

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

### BIG DATA ANALYTICS (20CS6PEBDA)

*Submitted by*

**NEEHAL (1BM19CS097)**

*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 "**BIG DATA ANALYTICS**" carried out by Neehal (**1BM19CS097**), 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.

**Pallavi GB**

Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

**Dr. Jyothi S Nayak**

Professor and Head  
Department of CSE  
BMSCE, Bengaluru

## Index Sheet

<b>Sl. No.</b>	<b>Experiment Title</b>	<b>Page No.</b>
<b>1</b>	<b>Employee Database</b>	<b>4</b>
<b>2</b>	<b>Library</b>	<b>6</b>
<b>3</b>	<b>Mongo (CRUD)</b>	<b>8</b>
<b>4</b>	<b>Hadoop installation</b>	<b>19</b>
<b>5</b>	<b>HDFS Commands</b>	<b>20</b>
<b>6</b>	<b>Create a Map Reduce program to a) find average temperature for each year from the NCDC data set. b) find the mean max temperature for every month</b>	<b>23</b>
<b>7</b>	<b>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.</b>	<b>33</b>
<b>8</b>	<b>Create a Map Reduce program to demonstrating join operation</b>	<b>38</b>
<b>9</b>	<b>Program to print word count on Scala shell and print “Hello world” on Scala IDE</b>	<b>46</b>
<b>10</b>	<b>Using RDD and Flat Map 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</b>	<b>47</b>

### 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

## **BDA LAB 1**

Program 1. Perform the following DB operations using Cassandra.

1. Create a key space by name Employee

```
cqlsh:employee2> create keyspace employee4 with replication={'class':'SimpleStrategy','replication_factor':1};  
cqlsh:employee2>
```

2. Create a column family by name Employee-Info with attributes

Emp\_Id Primary Key, Emp\_Name, Designation, Date\_of\_Joining,  
Salary, Dept\_Name

```
cqlsh:employee2> create employee_info (id int Primary Key,name text,designation text,doj timestamp,salary double,department t  
ext);
```

3. Insert the values into the table in batch

```
cqlsh:employee2> begin batch  
    ... insert into employee_info(id,name,designation,doj,salary,department) values (1,'aruna','manager','2019-05-04  
' ,125000,'management')  
    ... insert into employee_info(id,name,designation,doj,salary,department) values (2,'manju','developer','2020-05-04',  
'185000,'dev')  
    ... insert into employee_info(id,name,designation,doj,salary,department) values (3,'raj','developer','2021-05-04  
' ,100000,'dev')  
    ... apply batch;
```

4. Update Employee name and Department of Emp-Id 2

```
cqlsh:employee2> update employee_info set name='ram',department='management' where id=2;  
cqlsh:employee2> select * from employee_info;
```

5. Sort the details of Employee records based on salary

```
cqlsh:employee2> select * from employee_info where id in(1,2,3) orderby salary;
```

<b>id</b>	<b>department</b>	<b>designation</b>	<b>doj</b>	<b>name</b>	<b>salary</b>
1	management	manager	2019-05-03 18:30:00.000000+0000	aruna	1.25e+05
2	management	developer	2020-05-03 18:30:00.000000+0000	ram	1.05e+05
3	dev	developer	2021-05-03 18:30:00.000000+0000	raj	1e+05

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.

```
cqlsh:employee2> alter table employee_info add projects set <text>;  
cqlsh:employee2> select * from employee_info;
```

<b>id</b>	<b>department</b>	<b>designation</b>	<b>doj</b>	<b>name</b>	<b>projects</b>	<b>salary</b>
1	management	manager	2019-05-03 18:30:00.000000+0000	aruna	null	1.25e+05
2	management	developer	2020-05-03 18:30:00.000000+0000	ram	null	1.05e+05
3	dev	developer	2021-05-03 18:30:00.000000+0000	raj	null	1e+05

7. Update the altered table to add project names.

```
cqlsh:employee2> update employee_info set projects=projects+['p1','p2','p3'] where id=1;  
cqlsh:employee2> select * from employee_info;
```

<b>id</b>	<b>department</b>	<b>designation</b>	<b>doj</b>	<b>name</b>	<b>projects</b>	<b>salary</b>
1	management	manager	2019-05-03 18:30:00.000000+0000	aruna	{'p1', 'p2', 'p3'}	1.25e+05
2	management	developer	2020-05-03 18:30:00.000000+0000	ram	null	1.05e+05
3	dev	developer	2021-05-03 18:30:00.000000+0000	raj	null	1e+05

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

```
cqlsh:employee2> insert into employee_info(id,name,designation,doj,salary,department) values (4,'mamatha','marketing manager','2019-04-03',123000,'marketing') using ttl 45;
cqlsh:employee2> select * from employee_info;


| id | department | designation       | doj                             | name    | projects           | salary   |
|----|------------|-------------------|---------------------------------|---------|--------------------|----------|
| 1  | management | manager           | 2019-05-03 18:30:00.000000+0000 | aruna   | {'p1', 'p2', 'p3'} | 1.25e+05 |
| 2  | management | developer         | 2020-05-03 18:30:00.000000+0000 | ram     | null               | 1.05e+05 |
| 4  | marketing  | marketing manager | 2019-04-02 18:30:00.000000+0000 | mamatha | null               | 1.23e+05 |
| 3  | dev        | developer         | 2021-05-03 18:30:00.000000+0000 | raj     | null               | 1e+05    |


(4 rows)
cqlsh:employee2> select ttl(department) from employee_info;


| ttl(department) |
|-----------------|
| null            |
| null            |
| 2               |
| null            |


(4 rows)

cqlsh:employee2> select * from employee_info;


| id | department | designation | doj                             | name  | projects           | salary   |
|----|------------|-------------|---------------------------------|-------|--------------------|----------|
| 1  | management | manager     | 2019-05-03 18:30:00.000000+0000 | arena | {'p1', 'p2', 'p3'} | 1.25e+05 |
| 2  | management | developer   | 2020-05-03 18:30:00.000000+0000 | ram   | null               | 1.05e+05 |
| 3  | dev        | developer   | 2021-05-03 18:30:00.000000+0000 | raj   | null               | 1e+05    |


```

## **BDA LAB 2**

Perform the following DB operations using Cassandra:

1 Create a key space by name Library

```
cqlsh> create keyspace Library1 WITH REPLICATION={'class':'SimpleStrategy','replication_factor':2}; 2.  
cqlsh> use Library1;
```

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:library1> create table Library_info(id int,c_val counter,s_name text,b_name text,b_id int,doi timestamp,primary key(id,s_name,b_name,b_id,doi)); 3.
```

Insert the values into the table in batch

```
cqlsh:library1> update Library_info set c_val=c_val+1 where id=1 and s_name='aruna' and b_name='maths' and b_id=1 and doi='2020-02-23'; 4.  
cqlsh:library1> update Library_info set c_val=c_val+1 where id=1 and s_name='ram' and b_name='ccp' and b_id=2 and doi='2020-03-14';  
cqlsh:library1> update Library_info set c_val=c_val+1 where id=3 and s_name='raja' and b_name='BDA' and b_id=3 and doi='2020-03-04';  
cqlsh:library1> select * from Library_info ;
```

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

```
cqlsh:library1> update Library_info set c_val=c_val+2 where id=4 and s_name='manny' and b_name='CNS' and b_id=4 and doi='2020-05-23'; 5.  
cqlsh:library1> select * from Library_info ;
```

<b><u>id</u></b>	<b><u>s_name</u></b>	<b><u>b_name</u></b>	<b><u>b_id</u></b>	<b><u>doi</u></b>	<b><u>c_val</u></b>
1	aruna	maths	1	2020-02-22 18:30:00.000000+0000	1
1	ram	ccp	2	2020-03-13 18:30:00.000000+0000	1
4	manny	CNS	4	2020-05-22 18:30:00.000000+0000	5
3	raja	BDA	3	2020-03-03 18:30:00.000000+0000	1

Write a query to show that a student with id 1.

```
cqlsh:library1> select * from Library_info where id=1 and s_name='aruna' and b_name='maths' and b_id=1
;

```

<code>id</code>	<code>s_name</code>	<code>b_name</code>	<code>b_id</code>	<code>doi</code>	<code>c_val</code>
1	aruna	maths	1	2020-02-21 18:30:00.000000+0000	2
1	aruna	maths	1	2020-02-22 18:30:00.000000+0000	1

Export the created column to a csv file

```
cqlsh:library1> copy Library_info(id,s_name,b_name,b_id,doi,c_val) to '/home/bmsce/aruna.csv';
Using 11 child processes
7.

```

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

```
cqlsh:library1> copy Library_info2(id,s_name,b_name,b_id,doi,c_val) from '/home/bmsce/aruna.csv';
Using 11 child processes

```

```
Starting copy of library1.library_info2 with columns [id, s_name, b_name, b_id, doi, c_val].
Processed: 5 rows; Rate:      8 rows/s; Avg. rate:      12 rows/s
5 rows imported from 1 files in 0.429 seconds (0 skipped).
cqlsh:library1> select * from Library_info2
... ;

```

<code>id</code>	<code>s_name</code>	<code>b_name</code>	<code>b_id</code>	<code>doi</code>	<code>c_val</code>
1	aruna	maths	1	2020-02-21 18:30:00.000000+0000	2
1	aruna	maths	1	2020-02-22 18:30:00.000000+0000	1
1	ram	ccp	2	2020-03-13 18:30:00.000000+0000	1
4	manny	CNS	4	2020-05-22 18:30:00.000000+0000	5
3	raja	BDA	3	2020-03-03 18:30:00.000000+0000	4

(5 rows)

## BDA LAB 3

### Mongo (CRUD)

```
Q bmsce@bmsce-Precision-T1700:~ (base) bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("2139b1da-2835-4627-bad9-167fee5d8c1c") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-13T19:39:10.171+0530 I STORAGE  [initandlisten]
2022-04-13T19:39:10.171+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is
strongly recommended with the WiredTiger storage engine
2022-04-13T19:39:10.171+0530 I STORAGE  [initandlisten] **           See http://dochub.mongodb.or
g/core/prodnotes-filesystem
2022-04-13T19:39:14.150+0530 I CONTROL  [initandlisten]
2022-04-13T19:39:14.150+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enable
d for the database.
2022-04-13T19:39:14.150+0530 I CONTROL  [initandlisten] **           Read and write access to dat
a and configuration is unrestricted.
2022-04-13T19:39:14.150+0530 I CONTROL  [initandlisten]
> use arunaDB;
switched to db arunaDB
> db;
arunaDB
> show dbs;
admin    0.000GB
config   0.000GB
local    0.000GB
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({id:1,name:"Aruna Ravi K R",sem:6,hobbies:"cricket"});
WriteResult({ "nInserted" : 1 })
> db.Student.update({id:3,name:"Manjunatha",sem:1},{$set:{hobbies:"skating"}},{upsert:true})
WriteResult({
    "nMatched" : 0,
    "nUpserted" : 1,
    "nModified" : 0,
    "_id" : ObjectId("6256945713cdf85df653a993")
})
> db.Student.find({name:"Manjunatha"});
[ { "_id" : ObjectId("6256945713cdf85df653a993"), "id" : 3, "name" : "Manjunatha", "sem" : 1, "hob
bies" : "skating" }
> db.Student.find({}, {name:1});
[ { "_id" : ObjectId("625693993478889c646fe83f"), "name" : "Aruna Ravi K R" }
[ { "_id" : ObjectId("6256945713cdf85df653a993"), "name" : "Manjunatha" }
> db.Student.find({sem:6}).pretty();
```

```
Q bmsce@bmsce-Precision-T1700:~ 
{ "_id" : ObjectId("625693993478889c646fe83f"), "name" : "Aruna Ravi K R" }
{ "_id" : ObjectId("6256945713cdf85df653a993"), "name" : "Manjunatha" }
> db.Student.find({sem:6}).pretty();
{
    "_id" : ObjectId("625693993478889c646fe83f"),
    "id" : 1,
    "name" : "Aruna Ravi K R",
    "sem" : 6,
    "hobbies" : "cricket"
}
> db.Student.find({hobbies:{ $in:['sketing','cricket']} }).pretty();
{
    "_id" : ObjectId("625693993478889c646fe83f"),
    "id" : 1,
    "name" : "Aruna Ravi K R",
    "sem" : 6,
    "hobbies" : "cricket"
}
> db.Student.find({name:/^M/}).pretty();
{
    "_id" : ObjectId("6256945713cdf85df653a993"),
    "id" : 3,
    "name" : "Manjunatha",
    "sem" : 1,
    "hobbies" : "skating"
}
> db.Student.find({name:/R/}).pretty();
{
    "_id" : ObjectId("625693993478889c646fe83f"),
    "id" : 1,
    "name" : "Aruna Ravi K R",
    "sem" : 6,
    "hobbies" : "cricket"
}
> db.Student.count();
2
> db.Student.find.sort({name:-1}).pretty();
2022-04-13T14:53:27.847+0530 E QUERY [thread1] TypeError: db.Student.find.sort is not a function
ion :
@(shell):1:1
> db.Student.find().sort({name:-1}).pretty();
{
    "_id" : ObjectId("6256945713cdf85df653a993"),
```

```
Q bmsce@bmsce-Precision-T1700:~ [|] ⌂ - X  
        "name" : "Aruna Ravi K R",  
        "sem" : 6,  
        "hobbies" : "cricket"  
}  
> db.Student.count();  
2  
> db.Student.find.sort({name:-1}).pretty();  
2022-04-13T14:53:27.847+0530 E QUERY [thread1] TypeError: db.Student.find.sort is not a function :  
@(shell):1:1  
> db.Student.find().sort({name:-1}).pretty();  
{  
    "_id" : ObjectId("6256945713cdf85df653a993"),  
    "id" : 3,  
    "name" : "Manjunatha",  
    "sem" : 1,  
    "hobbies" : "skating"  
}  
{  
    "_id" : ObjectId("625693993478889c646fe83f"),  
    "id" : 1,  
    "name" : "Aruna Ravi K R",  
    "sem" : 6,  
    "hobbies" : "cricket"  
}  
> db.Student.save({name:"amar",sem:5});  
WriteResult({ "nInserted" : 1 })  
> db.Student.update({id:3},{$set:{Location:"Network"}));  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.Student.find({name:/r$/}).pretty();  
{ "_id" : ObjectId("625696fe3478889c646fe840"), "name" : "amar", "sem" : 5 }  
> db.Student.update({id:3},{$set{Location:null}});  
2022-04-13T14:58:18.222+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:30  
> db.Student.update({id:3},{$set{Location:null}});  
2022-04-13T14:58:32.140+0530 E QUERY [thread1] SyntaxError: missing : after property id @(shell):1:30  
> db.Student.update({id:3},{$set:{Location:null}});  
2022-04-13T14:59:05.114+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @(shell):1:47  
> db.Student.update({id:3},{$set:{Location:null}});  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.Student.count();
```

```
Q bmsce@bmsce-Precision-T1700:~ ✖  
> db.Student.update({id:3},{$set:{Location:null}});  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.Student.count();  
3  
> db.Student.count({sem:6});  
1  
> db.Student.find({sem:6}).limit(2).pretty();  
{  
    "_id" : ObjectId("625693993478889c646fe83f"),  
    "id" : 1,  
    "name" : "Aruna Ravi K R",  
    "sem" : 6,  
    "hobbies" : "cricket"  
}  
> db.Student.find().sort({name:1}).pretty();  
{  
    "_id" : ObjectId("625693993478889c646fe83f"),  
    "id" : 1,  
    "name" : "Aruna Ravi K R",  
    "sem" : 6,  
    "hobbies" : "cricket"  
}  
{  
    "_id" : ObjectId("6256945713cdf85df653a993"),  
    "id" : 3,  
    "name" : "Manjunatha",  
    "sem" : 1,  
    "hobbies" : "skating",  
    "Location" : null  
}  
{ "_id" : ObjectId("625696fe3478889c646fe840"), "name" : "amar", "sem" : 5 }  
> db.Student.find().skip(2).pretty();  
{ "_id" : ObjectId("625696fe3478889c646fe840"), "name" : "amar", "sem" : 5 }  
>  
> db.createcollection("food");  
2022-04-13T15:03:36.470+0530 E QUERY [thread1] TypeError: db.createcollection is not a function:  
on :  
@(shell):1:1  
> db.createCollection("food");  
{ "ok" : 1 }  
> db.food.insert({id:1,fruits:['m','a','g']});  
WriteResult({ "nInserted" : 1 })  
> db.food.insert({id:2,fruits:['g','m','c']});
```

```
Q bmsce@bmsce-Precision-T1700:~ 
> db.food.insert({id:1,fruits:['m','a','g']});
WriteResult({ "nInserted" : 1 })
> db.food.insert({id:2,fruits:['g','m','c']});
WriteResult({ "nInserted" : 1 })
> db.food.insert({id:3,fruits:['b','m']});
WriteResult({ "nInserted" : 1 })
> db.food.find({fruits:[ 'g', 'm', 'c' ]});
{ "_id" : ObjectId("625699523478889c646fe842"), "id" : 2, "fruits" : [ "g", "m", "c" ] }
> db.food.find({fruits:{$size:2}}).pretty();
{
    "_id" : ObjectId("625699603478889c646fe843"),
    "id" : 3,
    "fruits" : [
        "b",
        "m"
    ]
}
> db.food.find({id:1,{fruits:{$slice:2}}).pretty();
...
...
>
>
> db.food.find({fruits:{all:['m','g']}}).pretty();
{
    "_id" : ObjectId("6256993d3478889c646fe841"),
    "id" : 1,
    "fruits" : [
        "m",
        "a",
        "g"
    ]
}
{
    "id" : ObjectId("625699523478889c646fe842"),
    "id" : 2,
    "fruits" : [
        "g",
        "m",
        "c"
    ]
}
> db.food.update({id:3},{$set:fruits:['b','m','o']});
2022-04-13T15:11:26.973+0530 E QUERY    [thread1] SyntaxError: missing } after property list @s
```

```
bmsce@bmsce-Precision-T1700:~
```

```
> }  
{  
    "_id" : ObjectId("625699523478889c646fe842"),  
    "id" : 2,  
    "fruits" : [  
        "g",  
        "m",  
        "c"  
    ]  
}  
> db.food.update({id:3},{$set:fruits:['b','m','o']});  
2022-04-13T15:11:26.973+0530 E QUERY [thread1] SyntaxError: missing } after property list @shell:1:34  
> db.food.update({id:3},{$set:{fruits:['b','m','o']} });  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })  
> db.food.update({id:2},{$push:{price:{g=80,m:100,c:150,b:60,o:70}});  
...  
...  
> db.createCollection("Customer");  
{ "ok" : 1 }  
> db.Customer.aggregate({$group:{id:"$custID",TolAccBal:{$sum:"$AccBal"} }});  
  
assert: command failed: {  
    "ok" : 0,  
    "errmsg" : "The field 'id' must be an accumulator object",  
    "code" : 40234,  
    "codeName" : "Location40234"  
} : aggregate failed  
_getErrorWithCode@src/mongo/shell/utils.js:25:13  
doassert@src/mongo/shell/assert.js:16:14  
assert.commandWorked@src/mongo/shell/assert.js:403:5  
DB.prototype._runAggregate@src/mongo/shell/db.js:260:9  
DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12  
@(shell):1:1  
  
2022-04-13T15:16:14.450+0530 E QUERY [thread1] Error: command failed: {  
    "ok" : 0,  
    "errmsg" : "The field 'id' must be an accumulator object",  
    "code" : 40234,  
    "codeName" : "Location40234"  
} : aggregate failed  
_getErrorWithCode@src/mongo/shell/utils.js:25:13  
doassert@src/mongo/shell/assert.js:16:14
```

## 1) Using MongoDB

- i) Create a database for Students and Create a Student Collection (\_id,Name, USN, Semester, Dept\_Name, CGPA, Hobbies(Set)).
  - ii) Insert required documents to the collection.
  - iii) First Filter on “Dept\_Name:CSE” and then group it on “Semester” and compute the Average CPGA for that semester and filter those documents where the “Avg\_CPGA” is greater than 7.5.
  - iv) Command used to export MongoDB JSON documents from “Student” Collection into the “Students” database into a CSV file “Output.txt”.

```
> db.Student.insert({_id:1,Name:"Aravind",USN:"1BM19CS001",Sem:6,Dept_name:"CSE",CGPA:"9.6",Hobbies:"Badminton"});  
WriteResult({ "nInserted" : 1 })  
> db.Student.insert({_id:2,Name:"Aman",USN:"1BM19EC002",Sem:7,Dept_name:"ECE",CGPA:"9.1",Hobbies:"Swimming"});  
WriteResult({ "nInserted" : 1 })  
> db.Student.insert({_id:3,Name:"Latha",USN:"1BM19CS003",Sem:6,Dept_name:"CSE",CGPA:"8.1",Hobbies:"Reading"});  
WriteResult({ "nInserted" : 1 })  
> db.Student.insert({_id:4,Name:"Sam",USN:"1BM19CS004",Sem:6,Dept_name:"CSE",CGPA:"6.5",Hobbies:"Cycling"});  
WriteResult({ "nInserted" : 1 })  
> db.Student.insert({_id:5,Name:"Suman",USN:"1BM19CS005",Sem:5,Dept_name:"CSE",CGPA:"7.6",Hobbies:"Cycling"});  
WriteResult({ "nInserted" : 1 })
```

```

> db.Student.update({_id:1},{$set:{CGPA:9.0}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.update({_id:2},{$set:{CGPA:9.1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.update({_id:3},{$set:{CGPA:8.1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.update({_id:4},{$set:{CGPA:6.5}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.update({_id:5},{$set:{CGPA:8.6}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.aggregate([{$match:{Dept_name:"CSE"}},{$group:{_id:"$Sem",AvgCGPA:[$avg:"$CGPA"]}},{$match:{AvgCGPA:{$gt:7.5}}}],{_id : 5, "AvgCGPA" : 8.6 },{_id : 6, "AvgCGPA" : 7.8666666666666667 } )

```

Open	+	output.txt ~/Desktop
<u>_id</u> ,Name,USN,Sem,Dept-name,CGPA,Hobbies		
1,Aravind,1BM19CS001,6,,9,Badminton		
2,Aman,1BM19EC002,7,,9.1,Swimming		
3,Latha,1BM19CS003,6,,8.1,Reading		
4,Sam,1BM19CS004,6,,6.5,Cycling		
5,Suman,1BM19CS005,5,,8.6,Cycling		

2)Create a mongodb collection Bank. Demonstrate the following by choosing fields of your choice.

1. Insert three documents
2. Use Arrays(Use Pull and Pop operation)
3. Use Index
4. Use Cursors
5. Updation

```
> db.createCollection("Bank");
{ "ok" : 1 }
> db.insert({CustID:1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]});
uncaught exception: TypeError: db.insert is not a function
@(shell):1:1
> db.Bank.insert({CustID:1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:2, Name:"Vishvesh Bhat", Type:"Savings", Contact:["6325985615", "080-23651452"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:3, Name:"Vaishak Bhat", Type:"Savings", Contact:["8971456321", "080-33529458"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:4, Name:"Pramod P Parande", Type:"Current", Contact:["9745236589", "080-56324587"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:4, Name:"Shreyas R S", Type:"Current", Contact:["9445678321", "044-65611729", "080-25639856"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.find();
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231", "080-22364587" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ] }
> db.Bank.updateMany({custID:1},{$op:{Contact:1}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.Bank.find();
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ] }
```

```
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ] }
> db.Bank.updateMany({},{$pull:{Contact:"080-25639856"}});
{ "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 1 }
> db.Bank.find();
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] }
> db.Bank.createIndex({Name:1, Type:1},{name:1});
uncaught exception: SyntaxError: expected expression, got '}' :
@(shell):1:43
> db.Bank.createIndex({Name:1, Type:1},{name:"Find current account holders"});
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.Bank.find();
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] }
> db.Bank.getIndexes()
[ {
  "v" : 2,
```

```

@(shell):1:20
> db.Bank.update({_id:625d78659329139694f188a6}, {$set: {CustID:5}}, {upsert:true});
uncaught exception: SyntaxError: identifier starts immediately after numeric literal :
@(shell):1:20
> db.Bank.update({_id:"625d78659329139694f188a6"}, {$set: {CustID:5}}, {upsert:true});
WriteResult({
    "nMatched" : 0,
    "nUpserted" : 1,
    "nModified" : 0,
    "_id" : "625d78659329139694f188a6"
})
> db.Bank.find();
[{"_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] },
 {"_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] },
 {"_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] },
 {"_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] },
 {"_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] },
 {"_id" : "625d78659329139694f188a6", "CustID" : 5 }
> db.Bank.update({_id:"625d78659329139694f188a6"}, CustID:5), {$set: {Name:"Sumantha K S", Type:"Savings", Contact:["9856321478", "011-65897458"]}}, {upsert:true};
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Bank.find();
[{"_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] },
 {"_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] },
 {"_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] },
 {"_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] },
 {"_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] },
 {"_id" : "625d78659329139694f188a6", "CustID" : 5, "Contact" : [ "9856321478", "011-65897458" ], "Name" : "Sumantha K S", "Type" : "Savings" }
]

```

### 1) Using MongoDB,

- Create a database for Faculty and Create a Faculty Collection(Faculty\_id, Name, Designation ,Department, Age, Salary, Specialization(Set)).
- Insert required documents to the collection.
- First Filter on “Dept\_Name:MECH” and then group it on “Designation” and compute the Average Salary for that Designation and filter those documents where the “Avg\_Sal” is greater than 650000.
- Demonstrate usage of import and export commands

Write MongoDB queries for the following:

- To display only the product name from all the documents of the product collection.
- To display only the Product ID, ExpiryDate as well as the quantity from the document of the product collection where the \_id column is 1.
- To find those documents where the price is not set to 15000.
- To find those documents from the Product collection where the quantity is set to 9 and the product name is set to ‘monitor’.
- To find documents from the Product collection where the Product name ends in ‘d’.

```

}
> db.createCollection("faculty");
{ "ok" : 1 }
> db.faculty.insert({_id:1,name:"Dr. Balaraman Ravindran",designation:"Professor",department:"CSE",age:45,salary:100000,specialization:['python','mysql','sklearn', 'tensorflow']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:2,name:"Dr. Mahadev Ghorki",designation:"Assistant Professor",department:"CSE",age:35,salary:80000,specialization:['python','numpy','sklearn', 'tensorflow', 'java']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:3,name:"Dr. Praveen Borade",designation:"Associate Professor",department:"ME",age:40,salary:75000,specialization:['autocad', 'aerodynamics', 'thermal physics']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:4,name:"Dr. Madhav Nayak",designation:"Assistant Professor",department:"ME",age:37,salary:95000,specialization:['autocad', 'flight-dynamics', 'Finite Element Analysis']});
WriteResult({ "nInserted" : 1 })
> db.faculty.aggregate ( {$match:{department:"ME"}}, {$group : {_id : "$designation", AverageSal :{$avg:"$salary"} } }, { $match:{AverageSal:{$gt:50000}}});
{ "_id" : "Associate Professor", "AverageSal" : 75000 }
{ "_id" : "Assistant Professor", "AverageSal" : 95000 }
> db.createCollection("product");
{ "ok" : 1 }
> db.product.insert({pid:1,pname:"keyboard",mdate:2001,price:1800,quantity:2});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:2,pname:"mouse",mdate:2005,price:1500,quantity:5});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:3,pname:"monitor",mdate:2015,price:10000,quantity:9});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:4,pname:"motherboard",mdate:2021,price:15000,quantity:4});
WriteResult({ "nInserted" : 1 })
> db.product.find({}, {pname:1,_id:0})
[{"pname" : "keyboard" }
 {"pname" : "mouse" }
 {"pname" : "monitor" }
 {"pname" : "motherboard" }]
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
> db.product.find({price:{$ne:15000}},{pname:1,_id:0});
{ "pname" : "keyboard" }
{"-----" : "-----" }

```

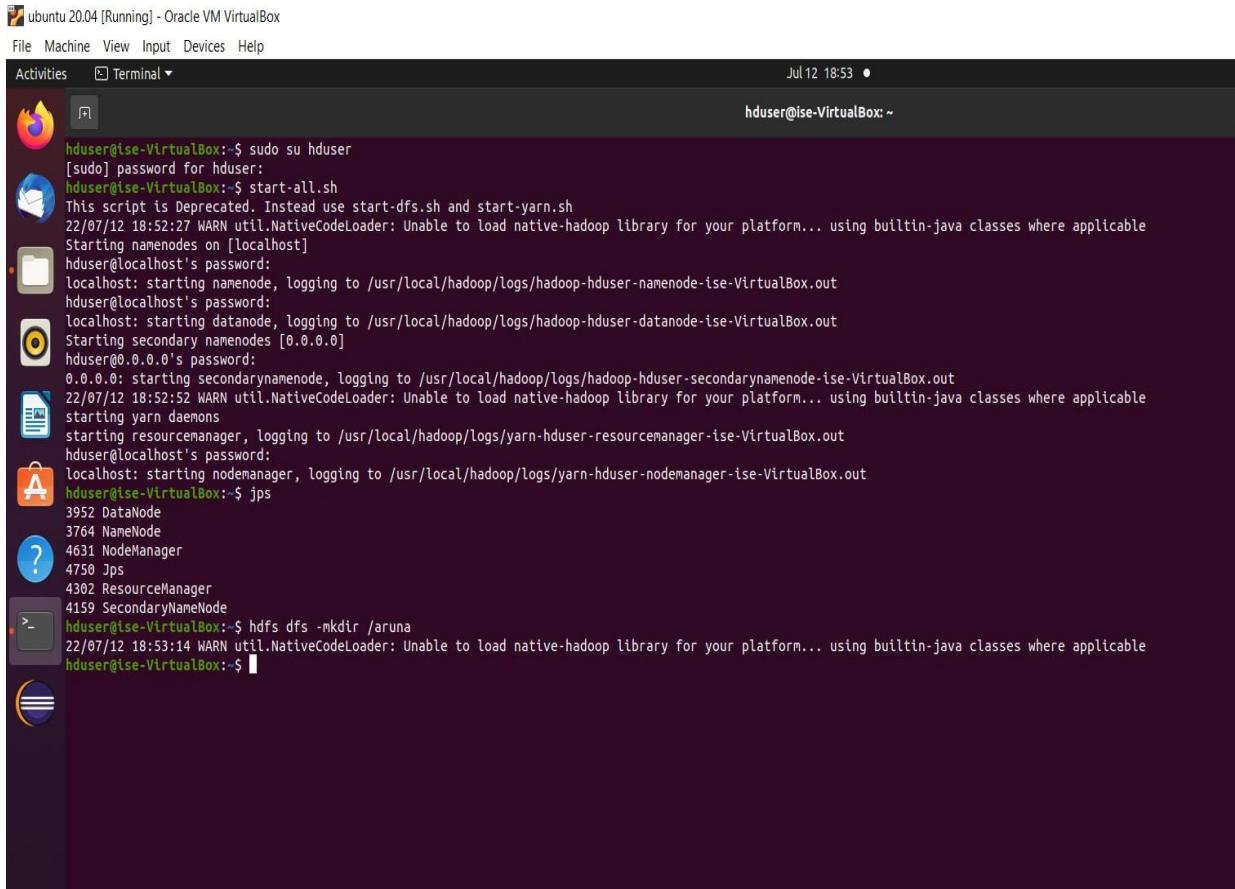
3)Create a mongodb collection Hospital. Demonstrate the following by choosing fields of your choice.

1. Insert three documents
2. Use Arrays(Use Pull and Pop operation)
3. Use Index
4. Use Cursors
5. Updation

```
[ "pname" : "motherboard" ]
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
> db.product.find({price:{$ne:15000}},{pname:1,_id:0});
{ "pname" : "keyboard" }
{ "pname" : "mouse" }
{ "pname" : "monitor" }
> db.product.find({$and:[{quantity:$eq:9},{pname:$eq:"monitor"}]}),{pname:1,_id:0})
{ "pname" : "monitor" }
> db.product.find({pname:/d$/},{pname:1,quantity:1,_id:0})
{ "pname" : "keyboard", "quantity" : 2 }
{ "pname" : "motherboard", "quantity" : 4 }
> db.createCollection("hospital");
{ "ok" : 1 }
> db.hospital.insert({_id:1, Name: "Anshuman Agarwal", age:23, diseases:["fever", "diarrhoea", "wheezing", "gastritis"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.insert({_id:2, Name: "Pinky Chaubey", age:35, diseases:["fever", "nausea", "food infection", "indigestion", "kidney stones"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.insert({_id:3, Name: "Amresh Chowpati", age:63, diseases:["hyperglycemia", "diabetes mellitus", "food poisoning", "cold"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.updateMany({},{$pull:{diseases:"fever"}});
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 2 }
> db.hospital.updateOne({_id:1},{$pop:{diseases:-1}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.hospital.find({"diseases.2":"nausea"});
> db.hospital.find({"diseases.1":"nausea"});
> d.hospital.find();
uncaught exception: ReferenceError: d is not defined :
@(shell):1:1
> db.hospital.find();
{ "_id" : 1, "Name" : "Anshuman Agarwal", "age" : 23, "diseases" : [ "wheezing", "gastritis" ] }
{ "_id" : 2, "Name" : "Pinky Chaubey", "age" : 35, "diseases" : [ "nausea", "food infection", "indigestion", "kidney stones" ] }
{ "_id" : 3, "Name" : "Amresh Chowpati", "age" : 63, "diseases" : [ "hyperglycemia", "diabetes mellitus", "food poisoning", "cold" ] }
> db.hospital.find({"diseases.0":"nausea"});
{ "_id" : 2, "Name" : "Pinky Chaubey", "age" : 35, "diseases" : [ "nausea", "food infection", "indigestion", "kidney stones" ] }
> db.hospital.update({_id:3},{$set:{'diseases.1':'sarscov'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> |
```

## **BDA LAB 4**

### 4. Hadoop Installation



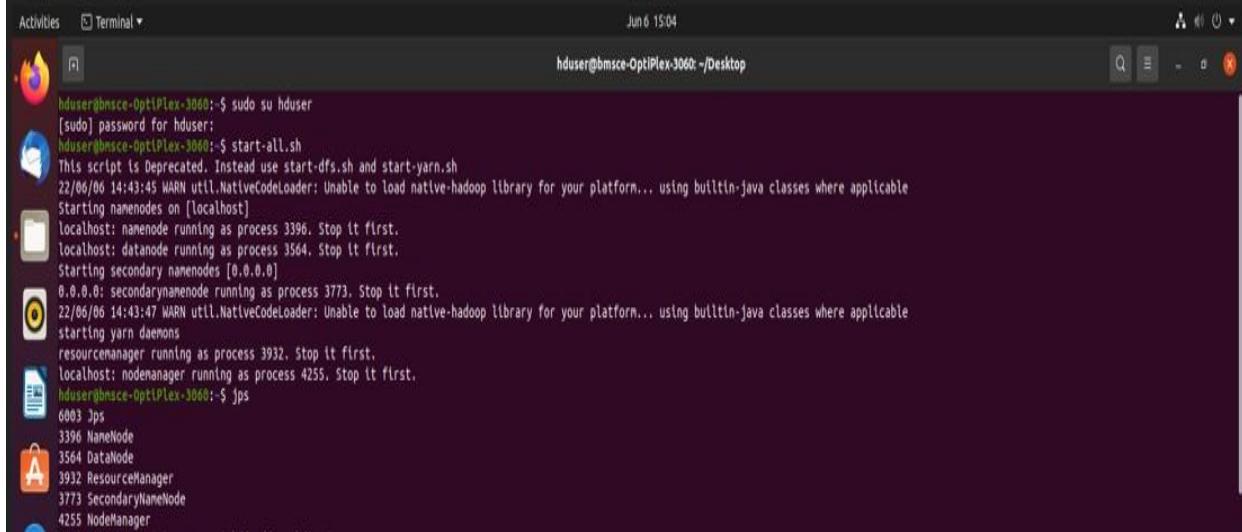
The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal". The terminal content shows the following command-line session:

```
ubuntu 20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jul 12 18:53 •
hduser@ise-VirtualBox:~$ sudo su hduser
[sudo] password for hduser:
hduser@ise-VirtualBox:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
22/07/12 18:52:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-ise-VirtualBox.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-ise-VirtualBox.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-ise-VirtualBox.out
22/07/12 18:52:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-ise-VirtualBox.out
hduser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-ise-VirtualBox.out
hduser@ise-VirtualBox:~$ jps
3952 DataNode
3764 NameNode
4631 NodeManager
4750 Jps
4302 ResourceManager
4159 SecondaryNameNode
hduser@ise-VirtualBox:~$ hdfs dfs -mkdir /aruna
22/07/12 18:53:14 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hduser@ise-VirtualBox:~$
```

## **BDA LAB 5**

5. Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)

### **1. jps**



```
hduser@bnse-OptiPlex-3060:~$ sudo su hduser
[sudo] password for hduser:
hduser@bnse-OptiPlex-3060:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
22/06/06 14:43:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: namenode running as process 3396. Stop it first.
localhost: datanode running as process 3564. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 3773. Stop it first.
22/06/06 14:43:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
starting yarn daemons
resourcemanager running as process 3932. Stop it first.
localhost: nodemanager running as process 4255. Stop it first.
hduser@bnse-OptiPlex-3060:~$ jps
6003 Jps
3396 NameNode
3564 DataNode
3932 ResourceManager
3773 SecondaryNameNode
4255 NodeManager
```

### **2. mkdir**

```
4159 SecondaryNameNode
hduser@ise-VirtualBox:~$ hdfs dfs -mkdir /aruna
```

### **3. ls**

```
hduser@ise-VirtualBox:~$ hdfs dfs -ls /
22/07/12 18:54:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform.
Found 17 items
-rw-r--r-- 1 hduser supergroup      59 2022-06-25 16:14 /DeptName.txt
-rw-r--r-- 1 hduser supergroup      50 2022-06-25 16:14 /DeptStrength.txt
drwxr-xr-x - hduser supergroup      0 2022-07-11 10:25 /InputDir
drwxr-xr-x - hduser supergroup      0 2022-07-11 10:27 /OutputDir
drwxr-xr-x - hduser supergroup      0 2022-07-12 18:53 /aruna
drwxr-xr-x - hduser supergroup      0 2022-05-13 11:05 /hbase
drwxr-xr-x - hduser supergroup      0 2022-06-25 15:45 /input
drwxr-xr-x - hduser supergroup      0 2022-06-22 19:15 /op.txt
drwxr-xr-x - hduser supergroup      0 2022-06-25 15:50 /op1.txt
drwxr-xr-x - hduser supergroup      0 2022-06-25 16:00 /op2
drwxr-xr-x - hduser supergroup      0 2022-06-21 18:09 /rgs
drwxr-xr-x - hduser supergroup      0 2022-06-21 18:10 /rgs1
-rw-r--r-- 1 hduser supergroup     27 2022-06-22 19:13 /sample.txt
drwxr-xr-x - hduser supergroup      0 2022-06-21 17:32 /test
drwxr-xr-x - hduser supergroup      0 2022-06-22 19:06 /test1
drwxrwxr-x - hduser supergroup      0 2022-05-13 11:04 /tmp
drwxr-xr-x - hduser supergroup      0 2022-05-13 11:05 /user
```

#### 4. put

```
hduser@bmse-OptiPlex-3060:~$ hdfs dfs -put /home/hduser/Desktop/6b.txt /Kusum/WC.txt
22/06/06 14:46:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hduser@bmse-OptiPlex-3060:~$ hdfs dfs -cat /Kusum/WC.txt
22/06/06 14:47:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hello from 6b
```

D

B

B

#### 5. copyFromLocal

```
hduser@bmse-OptiPlex-3060:~$ hdfs dfs -put /home/hduser/Desktop/6b.txt /Kusum/newWC.txt
22/06/06 14:50:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hduser@bmse-OptiPlex-3060:~$ hdfs dfs -cat /Kusum/newWC.txt
22/06/06 14:50:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hello from 6b
```

D

B

B

#### 6. Get

##### i)

```
hduser@bmse-OptiPlex-3060:~$ hdfs dfs -get /Kusum/WC.txt /home/hduser/Downloads/newWC.txt
22/06/06 14:51:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hduser@bmse-OptiPlex-3060:~$ cd Downloads
hduser@bmse-OptiPlex-3060:~/Downloads$ cat newWC.txt
hello from 6b
```

D

B

B

##### ii)

```
hduser@bmse-OptiPlex-3060:~$ hdfs dfs -ls /Kusum/
22/06/06 14:54:04 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 hduser supergroup 23 2022-06-06 14:46 /Kusum/WC.txt
-rw-r--r-- 1 hduser supergroup 23 2022-06-06 14:58 /Kusum/newWC.txt
hduser@bmse-OptiPlex-3060:~$ hdfs dfs -getmerge /Kusum/WC.txt /Kusum/newWC.txt /home/hduser/Desktop/newmerge.txt
22/06/06 14:55:18 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hduser@bmse-OptiPlex-3060:~$ cd Desktop
hduser@bmse-OptiPlex-3060:~/Desktop$ cat newmerge.txt
hello from 6b
```

D

B

B

hello from 6b

D

B

B

##### iii)

```
hduser@bmse-OptiPlex-3060:~/Desktop$ hadoop fs -getfacl /Kusum/
22/06/06 14:56:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
# file: /Kusum
# owner: hduser
# group: supergroup
user::rwx
group::r-x
other::r-x
```

## 7. copyToLocal

```
hduser@bnscce:OptiPlex-3060:~/Desktop$ hdfs dfs -copyToLocal /Kusum/WC.txt /home/hduser/Desktop  
22/06/06 14:58:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
hduser@bnscce:OptiPlex-3060:~/Desktop$ cat WC.txt  
hello from GB  
  
D  
B  
  
B
```

## 8. cat

```
hduser@bnscce:OptiPlex-3060:~/Desktop$ hdfs dfs -cat /Kusum/WC.txt  
22/06/06 14:58:59 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
hello from GB  
  
D  
B  
  
B
```

## 9. Mv

```
hduser@bnscce:OptiPlex-3060:~/Desktop$ hadoop fs -mv /Kusum /FFF  
22/06/06 14:59:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
hduser@bnscce:OptiPlex-3060:~/Desktop$ hadoop fs -ls /FFF  
22/06/06 15:00:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Found 2 items  
drwxr-xr-x - hduser supergroup 0 2022-06-06 14:50 /FFF/Kusum  
-rw-r--r-- 1 hduser supergroup , 17 2022-06-04 10:00 /FFF/WC.txt
```

## 10. cp

```
hduser@bnscce:OptiPlex-3060:~/Desktop$ hadoop fs -cp /FFF/ /LLL  
22/06/06 15:09:34 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
hduser@bnscce:OptiPlex-3060:~/Desktop$ hadoop fs -ls /LLL  
22/06/06 15:10:07 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Found 2 items  
drwxr-xr-x - hduser supergroup 0 2022-06-06 15:09 /LLL/Kusum  
-rw-r--r-- 1 hduser supergroup 17 2022-06-04 10:00 /LLL/WC.txt  
hduser@bnscce:OptiPlex-3060:~/Desktop$ []
```

## **BDA LAB 6**

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 the NCDC data set.

b) find the mean max temperature for every month

AverageDriver

```
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);
    }
}

AverageMapper
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));
}
}
}

```

### AverageReducer

```

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));
    }
}
}

```

b) find the mean max temperature for every month

```
MeanMax
MeanMaxDriver.class
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);
    }
}
```

```
MeanMaxMapper.class
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));
}
}

MeanMaxReducer.class
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));
    }
}

```

```

hduser@bmsce-Precision-T1700:~/Desktop/temperature$ start-all.sh This
script is Deprecated. Instead use start-dfs.sh and start-yarn.shStarting
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:~/Desktop/temperature$ jps
6832 NodeManager
6498 ResourceManager
6339 SecondaryNameNode
4887 org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar
6954 Jps
6123 DataNode
5951 NameNode
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -le /
-le: Unknown command
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /
Found 31 items

```

File Type	Owner	Group	Size	Last Modified	Last Accessed	Path
drwxr-xr-x	- hduser	supergroup	0	2022-06-06	12:35	/CSE
drwxr-xr-x	- hduser	supergroup	0	2022-06-06	12:23	/FFF
drwxr-xr-x	- hduser	supergroup	0	2022-06-06	12:36	/LLL
drwxr-xr-x	- hduser	supergroup	0	2022-06-20	12:06	/amit_bda
drwxr-xr-x	- hduser	supergroup	0	2022-06-27	11:42	/amit_lab
drwxr-xr-x	- hduser	supergroup	0	2022-06-03	14:52	/bharath
drwxr-xr-x	- hduser	supergroup	0	2022-06-03	14:43	/bharath035
drwxr-xr-x	- hduser	supergroup	0	2022-06-24	14:54	/chi
drwxr-xr-x	- hduser	supergroup	0	2022-05-31	10:21	/example
drwxr-xr-x	- hduser	supergroup	0	2022-06-01	15:13	/foldernew
drwxr-xr-x	- hduser	supergroup	0	2022-06-06	15:04	/hemang061

drwxr-xr-x	-	hduser	supergroup	0	2022-06-20	15:16	/input_Aruna
drwxr-xr-x	-	hduser	supergroup	0	2022-06-03	12:27	/irfan
drwxr-xr-x	-	hduser	supergroup	0	2022-06-22	10:44	/wde
drwxr-xr-x	-	hduser	supergroup	0	2022-06-27	13:03	/mapreducejoin_ami
drwxr-xr-x	-	hduser	supergroup	0	2022-06-22	15:32	/muskan
drwxr-xr-x	-	hduser	supergroup	0	2022-06-22	15:06	/muskan_op
drwxr-xr-x	-	hduser	supergroup	0	2022-06-22	15:35	/muskan_output
drwxr-xr-x	-	hduser	supergroup	0	2022-06-06	15:04	/new_folder
drwxr-xr-x	-	hduser	supergroup	0	2022-05-31	10:26	/one
drwxr-xr-x	-	hduser	supergroup	0	2022-06-24	15:30	/out55
drwxr-xr-x	-	hduser	supergroup	0	2022-06-20	12:17	/output
drwxr-xr-x	-	hduser	supergroup	0	2022-06-27	13:04	/output_TOPn
drwxr-xr-x	-	hduser	supergroup	0	2022-06-27	12:14	/output_Topn
drwxr-xr-x	-	hduser	supergroup	0	2022-06-24	12:42	/r1
drwxr-xr-x	-	hduser	supergroup	0	2022-06-24	12:24	/rgs

```

drwxr-xr-x - hduser supergroup 0 2022-06-03 12:08 /Aruna
drwxrwxr-x - hduser supergroup 0 2019-08-01 16:19 /tmp
drwxr-xr-x - hduser supergroup 0 2019-08-01 16:03 /user
drwxr-xr-x - hduser supergroup 0 2022-06-01 09:46 /user1
-rw-r--r-- 1 hduser supergroup 2436 2022-06-24 12:17 /wc.jar
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -mkdir /Aruna_temperature
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -put ./1901 /Aruna_temperature
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -put ./1902 /Aruna_temperature
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /Aruna_temperature
Found 2 items
-rw-r--r-- 1 hduser supergroup 888190 2022-06-27 14:47 /Aruna_temperature/1901
-rw-r--r-- 1 hduser supergroup 888978 2022-06-27 14:47 /Aruna_temperature/1902
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hadoop jar ./avgtemp.jar AverageDriver
/Aruna_temperature/1901 /Aruna_temperature/output/
Exception in thread "main" java.lang.ClassNotFoundException: AverageDriverat
java.net.URLClassLoader.findClass(URLClassLoader.java:382)
at java.lang.ClassLoader.loadClass(ClassLoader.java:418) at
java.lang.ClassLoader.loadClass(ClassLoader.java:351) at
java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:348)
at org.apache.hadoop.util.RunJar.run(RunJar.java:214) at
org.apache.hadoop.util.RunJar.main(RunJar.java:136)
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hadoop jar ./avgtemp.jar
temperature.AverageDriver /Aruna_temperature/1901 /Aruna_temperature/output/ 22/06/27
14:53:27 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session.id
22/06/27 14:53:27 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker,
sessionId=

```

```

22/06/27 14:53:27 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
22/06/27 14:53:27 INFO input.FileInputFormat: Total input paths to process : 122/06/27
14:53:27 INFO mapreduce.JobSubmitter: number of splits:1
22/06/27 14:53:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local254968295_0001
22/06/27 14:53:28 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/27 14:53:28 INFO mapreduce.Job: Running job: job_local254968295_0001
22/06/27 14:53:28 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/27 14:53:28 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Starting task: attempt_local254968295_0001_m_000000_0
22/06/27 14:53:28 INFO mapred.Task: Using ResourceCalculatorProcessTree :[ ]
22/06/27 14:53:28 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/Aruna_temperature/1901:0+888190
22/06/27 14:53:28 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/27 14:53:28 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/27 14:53:28 INFO mapred.MapTask: soft limit at 83886080
22/06/27 14:53:28 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/27 14:53:28 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/27 14:53:28 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/27 14:53:28 INFO mapred.LocalJobRunner:
22/06/27 14:53:28 INFO mapred.MapTask: Starting flush of map output
22/06/27 14:53:28 INFO mapred.MapTask: Spilling map output
22/06/27 14:53:28 INFO mapred.MapTask: bufstart = 0; bufend = 59076; bufvoid = 104857600
22/06/27 14:53:28 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26188144(104752576);
length = 26253/6553600
22/06/27 14:53:28 INFO mapred.MapTask: Finished spill 0

```

```
22/06/27 14:53:28 INFO mapred.Task: Task:attempt_local254968295_0001_m_000000_0 is done. And is inthe  
process of committing.  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: map  
22/06/27 14:53:28 INFO mapred.Task: Task 'attempt_local254968295_0001_m_000000_0' done.  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Finishing task: attempt_local254968295_0001_m_000000_0  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: map task executor complete.  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Waiting for reduce tasks  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Starting task: attempt_local254968295_0001_r_000000_0  
22/06/27 14:53:28 INFO mapred.Task: Using ResourceCalculatorProcessTree :[]  
22/06/27 14:53:28 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:  
org.apache.hadoop.mapreduce.task.reduce.Shuffle@262cb2a9  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: MergerManager:  
memoryLimit=349752512,maxSingleShuffleLimit=87438128, mergeThreshold=230836672,  
ioSortFactor=10, memToMemMergeOutputsThreshold=10  
22/06/27 14:53:28 INFO reduce.EventFetcher: attempt_local254968295_0001_r_000000_0 Thread started:  
EventFetcher for fetching Map Completion Events  
22/06/27 14:53:28 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map  
attempt_local254968295_0001_m_000000_0 decomp: 72206 len: 72210 to MEMORY  
22/06/27 14:53:28 INFO reduce.InMemoryMapOutput: Read 72206 bytes from map-output for  
attempt_local254968295_0001_m_000000_0  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 72206,  
inMemoryMapOutputs.size() -> 1, commitMemory > 0, usedMemory ->72206  
22/06/27 14:53:28 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-  
disk map-outputs  
22/06/27 14:53:28 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 14:53:28 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of totalsize:  
72199 bytes  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merged 1 segments, 72206 bytes to disk to satisfy  
reduce memory limit  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merging 1 files, 72210 bytes from disk  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce  
22/06/27 14:53:28 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 14:53:28 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of totalsize:  
72199 bytes
```

```
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merged 1 segments, 72206 bytes to disk to satisfy  
reduce memory limit  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merging 1 files, 72210 bytes from disk  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce  
22/06/27 14:53:28 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 14:53:28 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of totalsize:  
72199 bytes  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 14:53:28 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use  
mapreduce.job.skiprecords  
22/06/27 14:53:28 INFO mapred.Task: Task:attempt_local254968295_0001_r_000000_0 is done. And is inthe  
process of committing  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 14:53:28 INFO mapred.Task: Task attempt_local254968295_0001_r_000000_0 is allowed to commit  
now  
22/06/27 14:53:28 INFO output.FileOutputCommitter: Saved output of task  
'attempt_local254968295_0001_r_000000_0' to hdfs://localhost:54310/Aruna_temperature/output/_temporary/0/  
task_local254968295_0001_r_00000022/06/27 14:53:28 INFO mapred.LocalJobRunner: reduce > reduce  
22/06/27 14:53:28 INFO mapred.Task: Task 'attempt_local254968295_0001_r_000000_0' done.  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Finishing task: attempt_local254968295_0001_r_000000_0  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: reduce task executor complete.  
22/06/27 14:53:29 INFO mapreduce.Job: Job job_local254968295_0001 running in uber mode : false  
22/06/27 14:53:29 INFO mapreduce.Job: map 100% reduce 100%  
22/06/27 14:53:29 INFO mapreduce.Job: Job job_local254968295_0001 completed successfully  
22/06/27 14:53:29 INFO mapreduce.Job: Counters: 38  
File System Counters  
FILE: Number of bytes read=153102
```

```

FILE: Number of bytes written=723014
FILE: Number of read operations=0 FILE:
Number of large read operations=0FILE:
Number of write operations=0 HDFS:
Number of bytes read=1776380 HDFS:
Number of bytes written=8
HDFS: Number of read operations=13
HDFS: Number of large read operations=0
HDFS: Number of write operations=4
Map-Reduce Framework
Map input records=6565
Map output
records=6564Map output
bytes=59076
Map output materialized bytes=72210
Input split bytes=112
Combine input records=0
Combine output records=0
Reduce input groups=1
Reduce shuffle
bytes=72210Reduce input
records=6564 Reduce output
records=1 Spilled
Records=13128 Shuffled
Maps =1
Failed Shuffles=0 Merged
Map outputs=1 GC time
elapsed (ms)=55CPU
time spent (ms)=0
Physical memory (bytes) snapshot=0
Virtual memory (bytes) snapshot=0
Total committed heap usage (bytes)=999292928
Shuffle Errors
BAD_ID=0
CONNECTION
=0
IO_ERROR=0
WRONG_LEN
GTH=0
WRONG_MAP
=0


---


WRONG_RED
UCE=0
File Input Format Counters
Bytes Read=888190
File Output Format Counters
Bytes Written=8
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /Aruna_temperature/output/Found
2 items
-rw-r--r--    1 hduser supergroup          0 2022-06-27 14:53 /Aruna_temperature/output/_SUCCESS
-rw-r--r--    1 hduser supergroup          8 2022-06-27 14:53 /Aruna_temperature/output/part-r-00000
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -cat /Aruna_temperature/output/part-r-00000
1901 46
hduser@bmsce-Precision-T1700:~/Desktop/temperature$
```

---

## **BDA LAB 7**

7. For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top n maximum occurrences of words.

Driver-TopN.class

```
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);
    }

    public static 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);
    }
}
}
```

## TopNCombiner.class

```
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));
    }
}
```

## TopNMapper.class

```
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);
        }
    }
}
```

```

        }
    }
}

TopNReducer.class
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));
        }
    }
}

```

```

hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -mkdir /Aruna_topn hduser@bmsce-
Precision-T1700:~/Desktop/temperature$ hdfs dfs -put ./input.txt /Aruna_topn/hduser@bmsce-
Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /Aruna_topn/
Found 1 items
-rw-r--r-- 1 hduser supergroup          103 2022-06-27 15:43 /Aruna_topn/input.txt
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hadoop jar topn.jar TopNDriver
/Aruna_topn/input.txt /Aruna_topn/output
Exception in thread "main" java.lang.ClassNotFoundException: TopNDriverat
java.net.URLClassLoader.findClass(URLClassLoader.java:382)
at java.lang.ClassLoader.loadClass(ClassLoader.java:418) at
java.lang.ClassLoader.loadClass(ClassLoader.java:351) at
java.lang.Class.forName0(Native Method)
at java.lang.Class.forName(Class.java:348)
at org.apache.hadoop.util.RunJar.run(RunJar.java:214) at
org.apache.hadoop.util.RunJar.main(RunJar.java:136)
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hadoop jar topn.jar topn.TopNDriver
/Aruna_topn/input.txt /Aruna_topn/output
22/06/27 15:45:22 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/27 15:45:22 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker,
sessionId=
22/06/27 15:45:22 INFO input.FileInputFormat: Total input paths to process : 122/06/27
15:45:22 INFO mapreduce.JobSubmitter: number of splits:1
22/06/27 15:45:22 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local691635730_0001
22/06/27 15:45:22 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/27 15:45:22 INFO mapreduce.Job: Running job: job_local691635730_0001
22/06/27 15:45:22 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/27 15:45:22 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Starting task: attempt_local691635730_0001_m_000000_0
22/06/27 15:45:22 INFO mapred.Task: Using ResourceCalculatorProcessTree :[ ]
22/06/27 15:45:22 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/Aruna_topn/input.txt:0+103
22/06/27 15:45:22 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/27 15:45:22 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/27 15:45:22 INFO mapred.MapTask: soft limit at 83886080
22/06/27 15:45:22 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/27 15:45:22 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/27 15:45:22 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/27 15:45:22 INFO mapred.LocalJobRunner:

```

```

22/06/27 15:45:22 INFO mapred.MapTask: Starting flush of map output
22/06/27 15:45:22 INFO mapred.MapTask: Spilling map output
22/06/27 15:45:22 INFO mapred.MapTask: bufstart = 0; bufend = 187; bufvoid = 104857600
22/06/27 15:45:22 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214316(104857264);
length = 81/6553600
22/06/27 15:45:22 INFO mapred.MapTask: Finished spill 0
22/06/27 15:45:22 INFO mapred.Task: Task:attempt_local691635730_0001_m_000000_0 is done. And is inthe
process of committing
22/06/27 15:45:22 INFO mapred.LocalJobRunner: map
22/06/27 15:45:22 INFO mapred.Task: Task 'attempt_local691635730_0001_m_000000_0' done.
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Finishing task: attempt_local691635730_0001_m_000000_0
22/06/27 15:45:22 INFO mapred.LocalJobRunner: map task executor complete.
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Waiting for reduce tasks
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Starting task: attempt_local691635730_0001_r_000000_0
22/06/27 15:45:22 INFO mapred.Task: Using ResourceCalculatorProcessTree :[ ]

```

```
22/06/27 15:45:22 INFO mapred.Reduce Task: Using ShuffleConsumerPlugin:  
org.apache.hadoop.mapreduce.task.reduce.Shuffle@40a5e65a  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: MergerManager:  
memoryLimit=334338464,maxSingleShuffleLimit=83584616, mergeThreshold=220663392,  
ioSortFactor=10, memToMemMergeOutputsThreshold=10  
22/06/27 15:45:22 INFO reduce.EventFetcher: attempt_local691635730_0001_r_000000_0 Thread started:  
EventFetcher for fetching Map Completion Events  
22/06/27 15:45:22 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map  
attempt_local691635730_0001_m_000000_0 decomp: 231 len: 235 to MEMORY  
22/06/27 15:45:22 INFO reduce.InMemoryMapOutput: Read 231 bytes from map-output for  
attempt_local691635730_0001_m_000000_0  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 231,  
inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->231  
22/06/27 15:45:22 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and On-disk map-outputs  
22/06/27 15:45:22 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 15:45:22 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of totalsize: 226  
bytes  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: Merged 1 segments, 231 bytes to disk to satisfy  
reduce memory limit  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: Merging 1 files, 235 bytes from disk  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce  
22/06/27 15:45:22 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 15:45:22 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of totalsize: 226  
bytes  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 15:45:22 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use  
mapreduce.job.skiprecords  
22/06/27 15:45:23 INFO mapred.Task: Task:attempt_local691635730_0001_r_000000_0 is done. And is in the  
process of committing  
22/06/27 15:45:23 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 15:45:23 INFO mapred.Task: Task attempt_local691635730_0001_r_000000_0 is allowed to commit  
now  
22/06/27 15:45:23 INFO output.FileOutputCommitter: Saved output of task  
'attempt_local691635730_0001_r_000000_0' to hdfs://localhost:54310/Aruna_topn/output/_temporary/0/  
task_local691635730_0001_r_00000022/06/27 15:45:23 INFO mapred.LocalJobRunner: reduce >  
reduce
```

```
22/06/27 15:45:23 INFO mapred.Task: Task 'attempt_local691635730_0001_r_000000_0' done.  
22/06/27 15:45:23 INFO mapred.LocalJobRunner: Finishing task: attempt_local691635730_0001_r_000000_0  
22/06/27 15:45:23 INFO mapred.LocalJobRunner: reduce task executor complete.  
22/06/27 15:45:23 INFO mapreduce.Job: Job job_local691635730_0001 running in uber mode : false  
22/06/27 15:45:23 INFO mapreduce.Job: map 100% reduce 100%  
22/06/27 15:45:23 INFO mapreduce.Job: Job job_local691635730_0001 completed successfully  
22/06/27 15:45:23 INFO mapreduce.Job: Counters: 38  
File System Counters  
FILE: Number of bytes read=18078  
FILE: Number of bytes written=516697  
FILE: Number of read operations=0  
FILE: Number of large read operations=0  
FILE: Number of write operations=0 HDFS:  
Number of bytes read=206  
HDFS: Number of bytes written=105  
HDFS: Number of read operations=13  
HDFS: Number of large read operations=0  
HDFS: Number of write operations=4  
Map-Reduce Framework
```

```

Map input records=6
Map output
records=21 Map output
bytes=187
Map output materialized bytes=235
Input split bytes=110
Combine input records=0
Combine output
records=0Reduce input
groups=15 Reduce shuffle
bytes=235Reduce input
records=21 Reduce output
records=15Spilled
Records=42 Shuffled Maps
=1
Failed Shuffles=0 Merged
Map outputs=1 GC time
elapsed (ms)=42CPU
time spent (ms)=0
Physical memory (bytes) snapshot=0
Virtual memory (bytes) snapshot=0
Total committed heap usage (bytes)=578289664
Shuffle Errors
BAD_ID=0
CONNECTION
=0
IO_ERROR=0
WRONG_LEN
GTH=0
WRONG_MAP
=0
WRONG_RED
UCE=0
File Input Format Counters
Bytes Read=103
File Output Format Counters
Bytes Written=105
hduser@bmse-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /Aruna_topn/output/Found
2 items
-rw-r--r-- 1 hduser supergroup          0 2022-06-27 15:45 /Aruna_topn/output/_SUCCESS
-rw-r--r-- 1 hduser supergroup 105 2022-06-27 15:45 /Aruna_topn/output/part-r-00000
hduser@bmse-Precision-T1700:~/Desktop/temperature$ hdfs dfs -cat /Aruna_topn/output/part-r-00000
hadoop 4
op
is
am 2
hi 1
im 1
is 1
there 1
bye 1
learning 1
awesome 1
love 1
Aruna 1
cool 1
and 1
using 1

```

## **BDA LAB 8**

### 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);
}
}
}

// 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]));
}

}

//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);
}
} }
```

```
hduser@tse-VirtualBox:~$ hadoop jar MapReduceJoin.jar /aruna/DeptName.txt /aruna/DeptStrength.txt /aruna/output
```

```
22/06/27 15:12:24 INFO mapred.MapTask: soft limit at 83886080
22/06/27 15:12:24 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/27 15:12:24 INFO mapred.MapTask: kvstart = 26214396; length =
655360022/06/27 15:12:24 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/27 15:12:24 INFO mapred.LocalJobRunner:
22/06/27 15:12:24 INFO mapred.MapTask: Starting flush of map
output22/06/27 15:12:24 INFO mapred.MapTask: Spilling map output
22/06/27 15:12:24 INFO mapred.MapTask: bufstart = 0; bufend = 54; bufvoid = 104857600
22/06/27 15:12:24 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214384(104857536); length = 13/6553600
22/06/27 15:12:24 INFO mapred.MapTask: Finished spill 0
22/06/27 15:12:24 INFO mapred.Task: Task:attempt_local1238804660_0001_m_000001_0 is done. And is
inthe process of committing
22/06/27 15:12:24 INFO mapred.LocalJobRunner: hdfs://
localhost:54310/kusum_join/DeptStrength.txt:0+50
22/06/27 15:12:24 INFO mapred.Task: Task 'attempt_local1238804660_0001_m_000001_0'
done. 22/06/27 15:12:24 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1238804660_0001_m_000001_0
22/06/27 15:12:24 INFO mapred.LocalJobRunner: map task executor
complete. 22/06/27 15:12:24 INFO mapred.LocalJobRunner: Waiting for reduce
tasks
22/06/27 15:12:24 INFO mapred.LocalJobRunner: Starting task:
attempt_local1238804660_0001_r_000000_022/06/27 15:12:24 INFO mapred.Task: Using
ResourceCalculatorProcessTree : []
22/06/27 15:12:24 INFO mapred.ReduceTask: Using
ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@45cb1c
22/06/27 15:12:24 INFO reduce.MergeManagerImpl: MergerManager:
memoryLimit=334338464, maxSingleShuffleLimit=83584616, mergeThreshold=220663392,
ioSortFactor=10, memToMemMergeOutputsThreshold=10
22/06/27 15:12:24 INFO reduce.EventFetcher: attempt_local1238804660_0001_r_000000_0 Thread
started:EventFetcher for fetching Map Completion Events
22/06/27 15:12:24 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of
mapattempt_local1238804660_0001_m_000001_0 decomp: 64 len: 68 to MEMORY
22/06/27 15:12:24 INFO reduce.InMemoryMapOutput: Read 64 bytes from map-output
forattempt_local1238804660_0001_m_000001_0
22/06/27 15:12:24 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size:
64, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->64
22/06/27 15:12:24 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of
mapattempt_local1238804660_0001_m_000000_0 decomp: 73 len: 77 to MEMORY
22/06/27 15:12:24 INFO reduce.InMemoryMapOutput: Read 73 bytes from map-output
forattempt_local1238804660_0001_m_000000_0
22/06/27 15:12:24 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size:
73, inMemoryMapOutputs.size() -> 2, commitMemory -> 64, usedMemory ->137
22/06/27 15:12:24 INFO reduce.EventFetcher: EventFetcher is interrupted..
Returning22/06/27 15:12:24 INFO mapred.LocalJobRunner: 2 / 2 copied.
22/06/27 15:12:24 INFO reduce.MergeManagerImpl: finalMerge called with 2 in-memory map-outputs and
On-disk map-outputs
22/06/27 15:12:24 INFO mapred.Merger: Merging 2 sorted segments
22/06/27 15:12:24 INFO mapred.Merger: Down to the last merge-pass, with 2 segments left of
totalsize: 121 bytes
22/06/27 15:12:24 INFO reduce.MergeManagerImpl: Merged 2 segments, 137 bytes to disk to
satisfyreduce memory limit
22/06/27 15:12:24 INFO reduce.MergeManagerImpl: Merging 1 files, 139 bytes from disk
22/06/27 15:12:24 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into
reduce22/06/27 15:12:24 INFO mapred.Merger: Merging 1 sorted segments
22/06/27 15:12:24 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of
totalsize: 127 bytes
22/06/27 15:12:24 INFO mapred.LocalJobRunner: 2 / 2 copied.
```

```

FILE: Number of bytes read=26370
FILE: Number of bytes
written=782871FILE: Number of read
operations=0
FILE: Number of large read
operations=0FILE: Number of write
operations=0 HDFS: Number of bytes
read=277
HDFS: Number of bytes written=85
HDFS: Number of read operations=28
HDFS: Number of large read
operations=0HDFS: Number of write
operations=5
Map-Reduce
FrameworkMap input
records=8 Map output
records=8Map output
bytes=117
Map output materialized
bytes=145Input split bytes=443
Combine input records=0
Combine output
records=0Reduce input
groups=4 Reduce shuffle
bytes=145Reduce input
records=8 Reduce output
records=4 Spilled
Records=16 Shuffled Maps
=2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed
(ms)=2CPU time spent
(ms)=0
Physical memory (bytes)
snapshot=0Virtual memory (bytes)
snapshot=0
Total committed heap usage
(bytes)=913833984Shuffle Errors
BAD ID=0
CONNECTION=0
IO_ERROR=0
WRONG LENGTH=
OWRONG_MAP=0
WRONG_REDUCE=0
File Input Format
CountersBytes Read=0
File Output Format
CountersBytes Written=85

```

<b>Dept_ID</b>	<b>Total_Employee</b>	<b>Dept_Name</b>
A11	50	Finance
B12	100	HR
C13	250	Manufacturing

## **BDA LAB 9**

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

```
scala> println("Hello World!");
Hello World!
```

```
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;
```

```
hadoop@wave:~/d.../hadoop_rises$ spark-shell -i countwords.scala
21/06/14 13:01:47 WARN Utils: Your hostname, wave-ubu resolves to a loopback address: 127.0.1.1; using
21/06/14 13:01:47 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/06/14 13:01:47 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... usi
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://192.168.2.7:4040
Spark context available as 'sc' (master = local[*], app id = local-1623655911213).
Spark session available as 'spark'.
wasn't: 6
what: 5
as: 7
she: 13
it: 23
he: 5
for: 6
her: 12
the: 30
was: 19
be: 8
It: 7
but: 11
had: 5
would: 7
in: 9
you: 6
that: 8
a: 9
or: 5
to: 20
I: 5
of: 6
and: 16
welcome to
```

## **BDA LAB 10**

**Using RDD and Flat Map count how many times each word appears in a file and write out a list of words whose count is strictly greater than n using Spark**

```
scala> val textfile = sc.textFile("/home/sam/Desktop/abc.txt")
textfile: org.apache.spark.rdd.RDD[String] = /home/sam/Desktop/abc.txt MapPartitionsRDD[8] at textFile at <console>:25

scala> val counts = textfile.flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey(_+_)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[11] at reduceByKey at <console>:26

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

scala> val sorted = ListMap(counts.collect.sortWith(_._2 > _._2): *)
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(hello -> 3, apple -> 2, unicorn -> 1, world -> 1)

scala> println(sorted)
ListMap(hello -> 3, apple -> 2, unicorn -> 1, world -> 1)
```