VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB-2 REPORT
on

BIG DATA ANALYTICS
(20CS6PEBDA)

Submitted by

MITHIL RAJ(1BM18CS086)

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 **MITHIL RAJ(1BM18CS086),** who is 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.

Nameof the Lab-Incharge **ANTARA ROY CHOUDARY**

Designation ASSISTANT PROFESSOR
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

LAB 1 MONGODB (CRUD OPERATION):-

I. CREATE DATABASE IN MONGODB

use myDB;

Confirm the existence of your database

Db;

To list all databases

Show dbs;

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(…);

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”,Grade:”VII”,Hobbies:”InternetSurfing”});

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”,Grade:”VII”},{$set:{Hobbies:”Skating”}},{upsert:true});

5. FIND METHOD

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

Db.Student.find({StudName:”Aryan David”});

({cond..},{columns.. column:1, columnname:0})

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});

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

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

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 ();

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

db.Student.find({StudName:/^M/}).pretty();

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

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

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

Db.Student.count();

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

Db.Student.find().sort({StudName:-1}).pretty();

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

This command used at the command prompt exports MongoDB JSON documents from

"Customers" collection in the "test" database into a CSV file "Output.txt" in the D:drive.

Mongoexport –host localhost –db Student –collection airlines –csv –out /home/hduser/Desktop/output.txt –fields "Year","Quarter"

V. Save Method :

Save() method will insert a new document, if the document with the _id does not exist. If it exists it will replace the exisiting document.

```
Db.Students.save({StudName:”Vamsi”, Grade:”VI”})
```

VI. Add a new field to existing Document:

```
db.Students.update({_id:4},{$set:{Location:”Network”}})
```

VII. Remove the field in an existing Document

```
db.Students.update({_id:4},{$unset:{Location:”Network”}})
```

VIII. Finding Document based on search criteria suppressing few fields

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

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

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

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

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

IX. to set a particular field value to NULL

```
db.Students.update({_id:3},{$set:{Location:null}})
```

X. Count the number of documents in Student Collections

```
db.Students.count()
```

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

```
db.Students.count({Grade:”VII”})
```

retrieve first 3 documents

```
db.Students.find({Grade:”VII”}).limit(3).pretty();
```

Sort the document in Ascending order

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

Note:

for desending order : db.Students.find().sort({StudName:-1}).pretty();

to Skip the 1st two documents from the Students Collections
db.Students.find().skip(2).pretty()

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’] })

To find those documents from the “food” collection which has the “fruits array” constitute of “grapes”, “mango” and “apple”.

Db.food.find ({fruits: [‘grapes’,’mango’,’apple’] }). Pretty().

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}})

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

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

Db.food.find ({“fruits”: {$size:2}})

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}})

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”]}})

update on Array:

array with apple

db.food.update({_id:3},{$set:{‘fruits.1’:’apple’}})

insert new key value pairs in the fruits array

db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}})

Note: perform query operations using – pop, addToSet, pullAll and pull

XIII. Aggregate Function :

Create a collection Customers with fields custID, AcctBal, AcctType.
Now group on “custID” and compute the sum of “AccBal”.

Db.Customers.aggregate ({$group : { _id : “$custID”,TotAccBal : {$sum:”$AccBal”} } });

match on AcctType:”S” then group on “CustID” and compute the sum of “AccBal”.

Db.Customers.aggregate ({$match:{AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal :

{$sum:”$AccBal”} } });

match on AcctType:”S” then group on “CustID” and compute the sum of “AccBal” and total balance greater than 1200.

Db.Customers.aggregate ({$match:{AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal : {$sum:”$AccBal”} } }, {$match:{TotAccBal:{$gt:1200}}});

Assignment:

Creation of Cursor:

Create Collection “Alphabets”

Insert Documents with fields “_id” and “alphabet”

use cursor to iterate through the “Alphabets” Collection.

NAME:MITHIL RAJ

USN:1BM19CS086

BDA LAB-1

LAB 2 MONGO DB (CRUD OPERATIONS):-

MONGO DB

1) Using MongoDB

i) Create a database for Students and Create a Student Collection

(_id,Name, USN,Semester, Dept_Name, CGPA, Hobbies(Set)).

> use Students

switched to db Students

ii) Insert required documents to the collection.

- db.Student.insert({Studname:"MITHIL RAJ",USN:"1BM19CS086",Semester: "VII",Dept_name:"Computer Science",CGPA:9.6,Hobbies:["Sleep","eat"]}); WriteResult({ "nInserted" : 1 })

> db.Student.insert({Studname:"NITHIN",USN:"1BM19CS106",Semester: "VI",Dept_name:"Computer Science",CGPA:8.6,Hobbies:["Sleep","eat"]}); WriteResult({ "nInserted" : 1 })

> db.Student.insert({Studname:"Hailey",USN:"1BM19CS015",Semester :"VIII",Dept_name:"Computer Science",CGPA:7.4,Hobbies:["Sleep","eat","repeat"]}); WriteResult({ "nInserted" : 1 })

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.

> db.Student.aggregate({$match:{Dept_name:"Computer Science"}},{$group:{_id:"$Semester",AvgCGPA:{$avg:"$CGPA"}}},{$match:{AvgCGPA:{$gt:7.5}}});

{ "_id" : "VIII", "AvgCGPA" : 8.6 }
{ "_id" : "VII", "AvgCGPA" : 8.533333333333333 }
{ "_id" : "VI", "AvgCGPA" : 8.266666666666667 }

iv) Command used to export MongoDB JSON documents from "Student" Collection into the "Students" database into a CSV file "Output.txt".

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

> db.createCollection("Bank");
{ "ok" : 1 }

1. Insert three documents

db.Bank.insert({_id:1,name:"Ramesh",state:"Gujarat",country:"India",language:["gujarati","marathi","english"]})

db.Bank.insert({_id:2,name:"Mahesh",state:"Gujarat",country:"India",language:["gujarati","marwadi","english"]})

db.Bank.insert({_id:3,name:"Ghelbhai",state:"Maharashta",country:"India",language:["marathi","marwadi","english"]})

2. Use Arrays(Use Pull and Pop operation)

db.Bank.update({_id: 1}, {$push: {language: "hindi"}})

db.Bank.update({_id: 2}, {$pull: {language: "english"}})

3. Use Index
4. Use Cursors
5. Updation

3) Consider a table "Students " with the following columns:
1. StudRollNo / _id
2. StudName
3. Grade
4. Hobbies
5. DOJ

Write MongoDB queries for the following:

1. To display only the students name from all the documents of the Students collection.

> db.Students.find({},{Studname:1,_id:0});
{ "Studname" : "mithil" }

{ "Studname" : "varun" }
{ "Studname" : "Lodi" }
{ "Studname" : "Modi" }
{ "Studname" : "Nithin" }

2. To display only the student name, grade as well as the identifier from the document of the Student collection where the _id column is 1.

```
> db.Students.find({_id:{$eq:ObjectId("625fd1171e24dbace73bd604")}
},{Studname:1,Grade:1,_id:1});
{ "_id" : ObjectId("625fd1171e24dbace73bd604"), "Studname" : "mithil",
"Grade" : "VII" }
```

3. To find those documents where the grade is not set to VIII.

```
> db.Students.find({Grade:{$ne:"VII"}});
{ "_id" : ObjectId("625fd11d1e24dbace73bd605"), "Studname" :
"varun", "Grade" : "VIII", "Hobbies" : [ "cricket" ], "DOJ" : "12/8/2021" }
{ "_id" : ObjectId("625fd1241e24dbace73bd606"), "Studname" :
"Lodi", "Grade" : "VIII", "Hobbies" : [ "Sleep" ], "DOJ" : "12/8/2021" }
{ "_id" : ObjectId("625fd12d1e24dbace73bd607"), "Studname" :
"Modi", "Grade" : "VI", "Hobbies" : [ "Sleep", "eat" ], "DOJ" : "12/7/2001"
}
```

4. To find those documents from the Students collection where the hobbies is set to 'cricket' and the student name is set to 'varun'.

```
> db.Student.find({Hobbies :{
$in:['cricket']},Studname:{$eq:"varun"}}).pretty ();
{
"_id" : ObjectId("625fd0771e24dbace73bd602"),
"Studname" : "varun",
"Grade" : "VIII",
"Hobbies" : [
"cricket"
],
"DOJ" : "12/8/2021"
}
```

5.To find documents from the Students collection where the student name ends in 'j'

```
> db.Student.find({Studname:/j$/}).pretty();
{
"_id" : ObjectId("625fd09b1e24dbace73bd603"),
"Studname" : "mithil",
"Grade" : "VII",
"Hobbies" : [
"cricket"
],
"DOJ" : "12/8/2021"
}
```

4) Using MongoDB,

i) Create a database for Faculty and Create a Faculty Collection(Faculty_id, Name, Designation ,Department, Age, Salary, Specialization(Set)).

```
> use faculty

switched to db faculty

> db.createCollection("Faculty");
{ "ok" : 1 }
```

iii) Insert required documents to the collection.

```
> db.Faculty.insert({Name:"NITHIN",Designation:"Teacher",Department:"
CSE",Age:90,Salary:40000,Specialization:["Eating","Talking","Web
dev"]});
WriteResult({ "nInserted" : 1 })

> db.Faculty.insert({Name:"KHUSHIL",Designation:"Teacher",Depart
ment:"MECH",Age:90,Salary:120000,Specialization:["Eating","Talking"
,"Web dev"]});
WriteResult({ "nInserted" : 1 })

> db.Faculty.insert({Name:"ugrasen",Designation:"Assisstant",Departm
ent:"MECH",Age:20,Salary:1000,Specialization:["Eating","Talking","We
b dev"]});
WriteResult({ "nInserted" : 1 })

>
db.Faculty.insert({Name:"JEEVAN",Designation:"Assisstant",Departmen
t:"MECH",Age:20,Salary:111000,Specialization:["Eating","Talking","We
b dev"]});
WriteResult({ "nInserted" : 1 })
```

iii) 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 6500.

```
> db.Faculty.aggregate({$match:{Department:"MECH"}},{$group:{_id:"$
Designation",AvgSAL:{$avg:"$Salary"}}},{$match:{AvgSAL:{$gt:6500}
}});
{ "_id" : "Assisstant", "AvgSAL" : 56000 }
{ "_id" : "Teacher", "AvgSAL" : 120000 }
```

NAME:MITHIL RAJ

USN:1BM19CS086

BDA LAB-2

LAB-3 CASSANDRA EMPLOYEE QUESTION:-

1. Program 1. Perform the following DB operations using Cassandra

2. Create a key space by name Employee

3. 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

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

8. Update the altered table to add project names.

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

OUTPUT:-

```
bmsce@bmsce-Precision-T1700:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE Employee_info WITH REPLICATION =
{'class':'SimpleStrategy','replication_factor':2};
cqlsh> DESCRIBE KEYSPACES;

employee_info  system_auth  employee134  tranzmetro        employee
students       system       students1    studentinfo       system_traces
system_schema  library      tranz        system_distributed  students2
```

```
cqlsh> USE Employee_info;

cqlsh:employee_info> CREATE TABLE EMPLOYEE_INFO(Emp_id int PRIMARY KEY,
Emp_name text,Designation text,DOJ timestamp,salary int,Dept_name text);

cqlsh:employee_info> INSERT INTO EMPLOYEE-

INFO(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) VALUES (1,'manny','senior
employee','2018-06-01','1000000','CSE');

SyntaxException: line 1:20 no viable alternative at input '-' (INSERT INTO [EMPLOYEE]-…)
cqlsh:employee_info> INSERT INTO
EMPLOYEE_INFO(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) VALUES
(1,'manny','senior employee','2018-06-01','1000000','CSE');

InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid STRING
constant (1000000) for "salary" of type int"

cqlsh:employee_info> INSERT INTO
EMPLOYEE_INFO(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) VALUES
(1,'manny','senior employee','2018-06-01',1000000,'CSE');

cqlsh:employee_info> INSERT INTO
EMPLOYEE_INFO(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) VALUES
(2,'maddy','Manager','2017-04-01',100000,'ISE');

cqlsh:employee_info> INSERT INTO
EMPLOYEE_INFO(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) VALUES
(3,'nathen','junior employee','2019-01-01',200000,'EEE');
cqlsh:employee_info> SELECT * FROM EMPLOYEE_INFO;


 emp_id | dept_name | designation     | doj                             | emp_name | salary
--------+-----------+-----------------+---------------------------------+----------+---------
      1 |       CSE | senior employee | 2018-05-31 18:30:00.000000+0000 |    manny | 1000000
      2 |       ISE |         Manager | 2017-03-31 18:30:00.000000+0000 |    maddy |  100000
      3 |       EEE | junior employee | 2018-12-31 18:30:00.000000+0000 |   nathen |  200000

(3 rows)
cqlsh:employee_info> UPDATE EMPLOYEE_INFO SET Emp_name='mithil',Dept_name='EEE'
WHERE Emp_id=2;

cqlsh:employee_info> SELECT * FROM EMPLOYEE_INFO;


 emp_id | dept_name | designation     | doj                             | emp_name | salary
--------+-----------+-----------------+---------------------------------+----------+---------
      1 |       CSE | senior employee | 2018-05-31 18:30:00.000000+0000 |    manny | 1000000
      2 |       EEE |         Manager | 2017-03-31 18:30:00.000000+0000 |   mithil |  100000
      3 |       EEE | junior employee | 2018-12-31 18:30:00.000000+0000 |   nathen |  200000
```

```
(3 rows)
cqlsh:employee_info> ALTER TABLE EMPLOYEE_INFO ADD PROJECTS SET<text>;

cqlsh:employee_info> SELECT * FROM EMPLOYEE_INFO;


 emp_id | dept_name | designation     | doj                             | emp_name | projects | salary
--------+-----------+-----------------+---------------------------------+----------+----------+---------
      1 |       CSE | senior employee | 2018-05-31 18:30:00.000000+0000 |    manny |     null |
1000000
      2 |       EEE |         Manager | 2017-03-31 18:30:00.000000+0000 |   mithil |     null |  100000
      3 |       EEE | junior employee | 2018-12-31 18:30:00.000000+0000 |   nathen |     null
|  200000

(3 rows)
cqlsh:employee_info> UPDATE EMPLOYEE_INFO SET
PROJECTS=PROJECTS+{'WEBAPP','ANDROIDAPP'} WHERE Emp_id=1;

cqlsh:employee_info> UPDATE EMPLOYEE_INFO SET
PROJECTS=PROJECTS+{'WEBAPP1','ANDROIDAPP1'} WHERE Emp_id=2;

cqlsh:employee_info> UPDATE EMPLOYEE_INFO SET
PROJECTS=PROJECTS+{'WEBAPP2','ANDROIDAPP2'} WHERE Emp_id=3;

cqlsh:employee_info> UPDATE EMPLOYEE_INFO SET
PROJECTS=PROJECTS+{'WEBAPP1','ANDROIDAPP1'} WHERE Emp_id=2;

cqlsh:employee_info> SELECT * FROM EMPLOYEE-INFO;

SyntaxException: line 1:22 no viable alternative at input '-' (SELECT * FROM [EMPLOYEE]-…)
cqlsh:employee_info> SELECT * FROM EMPLOYEE_INFO;


 emp_id | dept_name | designation     | doj                             | emp_name | projects                  |
salary
--------+-----------+-----------------+---------------------------------+----------+---------------------------+---------
      1 |       CSE | senior employee | 2018-05-31 18:30:00.000000+0000 |    manny
|   {'ANDROIDAPP', 'WEBAPP'} | 1000000
      2 |       EEE |         Manager | 2017-03-31 18:30:00.000000+0000 |   mithil |
{'ANDROIDAPP1', 'WEBAPP1'} |  100000
      3 |       EEE | junior employee | 2018-12-31 18:30:00.000000+0000 |   nathen |
{'ANDROIDAPP2', 'WEBAPP2'} |  200000

(3 rows)
cqlsh:employee_info> INSERT INTO
EMPLOYEE_INFO(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) VALUES
(4,'nidhi','junior1 employee','2020-02-04',300000,'ECE') USING TTL 15;

cqlsh:employee_info> SELECT * FROM EMPLOYEE_INFO;
```

```
 emp_id | dept_name | designation     | doj                              | emp_name | projects                 |
salary
--------+-----------+-----------------+----------------------------------+----------+--------------------------+---------
      1 |       CSE | senior employee | 2018-05-31 18:30:00.000000+0000 |    manny
|   {‘ANDROIDAPP’, ‘WEBAPP’} | 1000000
      2 |       EEE |         Manager | 2017-03-31 18:30:00.000000+0000 |   mithil |
{‘ANDROIDAPP1’, ‘WEBAPP1’} |  100000
      3 |       EEE | junior employee | 2018-12-31 18:30:00.000000+0000 |   nathen |
{‘ANDROIDAPP2’, ‘WEBAPP2’} |  200000

(3 rows)
cqlsh:employee_info> INSERT INTO
EMPLOYEE_INFO(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) VALUES
(4,’nidhi’,’junior1 employee’,’2020-02-04’,300000,’ECE’) USING TTL 15;

cqlsh:employee_info> SELECT * FROM EMPLOYEE_INFO;


 emp_id | dept_name | designation      | doj                              | emp_name | projects                 |
salary
--------+-----------+------------------+----------------------------------+----------+--------------------------+---------
      1 |       CSE |  senior employee | 2018-05-31 18:30:00.000000+0000 |    manny
|   {‘ANDROIDAPP’, ‘WEBAPP’} | 1000000
      2 |       EEE |         Manager | 2017-03-31 18:30:00.000000+0000 |   mithil |
{‘ANDROIDAPP1’, ‘WEBAPP1’} |  100000
      4 |       ECE | junior1 employee | 2020-02-03 18:30:00.000000+0000 |    nidhi
|                 null |  300000
      3 |       EEE |  junior employee | 2018-12-31 18:30:00.000000+0000 |   nathen |
{‘ANDROIDAPP2’, ‘WEBAPP2’} |  200000

(4 rows)
cqlsh:employee_info> CREATE TABLE EMP(id int, salary int,name text,PRIMARY
KEY(id,salary));

cqlsh:employee_info> INSERT INTO EMP(id,salary,name) VALUES (1,100000,’myth’);

cqlsh:employee_info> INSERT INTO EMP(id,salary,name) VALUES (1,100000,’myth’);

cqlsh:employee_info> INSERT INTO EMP(id,salary,name) values (1,100000,’myth’);

cqlsh:employee_info> INSERT INTO EMP(id,salary,name) values (2,200000,’mith’);

cqlsh:employee_info> INSERT INTO EMP(id,salary,name) values (3,500000,’nith’);

cqlsh:employee_info> SELECT * FROM EMP WHERE ID IN (1,2,3,4) ORDER BY SALARY;
InvalidRequest: Error from server: code=2200 [Invalid query] message=”Cannot page queries
with both ORDER BY and a IN restriction on the partition key; you must either remove the
ORDER BY or the IN and sort client side, or disable paging for this query”

cqlsh:employee_info> PAGING OFF;
```

Disabled Query paging.

```
cqlsh:employee_info> SELECT * FROM EMP WHERE ID IN (1,2,3,4) ORDER BY SALARY;

 id | salary | name
----+--------+------
  1 | 100000 | myth
  2 | 200000 | mith
  3 | 500000 | nith

(3 rows)
```

NAME:MITHIL RAJ

USN:1BM19CS086

BDA LAB 3 CASSANDRA

LAB 4 CASSANDRA LIBRARY:-

CASSANDRA

Perform the following DB operations using Cassandra.

Program 2:

1 Create a key space 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

OUTPUT:-

```
bmscecse@bmscecse-HP-Pro-3330-MT:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.3 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> create keyspace library_info with replication =
{‘class’:’SimpleStrategy’,’replication_factor’:2};
AlreadyExists: Keyspace ‘library_info’ already exists
cqlsh> describe keyspaces;

library_info  system_auth         system_traces
student       system_distributed  system_views
system        system_schema       system_virtual_schema
```

```
cqlsh:library_info> create table library_details(stud_id int,counter_value counter,stud_name
text,book_id int,book_name text,date_of_issue timestamp,primary
key(stud_id,stud_name,book_name,date_of_issue,book_id));

AlreadyExists: Table 'library_info.library_details' already exists

cqlsh:library_info> create table library_information(stud_id int,counter_value counter,stud_name
text,book_id int,book_name text,date_of_issue timestamp,primary
key(stud_id,stud_name,book_name,date_of_issue,book_id));

cqlsh:library_info> update library_information set counter_value = counter_value+1 where
stud_id = 111 and stud_name ='mithil' and book_name ='BDA' and date_of_issue = '2020-11-
08' and book_id = 200;

cqlsh:library_info> update library_information set counter_value = counter_value+1 where
stud_id = 112 and stud_name ='myth' and book_name ='ML' and date_of_issue = '2020-05-01'
and book_id = 300;

cqlsh:library_info> update library_information set counter_value = counter_value+1 where
stud_id = 113 and stud_name ='mith' and book_name ='OOMD' and date_of_issue = '2020-01-
01' and book_id = 400;

cqlsh:library_info> select * from library-information;

SyntaxException: line 1:25 mismatched character 'o' expecting set null

cqlsh:library_info> select * from library_information;

 stud_id | stud_name | book_name | date_of_issue                   | book_id | counter_value
---------+-----------+-----------+---------------------------------+---------+---------------
     111 |    mithil |       BDA | 2020-11-07 18:30:00.000000+0000 |     200 |             1
     113 |      mith |      OOMD | 2019-12-31 18:30:00.000000+0000 |     400 |             1
     112 |      myth |        ML | 2020-04-30 18:30:00.000000+0000 |     300 |             1


(3 rows)

cqlsh:library_info> update library_information set counter_value = counter_value+1 where
stud_id = 111 and stud_name ='mithil' and book_name ='BDA' and date_of_issue = '2020-11-
08' and book_id = 200;

cqlsh:library_info>  select * from library_information where stud_id = 111;


 stud_id | stud_name | book_name | date_of_issue                   | book_id | counter_value
---------+-----------+-----------+---------------------------------+---------+---------------
     111 |    mithil |       BDA | 2020-11-07 18:30:00.000000+0000 |     200 |             2


cqlsh:library_info> copy
```

```
library_information(stud_id,stud_name,book_id,book_name,date_of_issue,counter_value) to
'/home/bmscecse/library_information.csv';

Using 3 child processes

Starting copy of library_info.library_information with columns [stud_id, stud_name, book_id,
book_name, date_of_issue, counter_value].
Processed: 3 rows; Rate:      32 rows/s; Avg. rate:      32 rows/s
3 rows exported to 1 files in 0.097 seconds.
cqlsh:library_info> truncate library_information;
cqlsh:library_info> copy
library_information(stud_id,stud_name,book_id,book_name,date_of_issue,counter_value) from
'/home/bmscecse/library_information.csv';
Using 3 child processes

Starting copy of library_info.library_information with columns [stud_id, stud_name, book_id,
book_name, date_of_issue, counter_value].
Processed: 3 rows; Rate:       5 rows/s; Avg. rate:       7 rows/s
3 rows imported from 1 files in 0.418 seconds (0 skipped).
cqlsh:library_info> select * from library_information;


 stud_id | stud_name | book_name | date_of_issue                   | book_id | counter_value
---------+-----------+-----------+---------------------------------+---------+---------------
     111 |    mithil |       BDA | 2020-11-07 18:30:00.000000+0000 |     200 |             2
     113 |      mith |      OOMD | 2019-12-31 18:30:00.000000+0000 |     400 |             1
     112 |      myth |        ML | 2020-04-30 18:30:00.000000+0000 |     300 |             1


(3 rows)
```

LAB – 5 HADOOP COMMANDS SCREENSHOTS:-

OUTPUT:-

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /
21/04/19 23:41:08 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:19 /mydir
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:21 /mydr
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:39 /newdir
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -mv /mydr /newdir
21/04/19 23:41:38 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /
21/04/19 23:41:44 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:19 /mydir
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:41 /newdir
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /newdir
21/04/19 23:42:05 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:21 /newdir/mydr
hduser@lab-VirtualBox:/usr/local/sbin$
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /
21/04/19 23:52:26 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:45 /mydir
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:48 /newdir
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -rm -R /mydir
21/04/19 23:52:56 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
21/04/19 23:52:57 INFO fs.TrashPolicyDefault: Namenode trash configuration: Del
etion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /mydir
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /
21/04/19 23:53:02 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:48 /newdir
hduser@lab-VirtualBox:/usr/local/sbin$
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -mkdir /mydir
21/04/19 22:58:30 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /
21/04/19 22:58:36 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 22:58 /mydir
drwxr-xr-x   - hduser supergroup          0 2021-04-18 19:27 /mydr
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -get /mydr ~/copyfromhadoop
21/04/19 23:25:49 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /
21/04/19 23:48:41 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:45 /mydir
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:41 /newdir
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -cp /mydir/sample.txt /newdir
21/04/19 23:48:56 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /newdir
21/04/19 23:49:22 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x   - hduser supergroup          0 2021-04-19 23:21 /newdir/mydr
-rw-r--r--   1 hduser supergroup         13 2021-04-19 23:48 /newdir/sample.txt
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -copyToLocal /mydir ~/hadoopco
py
21/04/19 23:29:39 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
hduser@lab-VirtualBox:/usr/local/sbin$
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -copyFromLocal ~/file1.txt /my
dir
21/04/19 23:19:36 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -ls /mydir
21/04/19 23:20:13 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--   1 hduser supergroup         30 2021-04-19 23:19 /mydir/file1.txt
```

```
hduser@lab-VirtualBox:/usr/local/sbin$ hadoop fs -cat /mydir/file1.txt
21/04/19 23:38:07 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
I am using Hadoop
line1
line2
```

HADOOP INSTALLATION:-

```
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\WINDOWS\system32>jps
7072 DataNode
13492 Jps
15844 ResourceManager
16196 NameNode
1388 NodeManager

C:\WINDOWS\system32>hdfs dfs -ls -R /
drwxr-xr-x   - khush supergroup          0 2022-06-27 14:09 /input
drwxr-xr-x   - khush supergroup          0 2022-06-21 09:03 /input/inputtest
-rw-r--r--   1 khush supergroup         21 2022-06-21 09:03 /input/inputtest/output.txt
-rw-r--r--   1 khush supergroup         21 2022-06-21 08:19 /input/sample.txt
-rw-r--r--   1 khush supergroup         21 2022-06-27 14:09 /input/sample2.txt
drwxr-xr-x   - khush supergroup          0 2022-06-21 13:30 /test
-rw-r--r--   1 khush supergroup         19 2022-06-21 13:30 /test/sample.txt

C:\WINDOWS\system32>hadoop version
Hadoop 3.3.3
Source code repository https://github.com/apache/hadoop.git -r d37586cbda38c338d9fe481addda5a05fb516f71
Compiled by stevel on 2022-05-09T16:36Z
Compiled with protoc 3.7.1
From source with checksum eb96dd4a797b6989ae0cdb9db6efc6
This command was run using /C:/hadoop-3.3.3/share/hadoop/common/hadoop-common-3.3.3.jar

C:\WINDOWS\system32>
```

HADOOP PROGRAMS :-

- HADOOP PROGRAM WORD COUNT TOP N
- HADOOP PROGRAM TEMPERATURE
- HADOOP PROGRAM USE OF JOIN

1) WORD COUNT MAPREDUCE LAB 06:-

TOP N:

```
// TopN.java package sortWords;

import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.Path; import
org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import
org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; import
org.apache.hadoop.mapreduce.Reducer; import
org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import
org.apache.hadoop.util.GenericOptionsParser; import utils.MiscUtils;
import java.io.IOException; import java.util.*;

public class TopN {
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs(); if
(otherArgs.length != 2) {
System.err.println("Usage: TopN <in> <out>");
System.exit(2);
}
Job job = Job.getInstance(conf); job.setJobName("Top N"); job.setJarByClass(TopN.class);
job.setMapperClass(TopNMapper.class); //job.setCombinerClass(TopNReducer.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);
}
```

MapperN:-

```
The mapper reads one line at the time, splits it into an array of single words and emits every *
word to the reducers with the value of 1.
*/

public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
private final static IntWritable one = new IntWritable(1); private Text word = new Text();
private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";
@Override
```

```
public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
String cleanLine = value.toString().toLowerCase().replaceAll(tokens, " "); StringTokenizer itr =
new StringTokenizer(cleanLine); while (itr.hasMoreTokens()) {
word.set(itr.nextToken().trim()); context.write(word, one);
}
}
}
```

ReducerN:-

```
The reducer retrieves every word and puts it into a Map: if the word already exists in the * map,
increments its value, otherwise sets it to 1.
*/
public static class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
private Map<Text, IntWritable> countMap = new HashMap<>();
@Override
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {
// computes the number of occurrences of a single word int sum = 0; for (IntWritable val :
values) { sum += val.get();
}
// puts the number of occurrences of this word into the map.
// We need to create another Text object because the Text instance
// we receive is the same for all the words countMap.put(new Text(key), new
IntWritable(sum));
}
@Override
protected void cleanup(Context context) throws IOException, InterruptedException {
Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(countMap);
int counter = 0; for (Text key : sortedMap.keySet()) { if (counter++ == 3) { break;
}
context.write(key, sortedMap.get(key));
}
}
}
```

CombinerN:-

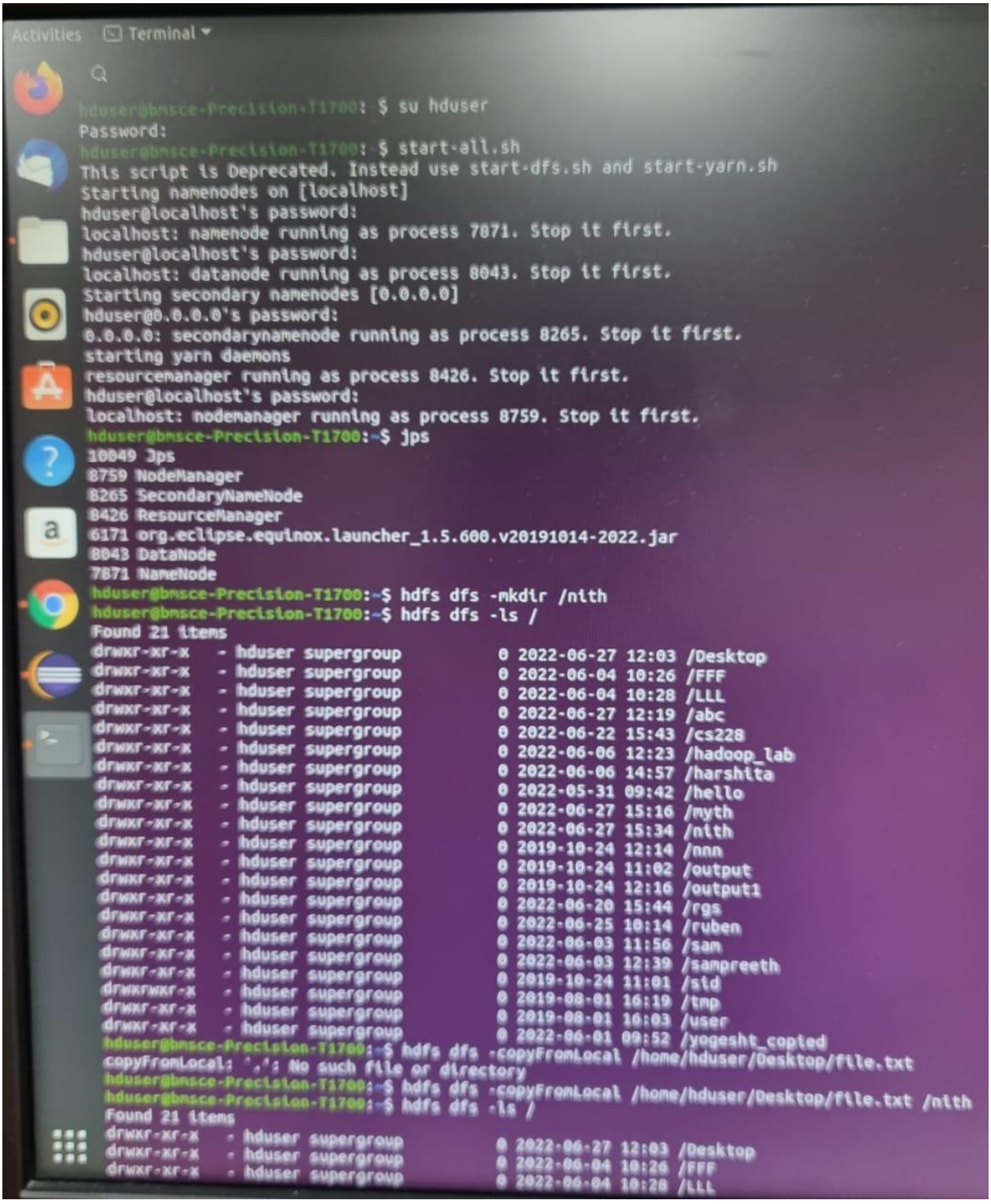
```
The combiner retrieves every word and puts it into a Map: if the word already exists in the *
map, increments its value, otherwise sets it to 1.
*/
public static class TopNCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {
```

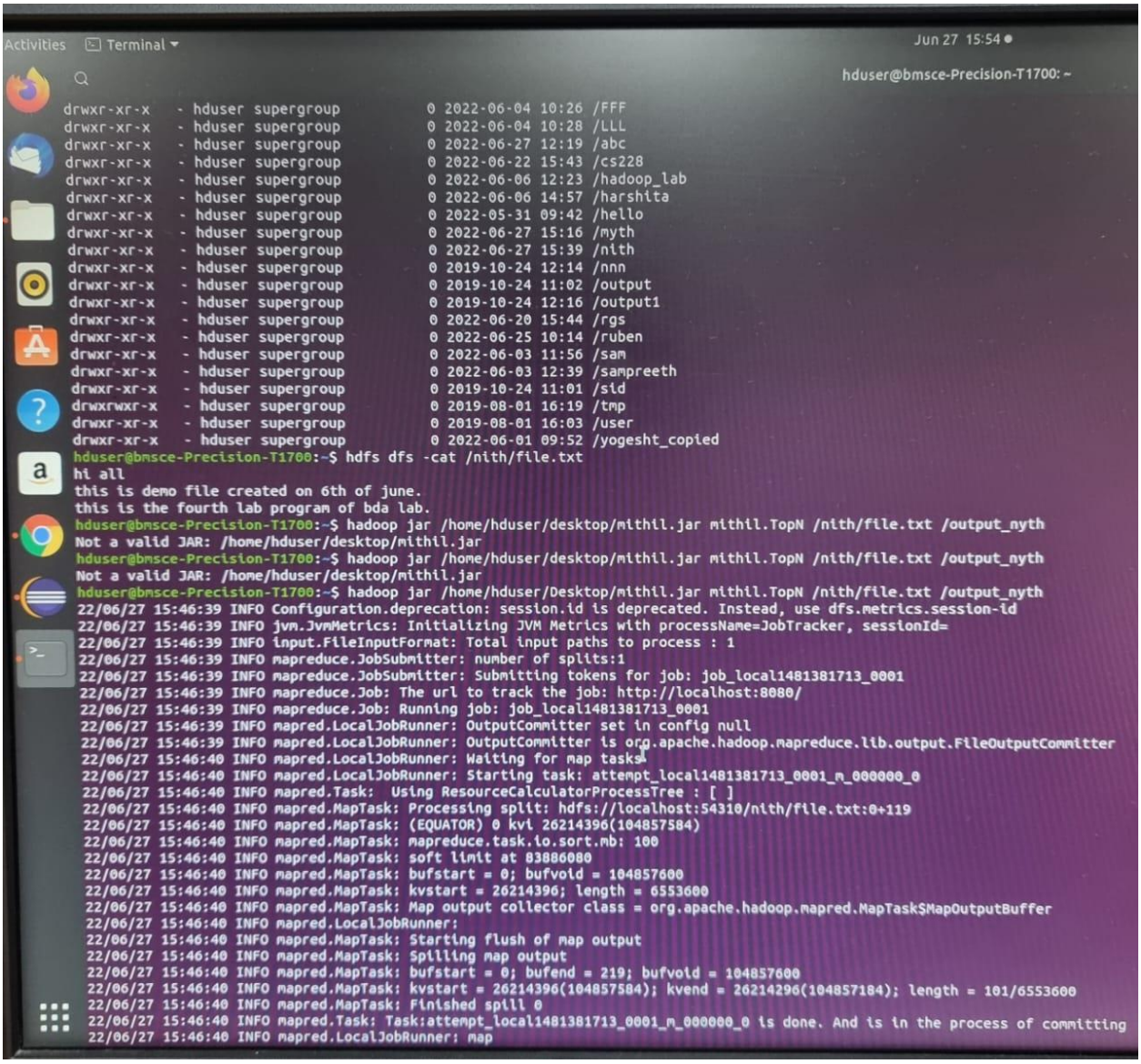
```
@Override
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {
// computes the number of occurrences of a single word int sum = 0; for (IntWritable val :
values) { sum += val.get();
}
context.write(key, new IntWritable(sum));
}
}
}
```

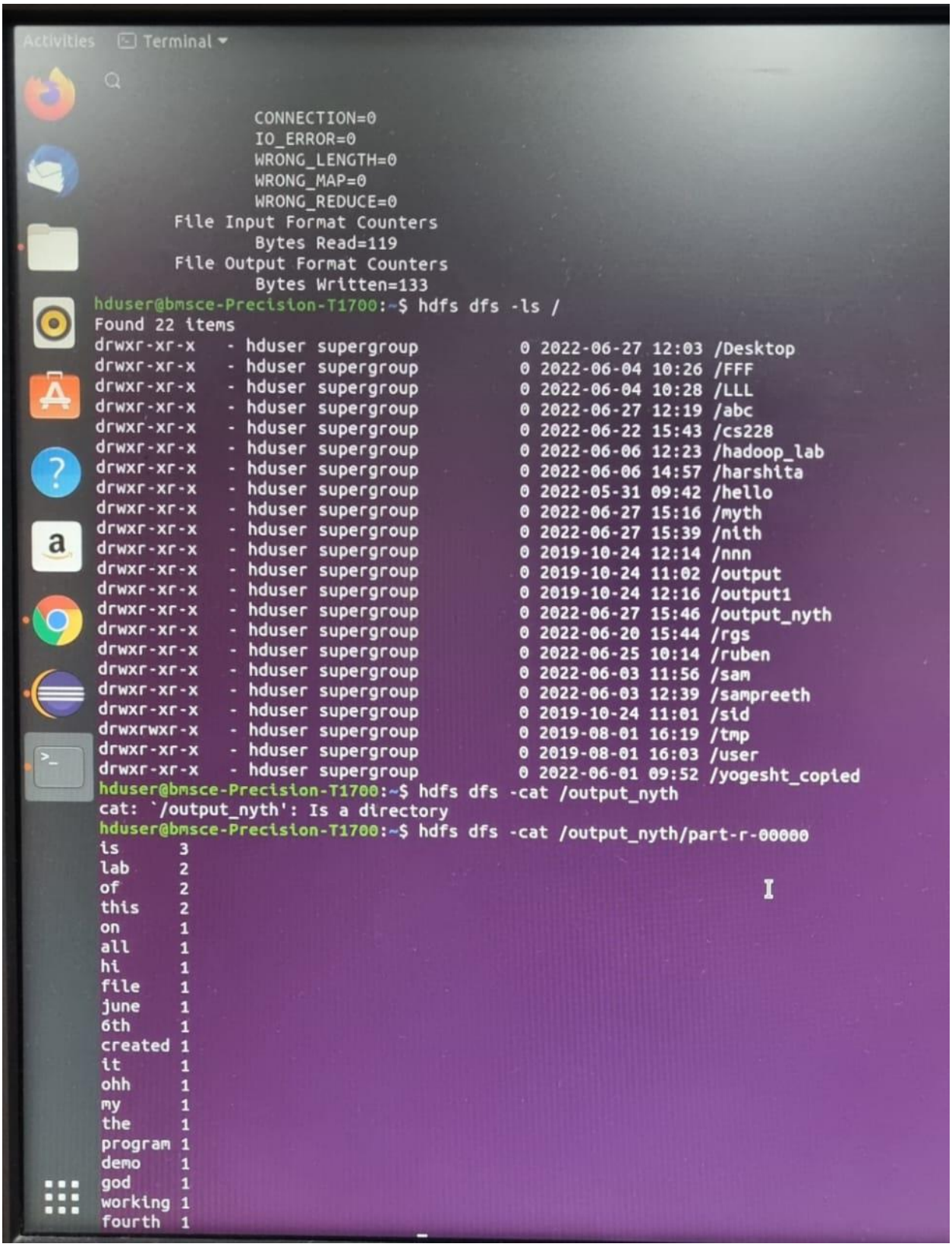
UlitsN:-

```
// MiscUtils.java package utils;
import java.util.*;
public class MiscUtils {
/**
sorts the map by values. Taken from:
http://javarevisited.blogspot.it/2012/12/how-to-sort-hashmap-java-by-key-and-value.html
*/
public static <K extends Comparable, V extends Comparable> Map<K, V> sortByValues(Map<K,
V> map) {
List<Map.Entry<K, V>> entries = new LinkedList<Map.Entry<K, V>>(map.entrySet());
Collections.sort(entries, new Comparator<Map.Entry<K, V>>() {
@Override public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) { return
o2.getValue().compareTo(o1.getValue());
}
});
//LinkedHashMap will keep the keys in the order they are inserted
//which is currently sorted on natural ordering
Map<K, V> sortedMap = new LinkedHashMap<K, V>();
for (Map.Entry<K, V> entry : entries) {
sortedMap.put(entry.getKey(), entry.getValue());
}
return sortedMap;
}
}
```

OUTPUT:-


Activities
Terminal
hduser@bmsce-Precision-T1700: $ su hduser
Password:
hduser@bmsce-Precision-T1700: $ 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: namenode running as process 7871. Stop it first.
hduser@localhost's password:
localhost: datanode running as process 8043. Stop it first.
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: secondarynamenode running as process 8265. Stop it first.
starting yarn daemons
resourcemanager running as process 8426. Stop it first.
hduser@localhost's password:
localhost: nodemanager running as process 8759. Stop it first.
hduser@bmsce-Precision-T1700:~$ jps
10049 Jps
8759 NodeManager
8265 SecondaryNameNode
8426 ResourceManager
6171 org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar
8043 DataNode
7871 NameNode
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /nith
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 21 items
drwxr-xr-x - hduser supergroup 0 2022-06-27 12:03 /Desktop
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:26 /FFF
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:28 /LLL
drwxr-xr-x - hduser supergroup 0 2022-06-27 12:19 /abc
drwxr-xr-x - hduser supergroup 0 2022-06-22 15:43 /cs228
drwxr-xr-x - hduser supergroup 0 2022-06-06 12:23 /hadoop_lab
drwxr-xr-x - hduser supergroup 0 2022-06-06 14:57 /harshita
drwxr-xr-x - hduser supergroup 0 2022-05-31 09:42 /hello
drwxr-xr-x - hduser supergroup 0 2022-06-27 15:16 /myth
drwxr-xr-x - hduser supergroup 0 2022-06-27 15:34 /nith
drwxr-xr-x - hduser supergroup 0 2019-10-24 12:14 /nnn
drwxr-xr-x - hduser supergroup 0 2019-10-24 11:02 /output
drwxr-xr-x - hduser supergroup 0 2019-10-24 12:16 /output1
drwxr-xr-x - hduser supergroup 0 2022-06-20 15:44 /rgs
drwxr-xr-x - hduser supergroup 0 2022-06-25 10:14 /ruben
drwxr-xr-x - hduser supergroup 0 2022-06-03 11:56 /sam
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:39 /sampreeth
drwxr-xr-x - hduser supergroup 0 2019-10-24 11:01 /sid
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 09:52 /yogesht_copied
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/Desktop/file.txt
copyFromLocal: `.': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/Desktop/file.txt /nith
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 21 items
drwxr-xr-x - hduser supergroup 0 2022-06-27 12:03 /Desktop
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:26 /FFF
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:28 /LLL


Activities
Terminal
Jun 27 15:54
hduser@bmsce-Precision-T1700: ~
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:26 /FFF
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:28 /LLL
drwxr-xr-x - hduser supergroup 0 2022-06-27 12:19 /abc
drwxr-xr-x - hduser supergroup 0 2022-06-22 15:43 /cs228
drwxr-xr-x - hduser supergroup 0 2022-06-06 12:23 /hadoop_lab
drwxr-xr-x - hduser supergroup 0 2022-06-06 14:57 /harshita
drwxr-xr-x - hduser supergroup 0 2022-05-31 09:42 /hello
drwxr-xr-x - hduser supergroup 0 2022-06-27 15:16 /myth
drwxr-xr-x - hduser supergroup 0 2022-06-27 15:39 /nith
drwxr-xr-x - hduser supergroup 0 2019-10-24 12:14 /nnn
drwxr-xr-x - hduser supergroup 0 2019-10-24 11:02 /output
drwxr-xr-x - hduser supergroup 0 2019-10-24 12:16 /output1
drwxr-xr-x - hduser supergroup 0 2022-06-20 15:44 /rgs
drwxr-xr-x - hduser supergroup 0 2022-06-25 10:14 /ruben
drwxr-xr-x - hduser supergroup 0 2022-06-03 11:56 /sam
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:39 /sampreeth
drwxr-xr-x - hduser supergroup 0 2019-10-24 11:01 /sid
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 09:52 /yogesht_copied
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /nith/file.txt
hi all
this is demo file created on 6th of june.
this is the fourth lab program of bda lab.
hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/desktop/mithil.jar mithil.TopN /nith/file.txt /output_nyth
Not a valid JAR: /home/hduser/desktop/mithil.jar
hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/desktop/mithil.jar mithil.TopN /nith/file.txt /output_nyth
Not a valid JAR: /home/hduser/desktop/mithil.jar
hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/Desktop/mithil.jar mithil.TopN /nith/file.txt /output_nyth
22/06/27 15:46:39 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
22/06/27 15:46:39 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
22/06/27 15:46:39 INFO input.FileInputFormat: Total input paths to process : 1
22/06/27 15:46:39 INFO mapreduce.JobSubmitter: number of splits:1
22/06/27 15:46:39 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1481381713_0001
22/06/27 15:46:39 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/27 15:46:39 INFO mapreduce.Job: Running job: job_local1481381713_0001
22/06/27 15:46:39 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/27 15:46:39 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/27 15:46:40 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/27 15:46:40 INFO mapred.LocalJobRunner: Starting task: attempt_local1481381713_0001_m_000000_0
22/06/27 15:46:40 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
22/06/27 15:46:40 INFO mapred.MapTask: Processing split: hdfs://localhost:54310/nith/file.txt:0+119
22/06/27 15:46:40 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/27 15:46:40 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/27 15:46:40 INFO mapred.MapTask: soft limit at 83886080
22/06/27 15:46:40 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/27 15:46:40 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/27 15:46:40 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/27 15:46:40 INFO mapred.LocalJobRunner:
22/06/27 15:46:40 INFO mapred.MapTask: Starting flush of map output
22/06/27 15:46:40 INFO mapred.MapTask: Spilling map output
22/06/27 15:46:40 INFO mapred.MapTask: bufstart = 0; bufend = 219; bufvoid = 104857600
22/06/27 15:46:40 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214296(104857184); length = 101/6553600
22/06/27 15:46:40 INFO mapred.MapTask: Finished spill 0
22/06/27 15:46:40 INFO mapred.Task: Task:attempt_local1481381713_0001_m_000000_0 is done. And is in the process of committing
22/06/27 15:46:40 INFO mapred.LocalJobRunner: map


Activities
Terminal
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=119
File Output Format Counters
Bytes Written=133
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 22 items
drwxr-xr-x - hduser supergroup 0 2022-06-27 12:03 /Desktop
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:26 /FFF
drwxr-xr-x - hduser supergroup 0 2022-06-04 10:28 /LLL
drwxr-xr-x - hduser supergroup 0 2022-06-27 12:19 /abc
drwxr-xr-x - hduser supergroup 0 2022-06-22 15:43 /cs228
drwxr-xr-x - hduser supergroup 0 2022-06-06 12:23 /hadoop_lab
drwxr-xr-x - hduser supergroup 0 2022-06-06 14:57 /harshita
drwxr-xr-x - hduser supergroup 0 2022-05-31 09:42 /hello
drwxr-xr-x - hduser supergroup 0 2022-06-27 15:16 /myth
drwxr-xr-x - hduser supergroup 0 2022-06-27 15:39 /nith
drwxr-xr-x - hduser supergroup 0 2019-10-24 12:14 /nnn
drwxr-xr-x - hduser supergroup 0 2019-10-24 11:02 /output
drwxr-xr-x - hduser supergroup 0 2019-10-24 12:16 /output1
drwxr-xr-x - hduser supergroup 0 2022-06-27 15:46 /output_nyth
drwxr-xr-x - hduser supergroup 0 2022-06-20 15:44 /rgs
drwxr-xr-x - hduser supergroup 0 2022-06-25 10:14 /ruben
drwxr-xr-x - hduser supergroup 0 2022-06-03 11:56 /sam
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:39 /sampreeth
drwxr-xr-x - hduser supergroup 0 2019-10-24 11:01 /sid
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 09:52 /yogesht_copied
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /output_nyth
cat: `/output_nyth': Is a directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /output_nyth/part-r-00000
is 3
lab 2
of 2
this 2
on 1
all 1
hi 1
file 1
june 1
6th 1
created 1
it 1
ohh 1
my 1
the 1
program 1
demo 1
god 1
working 1
fourth 1

2)HADOOP MAPREDUCE TEMPERATURE PROGRAM LAB 07:-

Find the average temperature for each year from the NCDC data set.

```
// AverageDriver.java package temperature;

import org.apache.hadoop.io.*; import org.apache.hadoop.fs.*; import
org.apache.hadoop.mapreduce.*; import
org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class AverageDriver
{ public static void main (String[] args) throws Exception
{
if (args.length != 2)
{
System.err.println("Please Enter the input and output parameters");
System.exit(-1);
}
Job job = new Job(); job.setJarByClass(AverageDriver.class); job.setJobName("Max
temperature");
FileInputFormat.addInputPath(job,new Path(args[0]));
FileOutputFormat.setOutputPath(job,new Path (args[1]));
job.setMapperClass(AverageMapper.class); job.setReducerClass(AverageReducer.class);
job.setOutputKeyClass(Text.class); job.setOutputValueClass(IntWritable.class);
System.exit(job.waitForCompletion(true)?0:1);
}
}
//AverageMapper.java package temperature;
import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.*; import
java.io.IOException;
public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{ public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
{
String line = value.toString(); String year = line.substring(15,19); int temperature; if
(line.charAt(87)=='+') temperature = Integer.parseInt(line.substring(88, 92));
else
temperature = Integer.parseInt(line.substring(87, 92)); String quality = line.substring(92, 93);
if(temperature != MISSING && quality.matches("[01459]")) context.write(new Text(year),new
IntWritable(temperature)); }
```

```
}
//AverageReducer.java package temperature;

import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import
org.apache.hadoop.mapreduce.*; import java.io.IOException;
public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable>
{
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException,InterruptedException
{
int max_temp = 0; int count = 0;
for (IntWritable value : values)
{
max_temp += value.get();
count+=1;
}
context.write(key, new IntWritable(max_temp/count));
}
}
```

OUTPUT:-

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

```
//TempDriver.java package temperatureMax;

import org.apache.hadoop.io.*; import org.apache.hadoop.fs.*; import
org.apache.hadoop.mapreduce.*; import
org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class TempDriver
{ 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(TempDriver.class); job.setJobName("Max temperature");
```

```
FileInputFormat.addInputPath(job,new Path(args[0]));
FileOutputFormat.setOutputPath(job,new Path (args[1]));

job.setMapperClass(TempMapper.class); job.setReducerClass(TempReducer.class);
job.setOutputKeyClass(Text.class); job.setOutputValueClass(IntWritable.class);

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

TempMapper:-

```
//TempMapper.java package temperatureMax;

import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.*; import
java.io.IOException;
public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>

{
 public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException

{

String line = value.toString(); String month = line.substring(19,21); int temperature; if
(line.charAt(87)=='+') temperature = Integer.parseInt(line.substring(88, 92));

else

temperature = Integer.parseInt(line.substring(87, 92)); String quality = line.substring(92, 93);
if(temperature != MISSING && quality.matches("[01459]")) context.write(new
Text(month),new IntWritable(temperature)); }
}
```

TempReducer:-

```
//TempReducer.java package temperatureMax;

import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.*; import
java.io.IOException;
public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{ public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
{
```

String line = value.toString(); String month = line.substring(19,21); int temperature; if (line.charAt(87)==’+’) temperature = Integer.parseInt(line.substring(88, 92));
else

temperature = Integer.parseInt(line.substring(87, 92)); String quality = line.substring(92, 93); if(temperature != MISSING &&

quality.matches(“[01459]”)) context.write(new Text(month),new IntWritable(temperature));
}
}

OUPUT:-

```
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
```

HADOOP MAPREDUCE PROGRAM USE OF JOIN LAB 08:-

Create a Hadoop Map Reduce program to combine information from the users file along with Information from the posts file by using the concept of join and display user_id, Reputation and Score.

// 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:-

```
// 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;
```

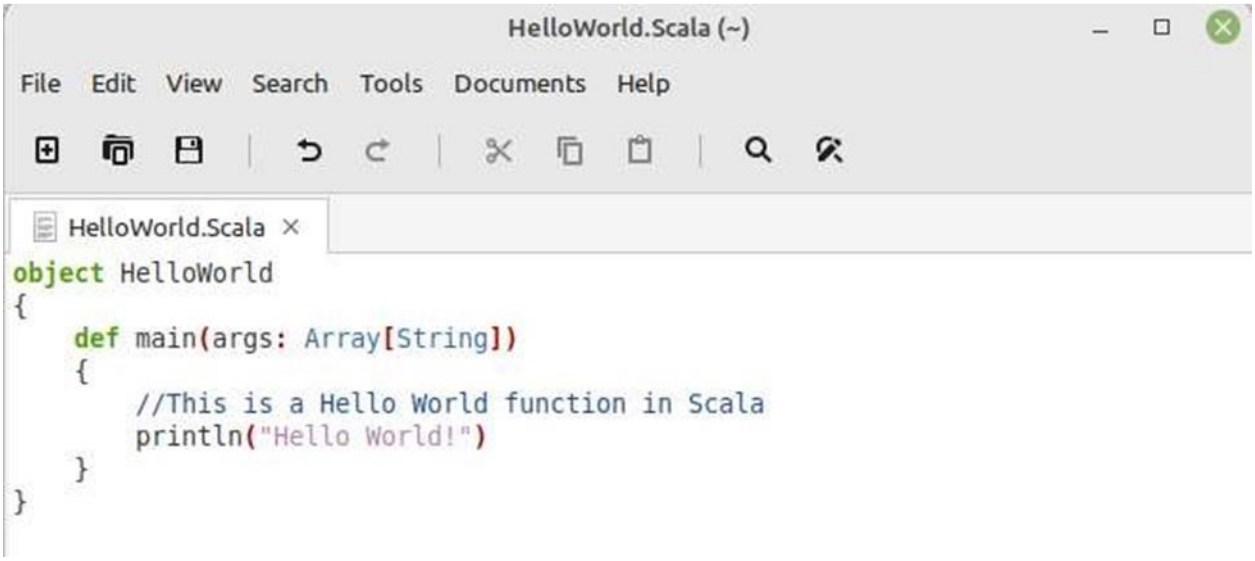
```
public TextPair() { set(new Text(), new Text());
}
public TextPair(String first, String second) { set(new Text(first), new Text(second));
}
public TextPair(Text first, Text second) { set(first, second);
}

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€;
}
}
}

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€;
}
}

@Override
public int compare(WritableComparable a, WritableComparable b) { if (a instanceof TextPair
&& b instanceof TextPair) { return ((TextPair) a).first.compareTo(((TextPair) b).first);
}
return super.compare(a, b);
}
}
}
```

OUTPUT:-

```
c:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \joinOutput\part-00000
"100005361"     "2"             "36134"
"100018705"     "2"             "76"
"100022094"     "0"             "6354"
```

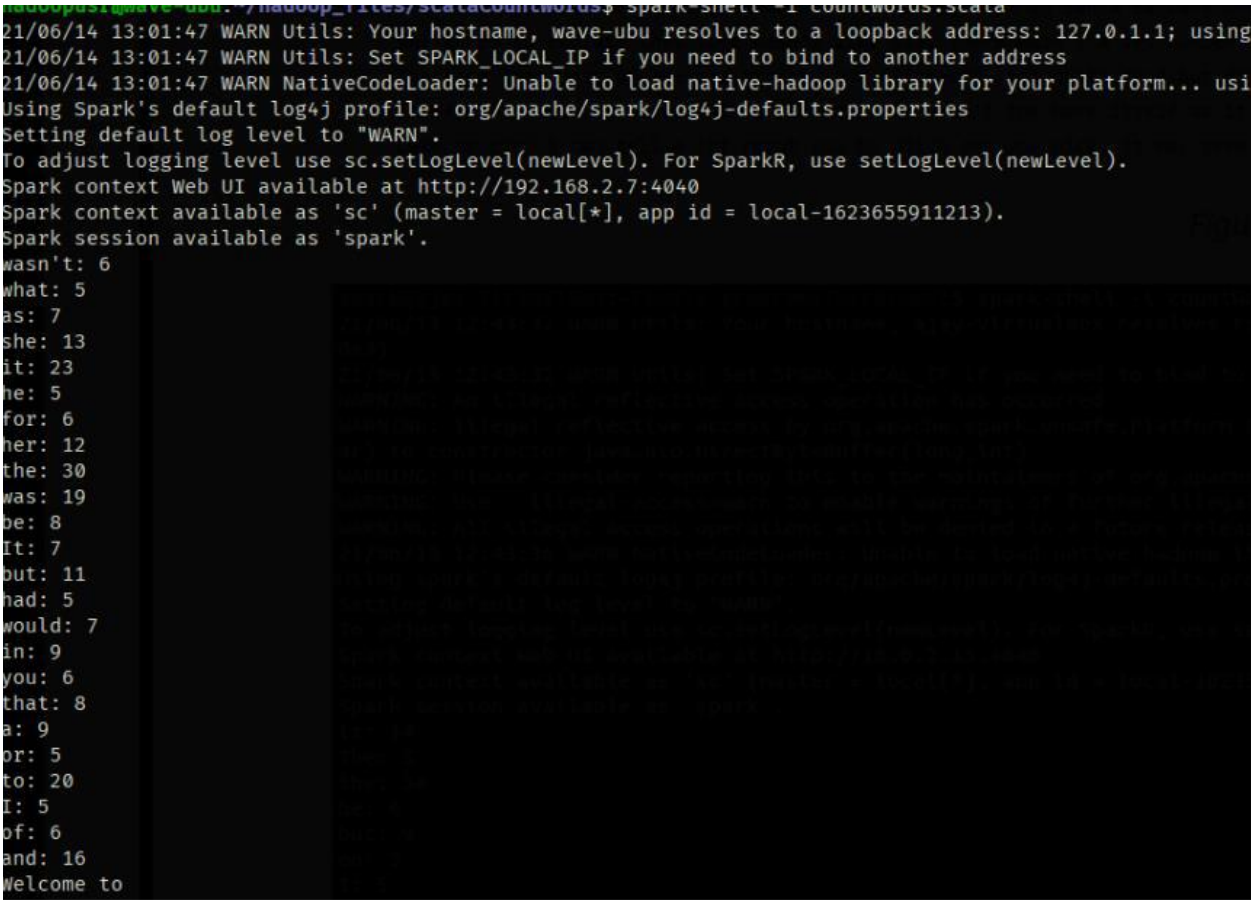
LAB 09:- SCALA HELLO WORLD PROGRAM FROM ONLINE SCALA IDE :-



SCALA WORD COUNT PROGRAM:-

```
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;
```

OUTPUT:-



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 4 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 <conso
le>: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)
```