WritableComparable a, WritableComparable b) {

if (a instanceof TextPair && b instanceof TextPair) {

return ((TextPair) a).first.compareTo(((TextPair) b).first);

}

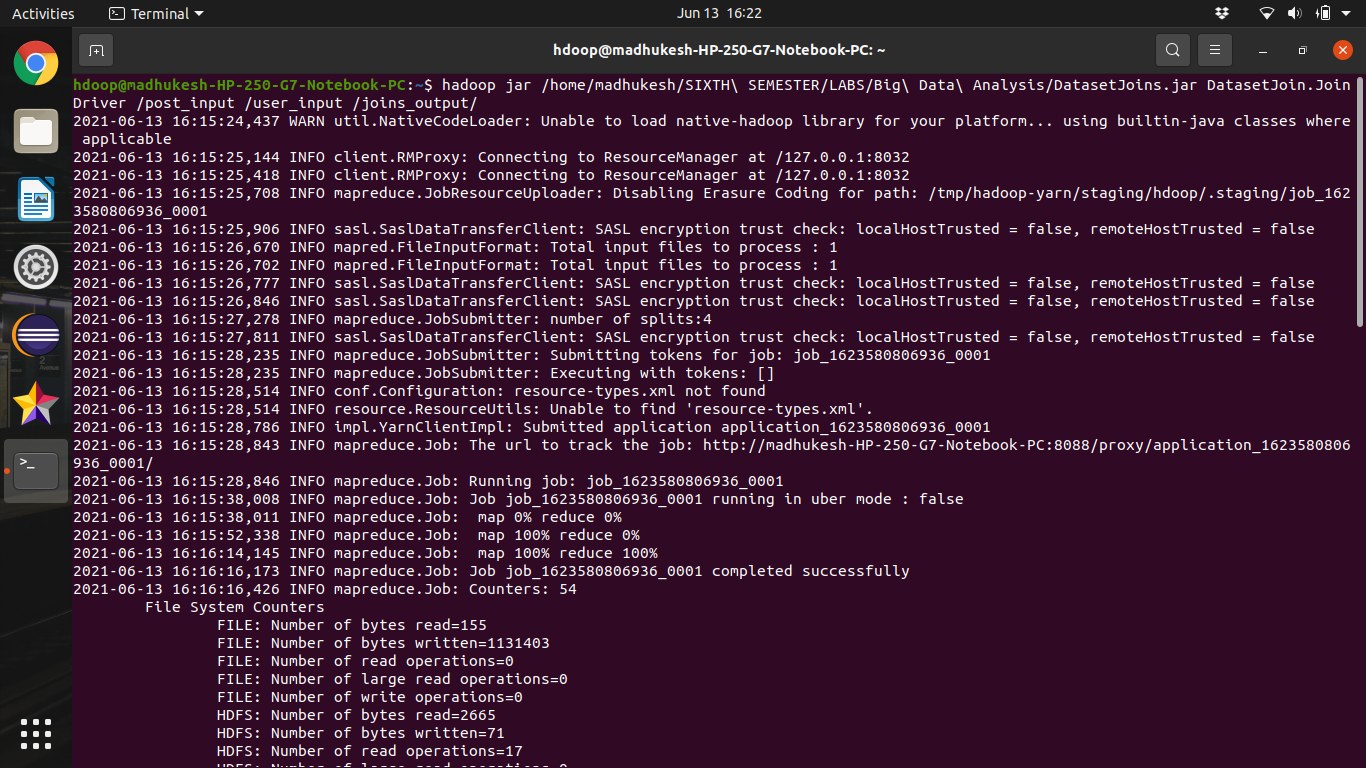
return super.compare(a, b);

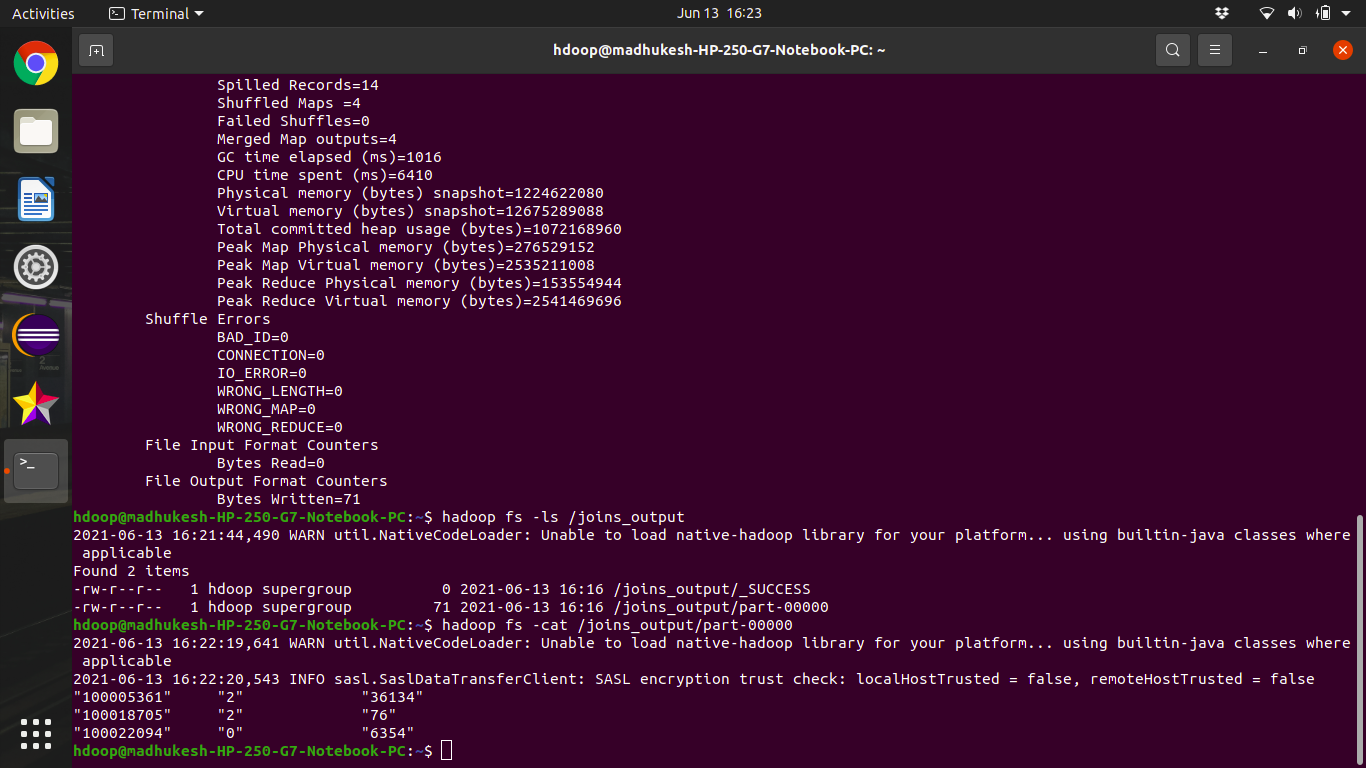
}

}

}

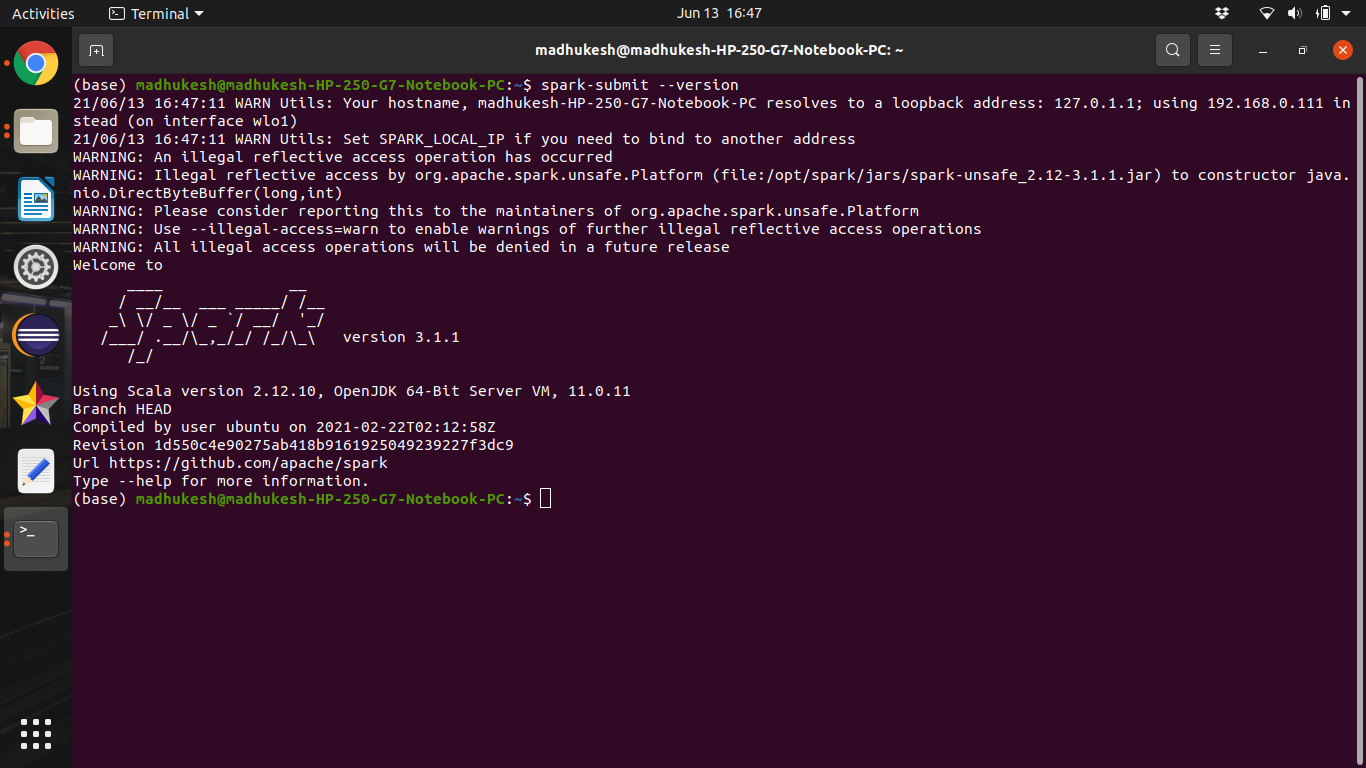
**Output :**

****

****

**PROGRAM – 9**

**SCREENSHOT OF SPARK INSTALLED:**

****

**PROGRAM – 10**

**USING RDD AND FLAMAP COUNT HOW MANY TIMES EACH WORD APPEARS IN A FILE AND WRITE OUT A LIST OF WORDS WHOSE COUNT IS STRICTLY GREATER THAN 4 USING SPARK:**

**Input:**

Mother Father Car Office Father

Sister Mother Bike Car Travel

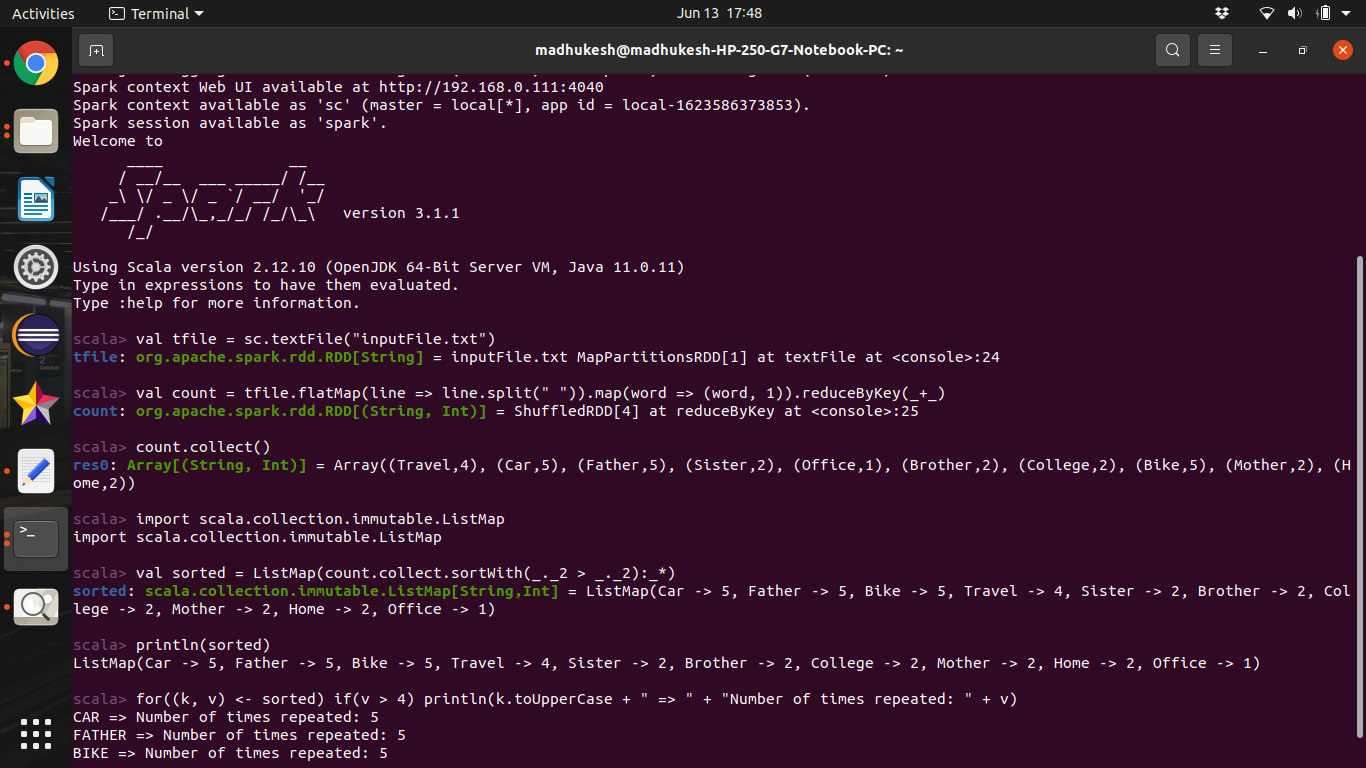
Father Travel Car Travel Home

Sister College Brother Bike Father

Brother Car Bike College Home

Car Father Travel Bike Bike

**Output:**

****