VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS

Submitted by

Tanishq Deshpande (1BM19CS213)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
May-2022 to July-2022

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "LAB COURSE **BIG DATA ANALYTICS '' carried** out by **Tanishq Deshpande (1BM19CS213),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics - (20CS6PEBDA)** work prescribed for the said degree.

Dr. Pallavi G.B	**Dr. Jyothi S Nayak**
Assistant Professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

Index Sheet

Course Outcome

CO1	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO2	Analyze the Big Data and obtain insight using data analytics mechanisms.
CO3	Design and implement Big data applications by applying NoSQL, Hadoop or Spark

PROGRAM 1

Perform the following DB operations using Cassandra.

1. Create a keyspace by name Employee

2. Create a column family by name

Employee-Info with attributes

Emp_Id Primary Key, Emp_Name,

Designation, Date_of_Joining, Salary, Dept_Name

3. Insert the values into the table in batch

4. Update Employee name and Department of Emp-Id 121

5. Sort the details of Employee records based on salary

6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

7. Update the altered table to add project names.

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

```
cqlsh:employee> CREATE KEYSPACE employee WITH REPLICATION={ 'class' :
'SimpleStrategy',

'replication_factor' : 1};

cqlsh:employee> USE employee;

cqlsh:employee> create table employee_info(emp_id int PRIMARY KEY, emp_name text,

 ... designation text, date_of_joining timestamp, salary double PRIMARY KEY, dept_name text);

cqlsh:employee> CREATE TABLE employee_info(emp_id int, emp_name text, designation text,
date_of_joining

timestamp, salary double, dept_name text, PRIMARY KEY(emp_id, salary));

cqlsh:employee> BEGIN BATCH INSERT INTO

 ... employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)

 ... VALUES(100,'John','MANAGER','2021-09-11',30000,'TESTING');

 ... INSERT INTO

 ... employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)

 ... VALUES(111,'Tom','ASSOCIATE','2021-06-22',25000,'DEVELOPING');
 ... INSERT INTO
 ... employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
 ... VALUES(121,'Elsa','MANAGER','2021-03-30',35000,'HR');
 ... INSERT INTO
 ... employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
 ... VALUES(115,'Chris','ASSISTANT','2021-12-30',20000,'DEVELOPING');
 ... INSERT INTO
 ... employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
 ... VALUES(105,'Sarah','ASSOCIATE','2021-06-25',25000,'TESTING');
 ... APPLY BATCH;
cqlsh:employee> SELECT * FROM employee_info
 ... ;
emp_id | salary | date_of_joining | dept_name | designation | emp_name
--------+--------+---------------------------------+------------+-------------+----------
 105 | 25000 | 2021-06-24 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sarah
 111 | 25000 | 2021-06-21 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Tom
 121 | 35000 | 2021-03-29 18:30:00.000000+0000 | HR | MANAGER | Elsa
 115 | 20000 | 2021-12-29 18:30:00.000000+0000 | DEVELOPING | ASSISTANT | Chris
 100 | 30000 | 2021-09-10 18:30:00.000000+0000 | TESTING | MANAGER | John
(5 rows)
cqlsh:employee> UPDATE employee_info SET emp_name = 'Jessica', dept_name =
'DEVELOPING' WHERE
emp_id = 121;
cqlsh:employee> UPDATE employee_info SET emp_name = 'Jessica', dept_name =
'DEVELOPING' WHERE
emp_id = 121 AND salary = 35000;
cqlsh:employee> SELECT * FROM employee_info ;
emp_id | salary | date_of_joining | dept_name | designation | emp_name
--------+--------+---------------------------------+------------+-------------+----------
```

```
 105 | 25000 | 2021-06-24 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sarah

 111 | 25000 | 2021-06-21 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Tom

 121 | 35000 | 2021-03-29 18:30:00.000000+0000 | DEVELOPING | MANAGER | Jessica

 115 | 20000 | 2021-12-29 18:30:00.000000+0000 | DEVELOPING | ASSISTANT | Chris

 100 | 30000 | 2021-09-10 18:30:00.000000+0000 | TESTING | MANAGER | John

(5 rows)

cqlsh:employee> SELECT * FROM employee_info WHERE emp_id in (105, 111, 121, 115,
100) order by salary;

cqlsh:employee> paging off

Disabled Query paging.

cqlsh:employee> SELECT * FROM employee_info WHERE emp_id in (105, 111, 121, 115,
100) order by salary;

emp_id | salary | date_of_joining | dept_name | designation | emp_name

--------+--------+---------------------------------+------------+-------------+----------

 115 | 20000 | 2021-12-29 18:30:00.000000+0000 | DEVELOPING | ASSISTANT | Chris

 105 | 25000 | 2021-06-24 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sarah

 111 | 25000 | 2021-06-21 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Tom

 100 | 30000 | 2021-09-10 18:30:00.000000+0000 | TESTING | MANAGER | John

 121 | 35000 | 2021-03-29 18:30:00.000000+0000 | DEVELOPING | MANAGER | Jessica

(5 rows)

cqlsh:employee> ALTER TABLE employee_info ADD projects text;

cqlsh:employee> UPDATE employee_info SET projects = 'Chat App' WHERE emp_id = 111;

cqlsh:employee> UPDATE employee_info SET projects = 'Chat App' WHERE emp_id = 111
and salary = 25000;

cqlsh:employee> UPDATE employee_info SET projects = 'Discord Bot' WHERE emp_id = 115
and salary =

20000;

cqlsh:employee> UPDATE employee_info SET projects = 'Campus Portal' WHERE emp_id =
105 and salary =

25000;
```

```
cqlsh:employee> UPDATE employee_info SET projects = 'YouTube Downloader' WHERE
emp_id = 100 and

salary = 30000;

cqlsh:employee> UPDATE employee_info SET projects = 'Library Management System '
WHERE emp_id = 121

and salary = 35000;

cqlsh:employee> SELECT * FROM employee_infor

 ... ;

cqlsh:employee> SELECT * FROM employee_info ;

emp_id | salary | date_of_joining | dept_name | designation | emp_name | projects

--------+--------+---------------------------------+------------+-------------+----------+----------------------
- - - -

 105 | 25000 | 2021-06-24 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sarah | Campus

Portal

 111 | 25000 | 2021-06-21 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Tom | Chat

App

 121 | 35000 | 2021-03-29 18:30:00.000000+0000 | DEVELOPING | MANAGER | Jessica |
Library

Management System

 115 | 20000 | 2021-12-29 18:30:00.000000+0000 | DEVELOPING | ASSISTANT | Chris |
Discord

Bot

 100 | 30000 | 2021-09-10 18:30:00.000000+0000 | TESTING | MANAGER | John | YouTube

Downloader

(5 rows)

cqlsh:employee> INSERT INTO

 ... employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)

 ...

 ... ;

cqlsh:employee> INSERT INTO
```

```
 ... employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
 ... VALUES(110,'SAM','ASSOCIATE','2021-01-11',28000,'TESTING') USING TTL 15;
cqlsh:employee> SELECT TTL(emp_name) from employee_info WHERE emp_id = 110;
ttl(emp_name)
- - - - - - - - - -
 3
(1 rows)
cqlsh:employee> SELECT * FROM employee_info;
emp_id | salary | date_of_joining | dept_name | designation | emp_name | projects
--------+--------+---------------------------------+------------+-------------+----------+----------------------
- - - -
 105 | 25000 | 2021-06-24 18:30:00.000000+0000 | TESTING | ASSOCIATE | Sarah | Campus
Portal
 111 | 25000 | 2021-06-21 18:30:00.000000+0000 | DEVELOPING | ASSOCIATE | Tom | Chat
App
 121 | 35000 | 2021-03-29 18:30:00.000000+0000 | DEVELOPING | MANAGER | Jessica |
Library
Management System
 115 | 20000 | 2021-12-29 18:30:00.000000+0000 | DEVELOPING | ASSISTANT | Chris |
Discord
Bot
 100 | 30000 | 2021-09-10 18:30:00.000000+0000 | TESTING | MANAGER | John | YouTube
Downloader
(5 rows)
```

PROGRAM 2

Perform the following DB operations using Cassandra.

1.Create a keyspace by name Library

2. 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

3. Insert the values into the table in batch

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

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

6. Export the created column to a csv file

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

```
cqlsh:library> CREATE KEYSPACE library WITH replication = {'class':
'SimpleStrategy','replication_factor':1};

cqlsh:library> USE library ;

cqlsh:library> CREATE TABLE Library_info(stud_id int, stud_name text, book_name text,
book_id text,

date_of_issue timestamp, counter_value counter, PRIMARY KEY(stud_id,stud_name,
book_name, book_id,

date_of_issue));

cqlsh:library> BEGIN COUNTER BATCH

 ... UPDATE library_info set counter_value +=1 where stud_id = 111 and stud_name = 'Manoj'
and

book_name = 'Operations Research' and book_id = '56TXT' and date_of_issue = '2021-09-12';

 ... UPDATE library_info set counter_value +=1 where stud_id = 112 and stud_name = 'Kamal'
and

book_name = 'Engineering Mathematics-3' and book_id = '5ERW4' and date_of_issue =
'2021-04-10';

 ... UPDATE library_info set counter_value +=1 where stud_id = 113 and stud_name = 'Mahesh'
and

book_name = 'Robinson Crusoe' and book_id = '34EDC' and date_of_issue = '2021-02-01';

 ... UPDATE library_info set counter_value +=1 where stud_id = 114 and stud_name = 'Raj' and
```

```
book_name = 'Engineering Drawing' and book_id = '123ER' and date_of_issue = '2021-04-03';

 ... APPLY BATCH;

cqlsh:library> SELECT * FROM library_info ;

stud_id | stud_name | book_name | book_id | date_of_issue | counter_value

---------+-----------+--------------------------+---------+--------------------------------+---------------

 114 | Raj | Engineering Drawing | 123ER | 2021-04-02 18:30:00.000000+0000 | 1

 111 | Manoj | Operations Research | 56TXT | 2021-09-11 18:30:00.000000+0000 | 1

 113 | Mahesh | Robinson Crusoe | 34EDC | 2021-01-31 18:30:00.000000+0000 | 1

 112 | Kamal | Engineering Mathematics-3 | 5ERW4 | 2021-04-09 18:30:00.000000+0000 | 1

(4 rows)

cqlsh:library> UPDATE library_info set counter_value += 1 where stud_id = 112 and stud_name
= 'Kamal' and

book_name = 'Engineering Mathematics-3' and book_id = '5ERW4' and date_of_issue =
'2021-04-09';

cqlsh:library> SELECT * FROM library_info ;

stud_id | stud_name | book_name | book_id | date_of_issue | counter_value

---------+-----------+--------------------------+---------+--------------------------------+---------------

 114 | Raj | Engineering Drawing | 123ER | 2021-04-02 18:30:00.000000+0000 | 1

 111 | Manoj | Operations Research | 56TXT | 2021-09-11 18:30:00.000000+0000 | 1

 113 | Mahesh | Robinson Crusoe | 34EDC | 2021-01-31 18:30:00.000000+0000 | 1

 112 | Kamal | Engineering Mathematics-3 | 5ERW4 | 2021-04-09 18:30:00.000000+0000 | 2

cqlsh:library> copy library_info(stud_id,stud_name, book_name, book_id,
date_of_issue,counter_value) to

'library_info.csv' ;

Using 11 child processes

Starting copy of library.library_info with columns [stud_id, stud_name, book_name, book_id,
date_of_issue,

counter_value].

Processed: 6 rows; Rate: 39 rows/s; Avg. rate: 39 rows/s

6 rows exported to 1 files in 0.165 seconds.
```

```
cqlsh:library> copy library_info(stud_id,stud_name, book_name, book_id,
date_of_issue,counter_value) from

'library_info.csv' ;

Using 11 child processes

Starting copy of library.library_info with columns [stud_id, stud_name, book_name, book_id,
date_of_issue,

counter_value].

Processed: 6 rows; Rate: 10 rows/s; Avg. rate: 15 rows/s

6 rows imported from 1 files in 0.392 seconds (0 skipped).
```

PROGRAM 3

MongoDB CREATE DATABASE IN MONGODB.

```
> use myDB
switched to db myDB
db;
myDB
show dbs;
admin 0.000GB
config 0.000GB
local 0.000GB
```

II. CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS

1. To create a collection by the name “Student”. Let us take a look at the collection list prior to the creation of the new collection “Student”.

```
db.createCollection(“Student”); => sql equivalent
CREATE TABLE STUDENT(…);
{ "ok" : 1 }
```

2.To drop a collection by the name “Student”.

```
db.Student.drop();
```

3.Create a collection by the name

“Students” and store the following data in it.

```
db.Student.insert({_id:1,StudName:"MichelleJacintha",Gra
de:"VII",Hobbies:"InternetSurfing"});
WriteResult({ "nInserted" : 1 })
```

4.Insert the document for “AryanDavid” in to the Students

collection only if it does not already exist in the collection.

However, if it is already present in the collection, then update the

document with new values. (Update his Hobbies from “Skating” to

“Chess”.) Use “Update else insert” (if there is an existing document, it will attempt to update it, if there is no existing document then it will insert it).

```
db.Student.update({_id:3,StudName:"AryanDavid",Grad
e:" VII"},{$set:{Hobbies:"Skating"}},{upsert:true});
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified"
: 0, "_id" : 3 })
```

5.FIND METHOD

A. To search for documents from the “Students” collection based on certain search criteria.

```
db.Student.find({StudName:"AryanDavid"});
({cond..},{columns.. column:1, columnname:0} )
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid",
"Hobbies" : "Skating" }
```

B. To display only the StudName and Grade from all the documents of the Students collection. The identifier_id should be suppressed and NOT displayed.

```
db.Student.find({},{StudName:1,Grade:1,_id:0});
{ "StudName" : "MichelleJacintha", "Grade" : "VII" }
{ "Grade" : "VII", "StudName" : "AryanDavid" }
```

C. To find those documents where the Grade is set to ‘VII’

```
db.Student.find({Grade:{$eq:'VII'}}).pretty();
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
```

```
{
"_id" : 3,
"Grade" : "VII",
"StudName" : "AryanDavid",
"Hobbies" : "Skating"
}
```

D. To find those documents from the Students collection where the Hobbies is set to either 'Chess' or is set to 'Skating'.

```
db.Student.find({Hobbies :{ $in: ['Chess','Skating']}}).pretty
();
{
"_id" : 3,
"Grade" : "VII",
"StudName" : "AryanDavid",
"Hobbies" : "Skating"
}
```

E. To find documents from the Students collection where the StudName begins with "M".

```
db.Student.find({StudName:/^M/}).pretty();
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
```

F. To find documents from the Students collection where the StudNamehas an "e" in any position.

```
db.Student.find({StudName:/e/}).pretty();
```

```
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
```

G. To find the number of documents in the Students collection.

```
db.Student.count();
2
```

H. To sort the documents from the Students collection in the descending order of StudName.

```
db.Student.find().sort({StudName:-1}).pretty();
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
{
"_id" : 3,
"Grade" : "VII",
"StudName" : "AryanDavid",
"Hobbies" : "Skating"
}
```

III. Import data from a CSV file

Given a CSV file "sample.txt" in the D:drive, import the file into the MongoDB collection, "SampleJSON". The collection is in the database "test".

mongoimport --db Student --collection airlines --type csv –

headerline --file /home/hduser/Desktop/airline.csv

IV. Export data to a CSV file

mongoexport --host localhost --db Student --collection

airlines --csv --out /home/hduser/Desktop/output.txt –

fields “Year”

,

”Quarter”

V. Save Method :.

db.Student.save({StudName:”Vamsi”, Grade:”VI”})

WriteResult({ "nInserted" : 1 })

VI. Add a new field to existing Document:

db.Student.update({_id:ObjectId("625695cc7d129fb98b44c8a1")},

{$set:{Location:"Network"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

VII. Remove the field in an existing Document

db.Student.update({_id:ObjectId("625695cc7d129fb98b44c8a1

")},

{$unset:{Location:”Network”}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

VIII. Finding Document based on search criteria suppressing few fields

db.Student.find({_id:1},{StudName:1,Grade:1,_id:0});

{ "StudName" : "MichelleJacintha", "Grade" : "VII" }

To find those documents where the Grade is not set to ‘VII’

db.Student.find({Grade:{$ne:'VII'}}).pretty();

{

"_id" : ObjectId("625695cc7d129fb98b44c8a1"),

```
"StudName" : "Vamsi",
"Grade" : "VI"
}
```

To find documents from the Students collection where the StudName ends with s.

```
db.Student.find({StudName:/s$/}).pretty();
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
```

IX. to set a particular field value to NULL

```
db.Student.update({_id:3},{$set:{Location:null}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

X. Count the number of documents in Student Collections

```
db.Student.count()
3
```

XI. Count the number of documents in Student Collections with grade :VII

```
db.Student.count({Grade:"VII"})
```

2 retrieve first 3 documents

```
db.Student.find({Grade:"VII"}).limit(1).pretty();
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
```

Sort the document in Ascending order

```
db.Student.find().sort({StudName:1}).pretty();
{
"_id" : 3,
"Grade" : "VII",
"StudName" : "AryanDavid",
"Hobbies" : "Skating",
"Location" : null
}
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
{
"_id" : ObjectId("625695cc7d129fb98b44c8a1"),
"StudName" : "Vamsi",
"Grade" : "VI"
}
```

Note: for desending order :

```
db.Students.find().sort({StudName:-
1}).pretty();
```

To Skip the 1st two documents from the Students Collections

```
db.Student.find().skip(2).pretty()
{
"_id" : ObjectId("625695cc7d129fb98b44c8a1"),
```

"StudName" : "Vamsi",

"Grade" : "VI"

}

XII. Create a collection by name "food" and add to each document add a "fruits" array

db.food.insert({ _id:1, fruits:['grapes','mango','apple'] })

db.food.insert({ _id:2, fruits:['grapes','mango','cherry'] })

db.food.insert({ _id:3, fruits:['banana','mango'] })

{ "_id" : 1, "fruits" : ["grapes", "mango", "apple"] }

{ "_id" : 2, "fruits" : ["grapes", "mango", "cherry"] }

{ "_id" : 3, "fruits" : ["banana", "mango"] }

To find those documents from the "food" collection which has the "fruits array" consists of "grapes" ,"mango" and "apple".

db.food.find ({fruits: ['grapes','mango','apple'] }). pretty();

{ "_id" : 1, "fruits" : ["grapes", "mango", "apple"] }

To find in "fruits" array having "mango" in the first index

position.

db.food.find ({"fruits.1":grapes'})

To find those documents from the "food" collection where the size of the array is two.

db.food.find ({"fruits": {$size:2}})

{ "_id" : 3, "fruits" : ["banana", "mango"] }

To find the document with a particular id and display the first two elements from the array "fruits"

db.food.find({_id:1},{"fruits":{$slice:2}})

{ "_id" : 1, "fruits" : ["grapes", "mango"] }

To find all the documets from the food collection which have elements mango and grapes in the array "fruits"

db.food.find({fruits:{$all:["mango","grapes"]}})

{ "_id" : 1, "fruits" : ["grapes", "mango", "apple"] }

{ "_id" : 2, "fruits" : ["grapes", "mango", "cherry"] }

Update on Array: using particular id replace the element present in the 1st index position of the fruits array with apple

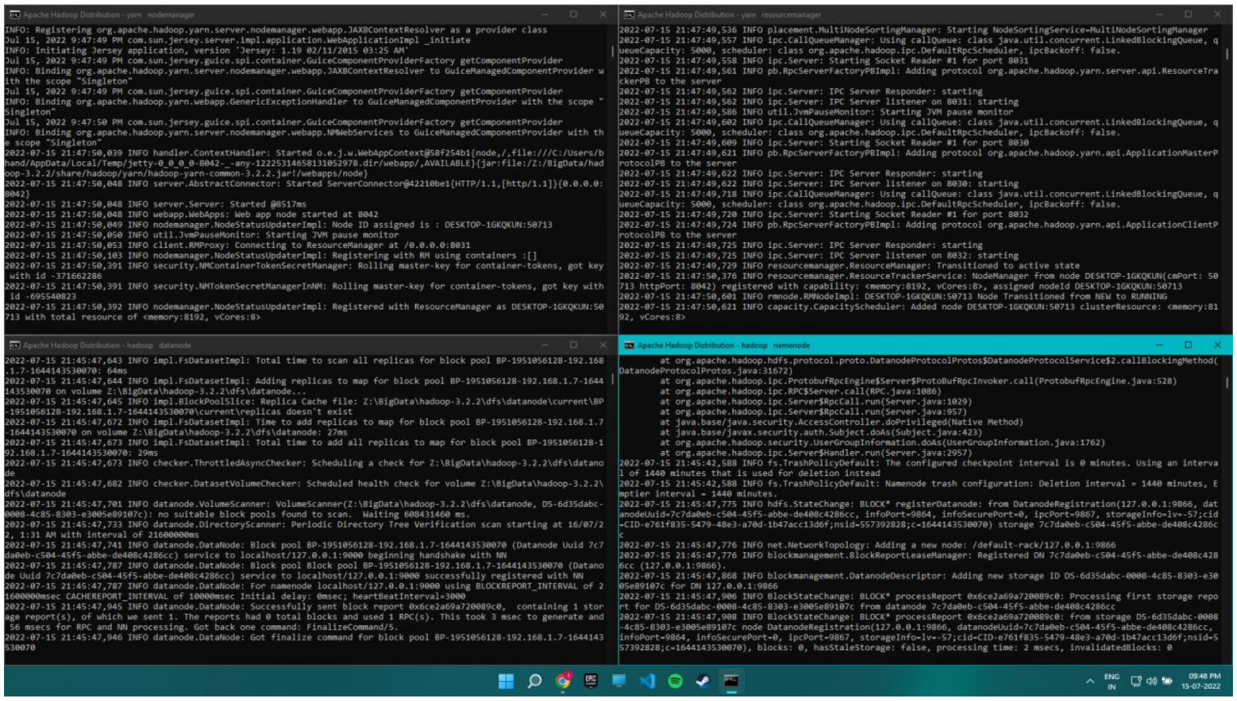
```
db.food.update({_id:3},{$set:{'fruits.1':'apple'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Insert new key value pairs in the fruits array

```
db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherr
y:100}}})
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ {
"grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
```

PROGRAM 4

4. Screenshot of Hadoop installed

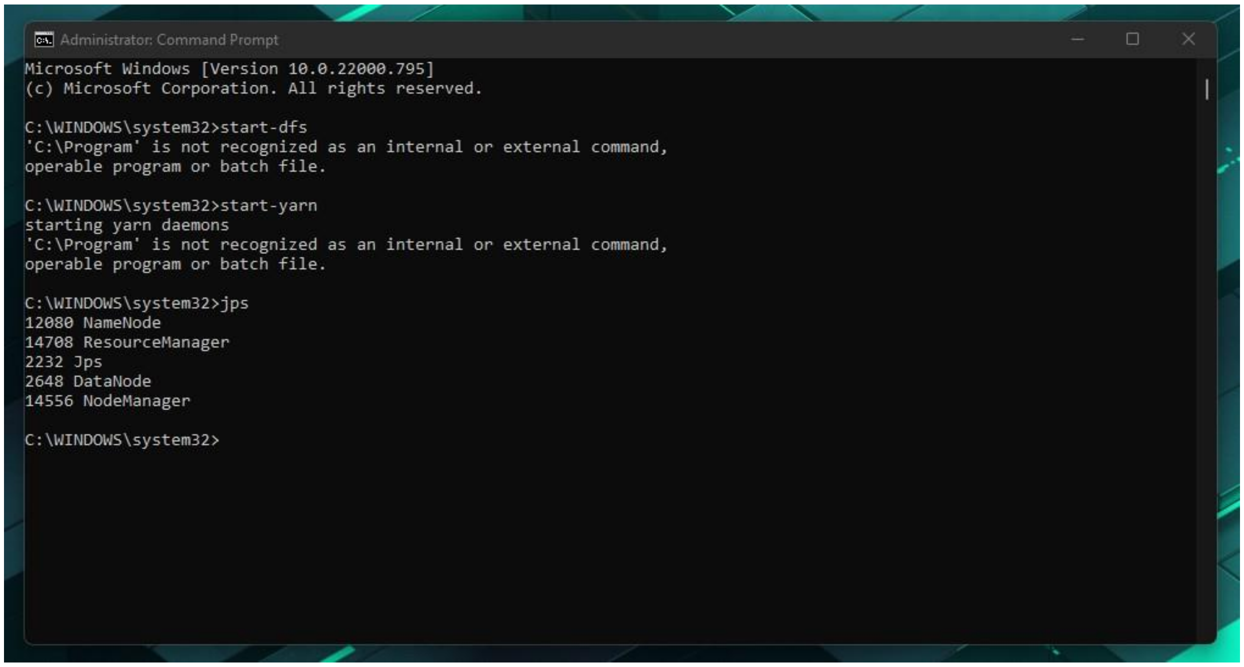

Apache Hadoop Distribution - yarn nodemanager
Apache Hadoop Distribution - yarn resourcemanager
Apache Hadoop Distribution - hadoop datanode
Apache Hadoop Distribution - hadoop namenode
ENG
IN
09:48 PM
15-07-2022

```
INFO: Registering org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver as a provider class
Jul 15, 2022 9:47:49 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.19 02/11/2015 03:25 AM'
Jul 15, 2022 9:47:49 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver to GuiceManagedComponentProvider w
ith the scope "Singleton"
Jul 15, 2022 9:47:49 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "
Singleton"
Jul 15, 2022 9:47:50 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with th
e scope "Singleton"
2022-07-15 21:47:50,039 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@58f254b1{node,/,file:///C:/Users/b
hand/AppData/Local/Temp/jetty-0_0_0_0-8042-_-any-12225314658131052978.dir/webapp/,AVAILABLE}{jar:file:/Z:/BigData/had
oop-3.2.2/share/hadoop/yarn/hadoop-yarn-common-3.2.2.jar!/webapps/node}
2022-07-15 21:47:50,048 INFO server.AbstractConnector: Started ServerConnector@42210be1{HTTP/1.1,[http/1.1]}{0.0.0.0:
8042}
2022-07-15 21:47:50,048 INFO server.Server: Started @8517ms
2022-07-15 21:47:50,048 INFO webapp.WebApps: Web app node started at 8042
2022-07-15 21:47:50,049 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : DESKTOP-1GKQKUN:50713
2022-07-15 21:47:50,050 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2022-07-15 21:47:50,053 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8031
2022-07-15 21:47:50,103 INFO nodemanager.NodeStatusUpdaterImpl: Registering with RM using containers :[]
2022-07-15 21:47:50,391 INFO security.NMContainerTokenSecretManager: Rolling master-key for container-tokens, got key
 with id -371662286
2022-07-15 21:47:50,391 INFO security.NMTokenSecretManagerInNM: Rolling master-key for container-tokens, got key with
 id -695540823
2022-07-15 21:47:50,392 INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceManager as DESKTOP-1GKQKUN:50
713 with total resource of <memory:8192, vCores:8>
```

```
2022-07-15 21:47:49,536 INFO placement.MultiNodeSortingManager: Starting NodeSortingService=MultiNodeSortingManager
2022-07-15 21:47:49,557 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, q
ueueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2022-07-15 21:47:49,558 INFO ipc.Server: Starting Socket Reader #1 for port 8031
2022-07-15 21:47:49,561 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.server.api.ResourceTra
ckerPB to the server
2022-07-15 21:47:49,562 INFO ipc.Server: IPC Server Responder: starting
2022-07-15 21:47:49,562 INFO ipc.Server: IPC Server listener on 8031: starting
2022-07-15 21:47:49,586 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2022-07-15 21:47:49,602 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, q
ueueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2022-07-15 21:47:49,609 INFO ipc.Server: Starting Socket Reader #1 for port 8030
2022-07-15 21:47:49,621 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMasterP
rotocolPB to the server
2022-07-15 21:47:49,622 INFO ipc.Server: IPC Server Responder: starting
2022-07-15 21:47:49,622 INFO ipc.Server: IPC Server listener on 8030: starting
2022-07-15 21:47:49,718 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, q
ueueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2022-07-15 21:47:49,720 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2022-07-15 21:47:49,724 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClientP
rotocolPB to the server
2022-07-15 21:47:49,725 INFO ipc.Server: IPC Server Responder: starting
2022-07-15 21:47:49,725 INFO ipc.Server: IPC Server listener on 8032: starting
2022-07-15 21:47:49,729 INFO resourcemanager.ResourceManager: Transitioned to active state
2022-07-15 21:47:50,376 INFO resourcemanager.ResourceTrackerService: NodeManager from node DESKTOP-1GKQKUN(cmPort: 50
713 httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId DESKTOP-1GKQKUN:50713
2022-07-15 21:47:50,601 INFO rmnode.RMNodeImpl: DESKTOP-1GKQKUN:50713 Node Transitioned from NEW to RUNNING
2022-07-15 21:47:50,621 INFO capacity.CapacityScheduler: Added node DESKTOP-1GKQKUN:50713 clusterResource: <memory:81
92, vCores:8>
```

```
2022-07-15 21:45:47,643 INFO impl.FsDatasetImpl: Total time to scan all replicas for block pool BP-1951056128-192.168
.1.7-1644143530070: 64ms
2022-07-15 21:45:47,644 INFO impl.FsDatasetImpl: Adding replicas to map for block pool BP-1951056128-192.168.1.7-1644
143530070 on volume Z:\BigData\hadoop-3.2.2\dfs\datanode...
2022-07-15 21:45:47,645 INFO impl.BlockPoolSlice: Replica Cache file: Z:\BigData\hadoop-3.2.2\dfs\datanode\current\BP
-1951056128-192.168.1.7-1644143530070\current\replicas doesn't exist
2022-07-15 21:45:47,672 INFO impl.FsDatasetImpl: Time to add replicas to map for block pool BP-1951056128-192.168.1.7
-1644143530070 on volume Z:\BigData\hadoop-3.2.2\dfs\datanode: 27ms
2022-07-15 21:45:47,673 INFO impl.FsDatasetImpl: Total time to add all replicas to map for block pool BP-1951056128-1
92.168.1.7-1644143530070: 29ms
2022-07-15 21:45:47,673 INFO checker.ThrottledAsyncChecker: Scheduling a check for Z:\BigData\hadoop-3.2.2\dfs\datano
de
2022-07-15 21:45:47,682 INFO checker.DatasetVolumeChecker: Scheduled health check for volume Z:\BigData\hadoop-3.2.2\
dfs\datanode
2022-07-15 21:45:47,701 INFO datanode.VolumeScanner: VolumeScanner(Z:\BigData\hadoop-3.2.2\dfs\datanode, DS-6d35dabc-
0008-4c85-8303-e3005e89107c): no suitable block pools found to scan.  Waiting 608431460 ms.
2022-07-15 21:45:47,733 INFO datanode.DirectoryScanner: Periodic Directory Tree Verification scan starting at 16/07/2
2, 1:31 AM with interval of 21600000ms
2022-07-15 21:45:47,741 INFO datanode.DataNode: Block pool BP-1951056128-192.168.1.7-1644143530070 (Datanode Uuid 7c7
da0eb-c504-45f5-abbe-de408c4286cc) service to localhost/127.0.0.1:9000 beginning handshake with NN
2022-07-15 21:45:47,787 INFO datanode.DataNode: Block pool Block pool BP-1951056128-192.168.1.7-1644143530070 (Datano
de Uuid 7c7da0eb-c504-45f5-abbe-de408c4286cc) service to localhost/127.0.0.1:9000 successfully registered with NN
2022-07-15 21:45:47,787 INFO datanode.DataNode: For namenode localhost/127.0.0.1:9000 using BLOCKREPORT_INTERVAL of 2
1600000msec CACHEREPORT_INTERVAL of 10000msec Initial delay: 0msec; heartBeatInterval=3000
2022-07-15 21:45:47,945 INFO datanode.DataNode: Successfully sent block report 0x6ce2a69a720089c0,  containing 1 stor
age report(s), of which we sent 1. The reports had 0 total blocks and used 1 RPC(s). This took 3 msec to generate and
 56 msecs for RPC and NN processing. Got back one command: FinalizeCommand/5.
2022-07-15 21:45:47,946 INFO datanode.DataNode: Got finalize command for block pool BP-1951056128-192.168.1.7-1644143
530070
```

```
        at org.apache.hadoop.hdfs.protocol.proto.DatanodeProtocolProtos$DatanodeProtocolService$2.callBlockingMethod(
DatanodeProtocolProtos.java:31672)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:528)
        at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:1086)
        at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:1029)
        at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:957)
        at java.base/java.security.AccessController.doPrivileged(Native Method)
        at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1762)
        at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2957)
2022-07-15 21:45:42,588 INFO fs.TrashPolicyDefault: The configured checkpoint interval is 0 minutes. Using an interva
l of 1440 minutes that is used for deletion instead
2022-07-15 21:45:42,588 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 1440 minutes, E
mptier interval = 1440 minutes.
2022-07-15 21:45:47,775 INFO hdfs.StateChange: BLOCK* registerDatanode: from DatanodeRegistration(127.0.0.1:9866, dat
anodeUuid=7c7da0eb-c504-45f5-abbe-de408c4286cc, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid
=CID-e761f835-5479-48e3-a70d-1b47acc13d6f;nsid=557392828;c=1644143530070) storage 7c7da0eb-c504-45f5-abbe-de408c4286c
c
2022-07-15 21:45:47,776 INFO net.NetworkTopology: Adding a new node: /default-rack/127.0.0.1:9866
2022-07-15 21:45:47,776 INFO blockmanagement.BlockReportLeaseManager: Registered DN 7c7da0eb-c504-45f5-abbe-de408c428
6cc (127.0.0.1:9866).
2022-07-15 21:45:47,868 INFO blockmanagement.DatanodeDescriptor: Adding new storage ID DS-6d35dabc-0008-4c85-8303-e30
05e89107c for DN 127.0.0.1:9866
2022-07-15 21:45:47,906 INFO BlockStateChange: BLOCK* processReport 0x6ce2a69a720089c0: Processing first storage repo
rt for DS-6d35dabc-0008-4c85-8303-e3005e89107c from datanode 7c7da0eb-c504-45f5-abbe-de408c4286cc
2022-07-15 21:45:47,908 INFO BlockStateChange: BLOCK* processReport 0x6ce2a69a720089c0: from storage DS-6d35dabc-0008
-4c85-8303-e3005e89107c node DatanodeRegistration(127.0.0.1:9866, datanodeUuid=7c7da0eb-c504-45f5-abbe-de408c4286cc,
infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-e761f835-5479-48e3-a70d-1b47acc13d6f;nsid=5
57392828;c=1644143530070), blocks: 0, hasStaleStorage: false, processing time: 2 msecs, invalidatedBlocks: 0
```



```
Microsoft Windows [Version 10.0.22000.795]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>start-dfs
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>start-yarn
starting yarn daemons
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>jps
12080 NameNode
14708 ResourceManager
2232 Jps
2648 DataNode
14556 NodeManager

C:\WINDOWS\system32>
```

PROGRAM 5

Execution of HDFS Commands for interaction with Hadoop

Environment. (Minimum 10 commands to be executed)

```
start-all.sh

This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh

Starting namenodes on [localhost]

hduser@localhost's password:

localhost: starting namenode, logging to
/usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-Precision-T1700.out

hduser@localhost's password:

localhost: starting datanode, logging to
/usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-Precision-T1700.out

Starting secondary namenodes [0.0.0.0]

hduser@0.0.0.0's password:

0.0.0.0: starting secondarynamenode, logging to
/usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-bmsce-Precision-T1700.out

starting yarn daemons

starting resourcemanager, logging to
/usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-Precision-T1700.out

hduser@localhost's password:

localhost: starting nodemanager, logging to
/usr/local/hadoop/logs/yarn-hduser-nodemanager-bmsce-Precision-T1700.out

hduser@bmsce-Precision-T1700:~$ jps

5072 SecondaryNameNode

4674 NameNode

4856 DataNode

5563 NodeManager

6507 Jps

5231 ResourceManager

hduser@bmsce-Precision-T1700:~$ ls
```

```
 derby.log                              Pictures
 Desktop                                pig_1564816082257.log
 Documents                                pig_1599215374374.log
 Downloads                                pt
 examples.desktop                           PT72Installer
 first.text                           Public
 hadoop-2.6.0.tar.gz                        R
 hive                                 TCPclient.py
 lol                                 TCPserver.py
 metastore_db                             Templates
 Music                                toinstalledlist
 newfile.txt                            UDPclient.py
newnewfile.txt                            UDPserver.py
'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz' Videos
hduser@bmsce-Precision-T1700:~$ /home/desktop
bash: /home/desktop: No such file or directory
hduser@bmsce-Precision-T1700:~$ /home/Desktop
bash: /home/Desktop: No such file or directory
hduser@bmsce-Precision-T1700:~$ cd desktop
bash: cd: desktop: No such file or directory
hduser@bmsce-Precision-T1700:~$ cd Desktop
hduser@bmsce-Precision-T1700:~/Desktop$ cd ..
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir ~/Desktop/Ishan
mkdir: `/home/hduser/Desktop/Ishan': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /abc
hduser@bmsce-Precision-T1700:~$ ls
 derby.log                              Pictures
 Desktop                                pig_1564816082257.log
 Documents                              pig_1599215374374.log
 Downloads                              pt
 examples.desktop                          PT72Installer
 first.text                         Public
 hadoop-2.6.0.tar.gz                       R
 hive                              TCPclient.py
 lol                              TCPserver.py
 metastore_db                            Templates
 Music                              toinstalledlist
 newfile.txt                          UDPclient.py
newnewfile.txt                          UDPserver.py
'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz' Videos
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 7 items
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:52 /SharmaJi
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:34 /abc
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:44 /bhavana
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:22 /lochan
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:45 /u1
-rw-r--r--   1 hduser supergroup         19 2022-05-31 11:01 /user
drwxr-xr-x   - hduser supergroup          0 2022-06-01 10:08 /vallisha
hduser@bmsce-Precision-T1700:~$ cat newfile.txt
SharmaJi
KhanwaJi
PaiJI
Kasturba
pandeyji
patilwa
```

```
Nairwa
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/newfile.txt /abc/ishan.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs cat /abc/ishan.txt
cat: Unknown command
Did you mean -cat? This command begins with a dash.
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/ishan.txt
SharmaJi
KhanwaJi
PaiJI
Kasturba
pandeyji
patilwa
Nairwa
hduser@bmsce-Precision-T1700:~$ cat newnewfile.txt
SharmaJi
KhanwaJi
PaiJI
Kasturba
pandeyji
patilwa
Nairwa
hduser@bmsce-Precision-T1700:~$ cat > ishan.txt
Hello
This is a new text file
^C
hduser@bmsce-Precision-T1700:~$ cat ishan.txt
Hello
This is a new text file
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/ishan.txt
/abc/ishan2.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/ishan2.txt

Hello

This is a new text file

hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/ishan2.txt /home/hduser/ishan2_copy.txt

hduser@bmsce-Precision-T1700:~$ ls

 derby.log           'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz'

 Desktop              Pictures

 Documents            pig_1564816082257.log

 Downloads            pig_1599215374374.log

 examples.desktop pt

 first.text           PT72Installer

 hadoop-2.6.0.tar.gz Public

 hive                 R

 ishan2_copy.txt      TCPclient.py

 ishan.txt            TCPserver.py

 lol                  Templates

 metastore_db         toinstalledlist

 Music                UDPclient.py

 newfile.txt          UDPserver.py

 newnewfile.txt       Videos

hduser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /abc/ishan.txt /abc/ishan2.txt
/home/hduser/ishan_merge.txt

hduser@bmsce-Precision-T1700:~$ ls

 derby.log           'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz'

 Desktop              Pictures

 Documents            pig_1564816082257.log

 Downloads            pig_1599215374374.log
 examples.desktop      pt
 first.text            PT72Installer
 hadoop-2.6.0.tar.gz Public
 hive                  R
 ishan2_copy.txt       TCPclient.py
 ishan_merge.txt       TCPserver.py
 ishan.txt             Templates
 lol                   toinstalledlist
 metastore_db          UDPclient.py
 Music                 UDPserver.py
 newfile.txt           Videos
 newnewfile.txt
hduser@bmsce-Precision-T1700:~$ ishan_merge.txt
ishan_merge.txt: command not found
hduser@bmsce-Precision-T1700:~$ cat ishan_merge.txt
SharmaJi
KhanwaJi
PaiJI
Kasturba
pandeyji
patilwa
Nairwa
Hello
This is a new text file
hduser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /abc/
# file: /abc
# owner: hduser
# group: supergroup
```

```
user::rwx
group::r-x
other::r-x

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /abc/ishan.txt /home/hduser/Desktop
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 7 items
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:52 /SharmaJi
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:40 /abc
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:44 /bhavana
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:22 /lochan
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:45 /u1
-rw-r--r--   1 hduser supergroup         19 2022-05-31 11:01 /user
drwxr-xr-x   - hduser supergroup          0 2022-06-01 10:08 /vallisha
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /ishan
hduser@bmsce-Precision-T1700:~$ hadoop fs -mv /abc /ishan
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /ishan
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:40 /ishan/abc
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /abc
ls: `/abc': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 7 items
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:52 /SharmaJi
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:44 /bhavana
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:59 /ishan
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:22 /lochan
drwxr-xr-x   - hduser supergroup          0 2022-06-03 15:45 /u1
```

```
-rw-r--r--   1 hduser supergroup         19 2022-05-31 11:01 /user
drwxr-xr-x   - hduser supergroup          0 2022-06-01 10:08 /vallisha
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat/ishan.txt
-cat/ishan.txt: Unknown command
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /ishan.txt
cat: `/ishan.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /ishan/ishan.txt
cat: `/ishan/ishan.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /vallisha
Found 1 items
-rw-r--r--   1 hduser supergroup         13 2022-06-01 09:52 /vallisha/sample1.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /ishan/abc
Found 2 items
-rw-r--r--   1 hduser supergroup         57 2022-06-04 09:37 /ishan/abc/ishan.txt
-rw-r--r--   1 hduser supergroup         30 2022-06-04 09:40 /ishan/abc/ishan2.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /ishan/ishan.txt
cat: `/ishan/ishan.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /ishan/abc/ishan.txt
SharmaJi
KhanwaJi
PaiJI
Kasturba
pandeyji
patilwa
Nairwa
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cp /vallisha/sample1.txt /ishan
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /ishan
Found 2 items
```

```
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:40 /ishan/abc
-rw-r--r--   1 hduser supergroup         13 2022-06-04 10:07 /ishan/sample1.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /ishan/sample1.txt
sample1 text
hduser@bmsce-Precision-T1700:~$ sudo nano new_file.txt
[sudo] password for hduser:
hduser@bmsce-Precision-T1700:~$ ls
 derby.log            newnewfile.txt
 Desktop             'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz'
 Documents            Pictures
 Downloads            pig_1564816082257.log
 examples.desktop     pig_1599215374374.log
 first.text           pt
 hadoop-2.6.0.tar.gz  PT72Installer
 hive                 Public
 ishan2_copy.txt      R
 ishan_merge.txt      TCPclient.py
 ishan.txt            TCPserver.py
 lol                  Templates
 metastore_db         toinstalledlist
 Music                UDPclient.py
 new_file.txt         UDPserver.py
 newfile.txt          Videos
hduser@bmsce-Precision-T1700:~$ cat new_file.txt
This is a new file, created using sudo nano
```

PROGRAM 6

From the following link extract the weather data

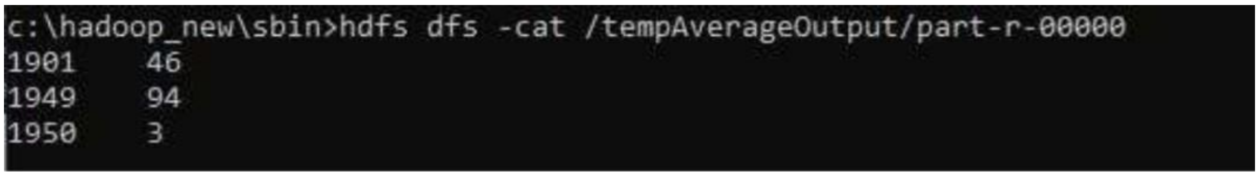
https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all

Create a Map Reduce program to

a) find average temperature for each year from NCDC data set.

```
\\AVERAGE DRIVER
package temp;
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 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);
}
}

\\AVERAGE MAPPER
package temp;
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 AverageMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
int temperature;
String line = value.toString();
String year = line.substring(15, 19);
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 != 9999 && quality.matches("[01459]"))
context.write(new Text(year), new IntWritable(temperature));
}
```

```
}

\\AVERAGE REDUCER
package temp;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class AverageReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int count = 0;
for (IntWritable value : values) {
max_temp += value.get();
count++;

}
context.write(key, new IntWritable(max_temp / count));
}
}
```



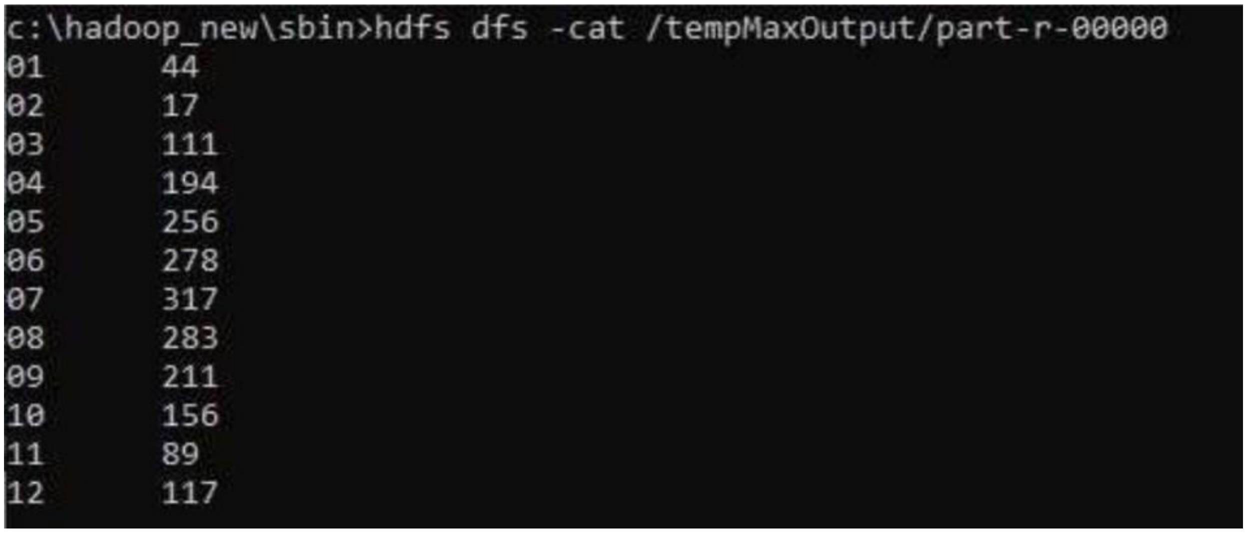
```
c:\hadoop_new\sbin>hdfs dfs -cat /tempAverageOutput/part-r-00000
1901    46
1949    94
1950    3
```

b) find the mean max temperature for every month

```
\\MEAN MAX DRIVER
package meanmax;
```

```
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 MeanMaxDriver {
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(MeanMaxDriver.class);
job.setJobName("Max temperature");
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.setMapperClass(MeanMaxMapper.class);
job.setReducerClass(MeanMaxReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

\\MEAN MAX MAPPER
package meanmax;
```

```
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 MeanMaxMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
int temperature;
String line = value.toString();
String month = line.substring(19, 21);
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 != 9999 && quality.matches("[01459]"))
context.write(new Text(month), new IntWritable(temperature));
}
}

\\MEAN MAX REDUCER
package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class MeanMaxReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int total_temp = 0;
int count = 0;

int days = 0;
for (IntWritable value : values) {
int temp = value.get();
if (temp > max_temp)
max_temp = temp;
count++;
if (count == 3) {
total_temp += max_temp;
max_temp = 0;
count = 0;
days++;
}
}
context.write(key, new IntWritable(total_temp / days));
}
}
```



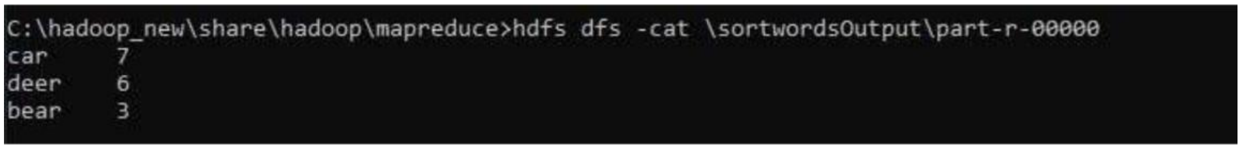
```
c:\hadoop_new\sbin>hdfs dfs -cat /tempMaxOutput/part-r-00000
01      44
02      17
03      111
04      194
05      256
06      278
07      317
08      283
09      211
10      156
11      89
12      117
```

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.

```
\\TOPN

package samples.topn;

import java.io.IOException;

import java.util.StringTokenizer;

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.lib.input.FileInputFormat;

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

import org.apache.hadoop.util.GenericOptionsParser;

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);
}

\\TOPN COMBINER
package samples.topn;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class TopNCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values)
sum += val.get();
context.write(key, new IntWritable(sum));
}
}

\\TOPN MAPPER
package samples.topn;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
private static final IntWritable one = new IntWritable(1);
private Text word = new Text();
private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";
public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
StringTokenizer itr = new StringTokenizer(cleanLine);
while (itr.hasMoreTokens()) {
this.word.set(itr.nextToken().trim());
context.write(this.word, one);
}
}
}

\\TOPN REDUCER
package samples.topn;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;
public class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
```

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

public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values)
sum += val.get();
this.countMap.put(new Text(key), new IntWritable(sum));
}
protected void cleanup(Reducer<Text, IntWritable, Text, IntWritable>.Context context)
throws IOException, InterruptedException {
Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(this.countMap);
int counter = 0;
for (Text key : sortedMap.keySet()) {
if (counter++ == 20)
break;
context.write(key, sortedMap.get(key));
}
}
}
```



```
C:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \sortwordsOutput\part-r-00000
car     7
deer    6
bear    3
```

PROGRAM 8

Create a Map Reduce program to demonstrating join operation

```
// JoinDriver.java
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,
User.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
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
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]));
}
}

// TextPair.java
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);
}
// ^^ TextPair
// vv TextPairComparator
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);
}
} }

// User.java
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 User 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();

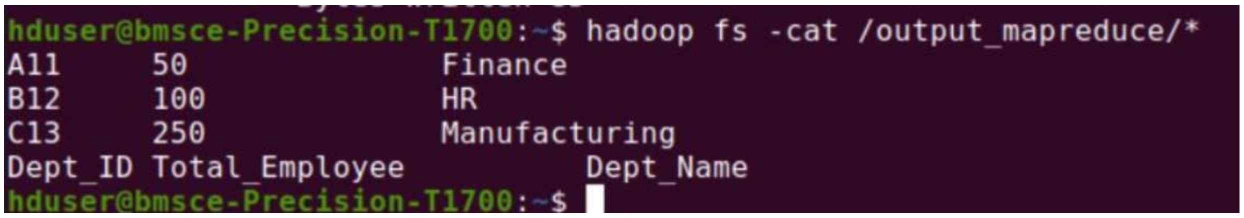
String[] SingleNodeData = valueString.split("\t");

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

Text(SingleNodeData[1]));

}

}



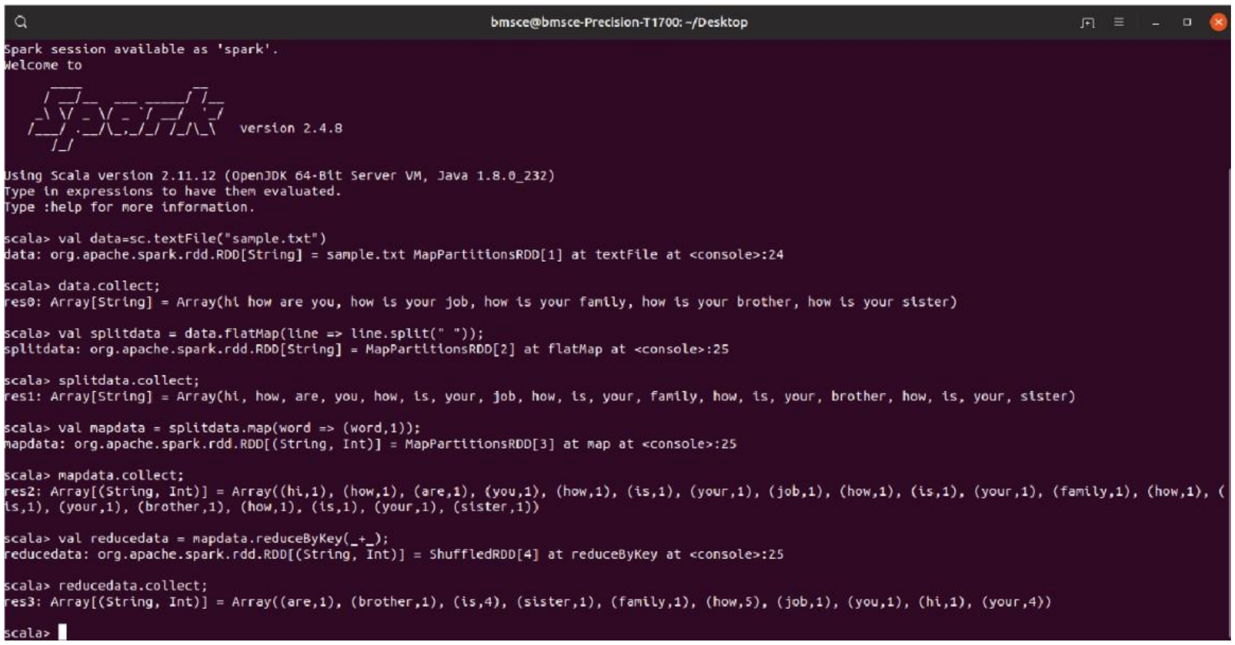
```
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat /output_mapreduce/*
A11     50              Finance
B12     100             HR
C13     250             Manufacturing
Dept_ID Total_Employee          Dept_Name
hduser@bmsce-Precision-T1700:~$
```

PROGRAM 9

Program to print word count on scala shell and print “Hello world” on scala IDE

```
val data=sc.textFile("sparkdata.txt")

data.collect;

val splitdata = data.flatMap(line => line.split(" "));

splitdata.collect;

val mapdata = splitdata.map(word => (word,1));

mapdata.collect;

val reducedata = mapdata.reduceByKey(_+_);

reducedata.collect;
```

```
package hellooWorld

object hello {

  def main (args: Array[String]) {

    println("Hello World")

  }

}
```



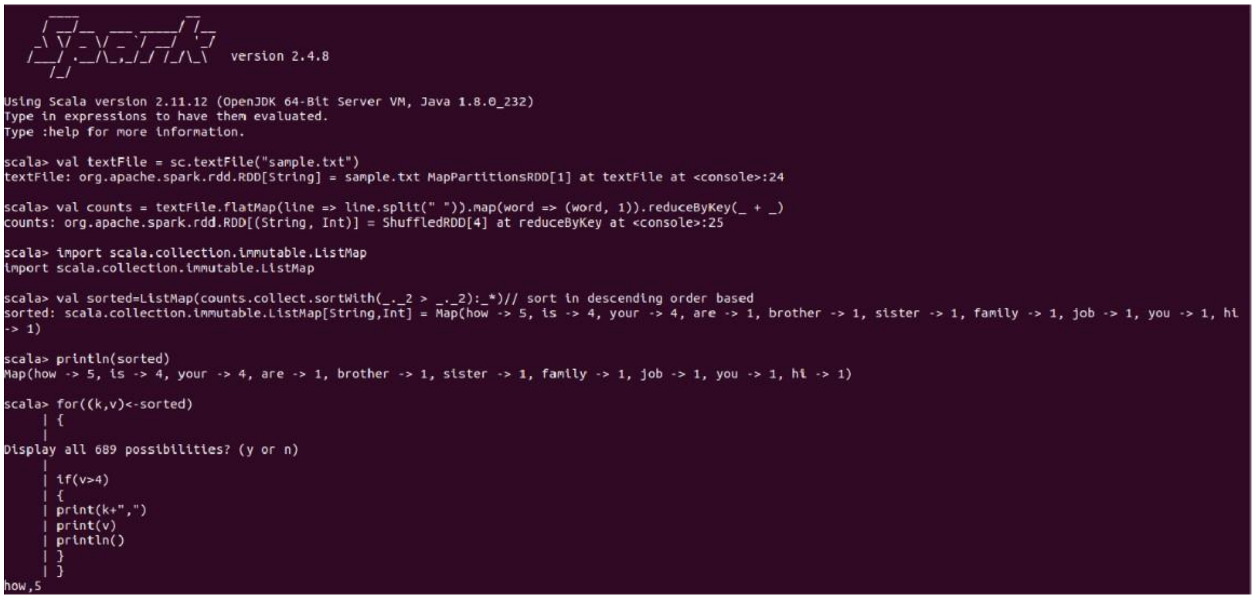
bmsce@bmsce-Precision-T1700: ~/Desktop

```
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.8
      /_/

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_232)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val data=sc.textFile("sample.txt")
data: org.apache.spark.rdd.RDD[String] = sample.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> data.collect;
res0: Array[String] = Array(hi how are you, how is your job, how is your family, how is your brother, how is your sister)

scala> val splitdata = data.flatMap(line => line.split(" "));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:25

scala> splitdata.collect;
res1: Array[String] = Array(hi, how, are, you, how, is, your, job, how, is, your, family, how, is, your, brother, how, is, your, sister)

scala> val mapdata = splitdata.map(word => (word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:25

scala> mapdata.collect;
res2: Array[(String, Int)] = Array((hi,1), (how,1), (are,1), (you,1), (how,1), (is,1), (your,1), (job,1), (how,1), (is,1), (your,1), (family,1), (how,1), (
is,1), (your,1), (brother,1), (how,1), (is,1), (your,1), (sister,1))

scala> val reducedata = mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:25

scala> reducedata.collect;
res3: Array[(String, Int)] = Array((are,1), (brother,1), (is,4), (sister,1), (family,1), (how,5), (job,1), (you,1), (hi,1), (your,4))

scala> 
```

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

```
val textFile = sc.textFile("/home/bhoom/Desktop/wc.txt")

val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ +
_)

import scala.collection.immutable.ListMap

val sorted=ListMap(counts.collect.sortWith(_._2 > _._2):_*)// sort in descending order based

on values

println(sorted)

for((k,v)<-sorted)

{

if(v>4)

{

print(k+",")

print(v)

println()

}

}
```



```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.8
      /_/

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_232)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val textFile = sc.textFile("sample.txt")
textFile: org.apache.spark.rdd.RDD[String] = sample.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:25

scala> import scala.collection.immutable.ListMap
import scala.collection.immutable.ListMap

scala> val sorted=ListMap(counts.collect.sortWith(_._2 > _._2):_*)// sort in descending order based
sorted: scala.collection.immutable.ListMap[String,Int] = Map(how -> 5, is -> 4, your -> 4, are -> 1, brother -> 1, sister -> 1, family -> 1, job -> 1, you -> 1, hi
-> 1)

scala> println(sorted)
Map(how -> 5, is -> 4, your -> 4, are -> 1, brother -> 1, sister -> 1, family -> 1, job -> 1, you -> 1, hi -> 1)

scala> for((k,v)<-sorted)
     | {
     |
Display all 689 possibilities? (y or n)
     |
     | if(v>4)
     | {
     | print(k+",")
     | print(v)
     | println()
     | }
     | }
how,5
```