# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**LAB REPORT on**

# BIG DATA ANALYTICS
# (20CS6PEBDA)

***Submitted by***

**ARKA SINHA (1BM19CS024)**

***in partial fulfilment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**B.M.S. COLLEGE OF ENGINEERING BENGALURU-560019 May-2022 to July-2022**

**(Autonomous Institution under VTU)**

**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 **Arka Sinha (1BM19CS024),** 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 Big data analytics - (20CS6PEBDA) work prescribed for the said degree.

Name of the Lab-In charge
ANTARA ROY CHOUDHURY
Designation- Assistant Professor
Department of CSE
BMSCE, Bengaluru

# Index Sheet

| Sl. No. | Experiment Title | Page No. |
|---|---|---|
| 1 | MongoDB- CRUD Demonstration | 5 |
| 2 | Cassandra Lab Program 1: - Student Database | 16 |
| 3 | Cassandra Lab Program 2: - Library Database | 20 |
| 4 | Hadoop Installation | 22 |
| 5 | Hadoop Commands | 23 |
| 6 | Hadoop Programs: Word Count | 26 |
| 7 | Hadoop Programs: Top N | 32 |
| 8 | Hadoop Programs: Average Temperature | 37 |
| 9 | Hadoop Programs: Join | 44 |
| 10 | Scala Programs: Word Count | 53 |
| 11 | Scala Programs: Word Count greater than 4 | 54 |

## 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:

## I.CREATE DATABASE IN MONGODB.

**> use arkaDB switched to**

**db arkaDB db;**

arkaDB **show dbs; admin**

**0.000GB config**

**0.000GB local 0.000GB**

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

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

**db.createCollection(“Student”); => *sql equivalent***
**CREATE TABLE STUDENT(…);**

**{ "ok" : 1 }** 2.To drop a collection by the

name “Student”.

**db.Student.drop();** 3.Create a collection by the name “Students” and store the following data in it.
**db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing"});**

WriteResult({ "nInserted" : 1 })

4. Insert the document for “AryanDavid” in to the Students collection only if it does not already exist in the collection. However, if it is already present in the collection, then update the document with new values. (Update his Hobbies from “Skating” to “Chess”. ) Use “Update else insert” (if there is an existing

document, it will attempt to update it, if there is no existing document then it will insert it).

**db.Student.update({_id:3,StudName:"AryanDavid",Grade:"VII"},{$set:{Hobbies:"Skating"}},{upsert:true});**

**WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 3 })**

5. FIND METHOD

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

**db.Student.find({StudName:"AryanDavid"});**
**({cond..},{columns.. column:1, columnname:0} )**

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

**B.** To display only the StudName and Grade from all the documents of the Students collection. The identifier_id should be suppressed and NOT displayed.
**db.Student.find({},{StudName:1,Grade:1,_id:0});**

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

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

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

**{**
**"_id" : 1,**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## III. Import data from a CSV file

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

**mongoimport --db Student --collection airlines --type csv –headerline --file /home/hduser/Desktop/airline.csv**

## IV. Export data to a CSV file

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.Student.save({StudName:”Vamsi”, Grade:”VI”})
WriteResult({ "nInserted" : 1 })
```

## VI. Add a new field to existing Document:

```
db.Student.update({_id:ObjectId("625695cc7d129fb98b44c8a1")},
{$set:{Location:"Network"}})

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

**VII. Remove the field in an existing Document**

```
db.Student.update({_id:ObjectId("625695cc7d129fb98b44c8a1
")},
{$unset:{Location:”Network”}})
```

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

**VIII. Finding Document based on search criteria suppressing few fields**

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

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

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

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

```
{
    "_id" : ObjectId("625695cc7d129fb98b44c8a1"),
    "StudName" : "Vamsi",
    "Grade" : "VI"
}
```

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

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

```
{
    "_id" : 1,
```

```
    "StudName" : "MichelleJacintha",
    "Grade" : "VII",
    "Hobbies" : "InternetSurfing"
}
```

**IX. to set a particular field value to NULL**

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

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

**X. Count the number of documents in Student Collections**

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

```
3
```

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

```
db.Student.count({Grade:"VII"
}) 2 retrieve first 3
documents

db.Student.find({Grade:"VII"}).limit(1).pretty();

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

}

**Sort the document in Ascending order**

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


{
"_id" : 3,
"Grade" : "VII",
"StudName" : "AryanDavid",
"Hobbies" : "Skating",
"Location" : null
}
{
"_id" : 1,
"StudName" : "MichelleJacintha",
"Grade" : "VII",
"Hobbies" : "InternetSurfing"
}
{
"_id" : ObjectId("625695cc7d129fb98b44c8a1"),
"StudName" : "Vamsi",
"Grade" : "VI"
}


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

**to Skip the 1st two documents from the Students Collections**

db.Student.find().skip(2).pretty()


{

```
    "_id" : ObjectId("625695cc7d129fb98b44c8a1"),
    "StudName" : "Vamsi",
    "Grade" : "VI"
}
```

**XII.** Create a collection by name “food” and add to each document add a “fruits” array

```
db.food.insert( { _id:1, fruits:['grapes','mango','apple'] } )
db.food.insert( { _id:2, fruits:['grapes','mango','cherry'] } )
db.food.insert( { _id:3, fruits:['banana','mango'] } )
```

```
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
{ "_id" : 3, "fruits" : [ "banana", "mango" ] }
```

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

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

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

**To find in “fruits” array having “mango” in the first index position.**

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

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

```
db.food.find ( {“fruits”: {$size:2}} )
```

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

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

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

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

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

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

{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }

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

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

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 }) insert new key value pairs in the fruits array

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

{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }

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

# LAB 2:

Perform the following DB operations using Cassandra.

1. Create a key space by name Employee

```
create keyspace "Employee" with replication =
{'class':'SimpleStrategy','replication_factor':1};
cqlsh> use Employee;
```

2. Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining,
Salary, Dept_Name

```
create table Employee_Info(Emp_id int PRIMARY
KEY,Emp_name text,Date_of_Joining timestamp,Salary
float,Dept_Name text) ;
```

3. Insert the values into the table in batch

```
cqlsh:employee> begin batch
        ... insert into
Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,
Dept_N ame) values(1,'Khushil','2021-04-23',50000,'CSE')
        ... insert into
Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,
Dept_N ame) values(2,'Tarun','2020-06-21',10000,'ISE')
        ... insert into
Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,
Dept_N ame) values(3,'Suresh','2011-02-12',30000,'ECE')
        ... insert into
Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_N
ame) values(4,'Yuresh','2015-09-02',90000,'EEE')
        ... insert into
Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,
Dept_N ame)
values(5,'Dharmesh','2016-01-09',70000,'CSE')
        ... apply batch;
```

```
cqlsh> create keyspace Employee with replication = {'class':'SimpleStrategy'
,'replication_factor':1};
cqlsh> use Employee
   ... ;
cqlsh:employee> create table Employee_Info(Emp_id int PRIMARY KEY,Emp_name t
ext,Date_of_Joining timestamp,Salary float,Dept_Name text);
cqlsh:employee> begin batch
            ... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Sa
lary,Dept_Name) values(1,'Nithin','2021-04-23',50000,'CSE')
            ...  insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,S
alary,Dept_Name) values(2,'Tarun','2020-06-21',10000,'ISE')
            ...  insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,S
alary,Dept_Name) values(3,'Suresh','2011-02-12',30000,'ECE')
            ...  insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,S
alary,Dept_Name) values(4,'Yuresh','2015-09-02',90000,'EEE')
            ...  insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,S
alary,Dept_Name) values(5,'Dharmesh','2016-01-09',70000,'CSE')
            ... apply batch;
cqlsh:employee> select * from Employee_info;

 emp_id | date_of_joining                 | dept_name | emp_name | salary
--------+---------------------------------+-----------+----------+--------
      5 | 2016-01-09 00:00:00.000000+0000 |       CSE | Dharmesh |  70000
      1 | 2021-04-23 00:00:00.000000+0000 |       CSE |   Nithin |  50000
      2 | 2020-06-21 00:00:00.000000+0000 |       ISE |    Tarun |  10000
      4 | 2015-09-02 00:00:00.000000+0000 |       EEE |   Yuresh |  90000
      3 | 2011-02-12 00:00:00.000000+0000 |       ECE |   Suresh |  30000
```

4. Update Employee name and Department of Emp-Id 1 update employee_info set Dept_Name='Mech',emp_name='Sreekar' where emp_id=1;
5. cqlsh:employee> select * from employee_info;

```
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | emp_name | salary
--------+---------------------------------+-----------+----------+--------
      5 | 2016-01-09 00:00:00.000000+0000 |       CSE | Dharmesh |  70000
      1 | 2021-04-23 00:00:00.000000+0000 |      Mech |  Sreekar |  50000
      2 | 2020-06-21 00:00:00.000000+0000 |       ISE |    Tarun |  10000
      4 | 2015-09-02 00:00:00.000000+0000 |       EEE |   Yuresh |  90000
      3 | 2011-02-12 00:00:00.000000+0000 |       ECE |   Suresh |  30000

(5 rows)
```

6. Sort the details of Employee records based on salary

```
(0 rows)
cqlsh:employee> begin batch
           ... insert into Employee_information(Emp_id,Emp_name,Date_of_Joi
ning,Salary,Dept_Name) values(1,'Nithin','2021-04-23',50000,'CSE')
           ... insert into Employee_information(Emp_id,Emp_name,Date_of_Joi
ning,Salary,Dept_Name) values(2,'Tarun','2020-06-21',10000,'ISE')
           ... insert into Employee_information(Emp_id,Emp_name,Date_of_Joi
ning,Salary,Dept_Name) values(3,'Suresh','2011-02-12',30000,'ECE')
           ... apply batch;
cqlsh:employee> select * from Employee_information;

 emp_id | salary | date_of_joining                 | dept_name | emp_name
--------+--------+---------------------------------+-----------+----------
      1 |  50000 | 2021-04-23 00:00:00.000000+0000 |       CSE |   Nithin
      2 |  10000 | 2020-06-21 00:00:00.000000+0000 |       ISE |    Tarun
      3 |  30000 | 2011-02-12 00:00:00.000000+0000 |       ECE |   Suresh

(3 rows)
cqlsh:employee> describe Employee_information;

CREATE TABLE employee.employee_information (
    emp_id int,
    salary float,
    date_of_joining timestamp,
    dept_name text,
    emp_name text,
    PRIMARY KEY (emp_id, salary)
) WITH CLUSTERING ORDER BY (salary ASC)
```

cqlsh:employee> select * from Employee_information where emp_id in (1,2,3) order by Salary;

```
cqlsh:employee> paging off
Disabled Query paging.
cqlsh:employee> select * from Employee_information where emp_id in (1,2,3) o
rder by Salary;

 emp_id | salary | date_of_joining                 | dept_name | emp_name
--------+--------+---------------------------------+-----------+----------
      2 |  10000 | 2020-06-21 00:00:00.000000+0000 |       ISE |    Tarun
      3 |  30000 | 2011-02-12 00:00:00.000000+0000 |       ECE |   Suresh
      1 |  50000 | 2021-04-23 00:00:00.000000+0000 |       CSE |   Nithin

(3 rows)
```

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.

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

8. Update the altered table to add project names.

cqlsh:employee> update employee_info set projects=projects+{'project1','project2','project3'} where emp_id=1;

```
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | emp_name | projects                                | salary
--------+---------------------------------+-----------+----------+-----------------------------------------+--------
      5 | 2016-01-09 00:00:00.000000+0000 |       CSE | Dharmesh |                                    null |  70000
      1 | 2021-04-23 00:00:00.000000+0000 |      Mech |  Sreekar | {'project1', 'project2', 'project3'} |  50000
      2 | 2020-06-21 00:00:00.000000+0000 |       ISE |    Tarun |                                    null |  10000
      4 | 2015-09-02 00:00:00.000000+0000 |       EEE |   Yuresh |                                    null |  90000
      3 | 2011-02-12 00:00:00.000000+0000 |       ECE |   Suresh |                                    null |  30000

(5 rows)
```

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

```
cqlsh:employee> begin batch
           ... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(6,'Rahul','2021-05-03',10000,'ISE') USING TTL 15;
           ... apply batch;
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | emp_name | projects                                | salary
--------+---------------------------------+-----------+----------+-----------------------------------------+--------
      5 | 2016-01-09 00:00:00.000000+0000 |       CSE | Dharmesh |                                    null |  70000
      1 | 2021-04-23 00:00:00.000000+0000 |      Mech |  Sreekar | {'project1', 'project2', 'project3'} |  50000
      2 | 2020-06-21 00:00:00.000000+0000 |       ISE |    Tarun |                {'project4', 'project5'} |  10000
      4 | 2015-09-02 00:00:00.000000+0000 |       EEE |   Yuresh |                                    null |  90000
      6 | 2021-05-03 00:00:00.000000+0000 |       ISE |    Rahul |                                    null |  10000
      3 | 2011-02-12 00:00:00.000000+0000 |       ECE |   Suresh |                                    null |  30000

(6 rows)
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | emp_name | projects                                | salary
--------+---------------------------------+-----------+----------+-----------------------------------------+--------
      5 | 2016-01-09 00:00:00.000000+0000 |       CSE | Dharmesh |                                    null |  70000
      1 | 2021-04-23 00:00:00.000000+0000 |      Mech |  Sreekar | {'project1', 'project2', 'project3'} |  50000
      2 | 2020-06-21 00:00:00.000000+0000 |       ISE |    Tarun |                {'project4', 'project5'} |  10000
      4 | 2015-09-02 00:00:00.000000+0000 |       EEE |   Yuresh |                                    null |  90000
      3 | 2011-02-12 00:00:00.000000+0000 |       ECE |   Suresh |                                    null |  30000

(5 rows)
```

## LAB 3:

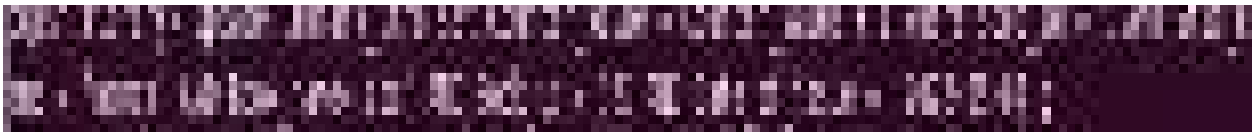
1. Create a key space by name Library

```
cqlsh> create keyspace Library WITH REPLICATION = {'class' : 'SimpleStrategy','replication_factor' :
1};
cqlsh> use Library;
```

2. Create a column family by name Library-Info with attributes Stud_Id Primary Key, Counter_value of type Counter,

```
cqlsh:library> create table Library_Info(Stud_Id int,Counter_value counter,Stud_Name varchar,Book_nam
e varchar,Book_Id int,Date_of_Issue date,primary key(Stud_Id,Stud_name,Book_name,Book_Id,Date_of_Issu
e));
```

3. Insert the values into the table in batch



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

```
cqlsh:library> update library_info set Counter_value = Counter_value + 1 where Stud_Id = 1 AND Stud_n
ame = 'naman' AND Book_name='abc' AND Book_Id = 123 AND Date_of_Issue = '2022-05-04';
cqlsh:library> select * from Library_Info;

 stud_id | stud_name | book_name | book_id | date_of_issue | counter_value
---------+-----------+-----------+---------+---------------+---------------
       1 |     naman |       abc |     123 |    2022-05-04 |             2
```

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

```
cqlsh:library> select counter_value as borrow_count from library_info where stud_id=1 AND book_id=123
;

 borrow_count
--------------
            2
```

6. Export the created column to a csv file

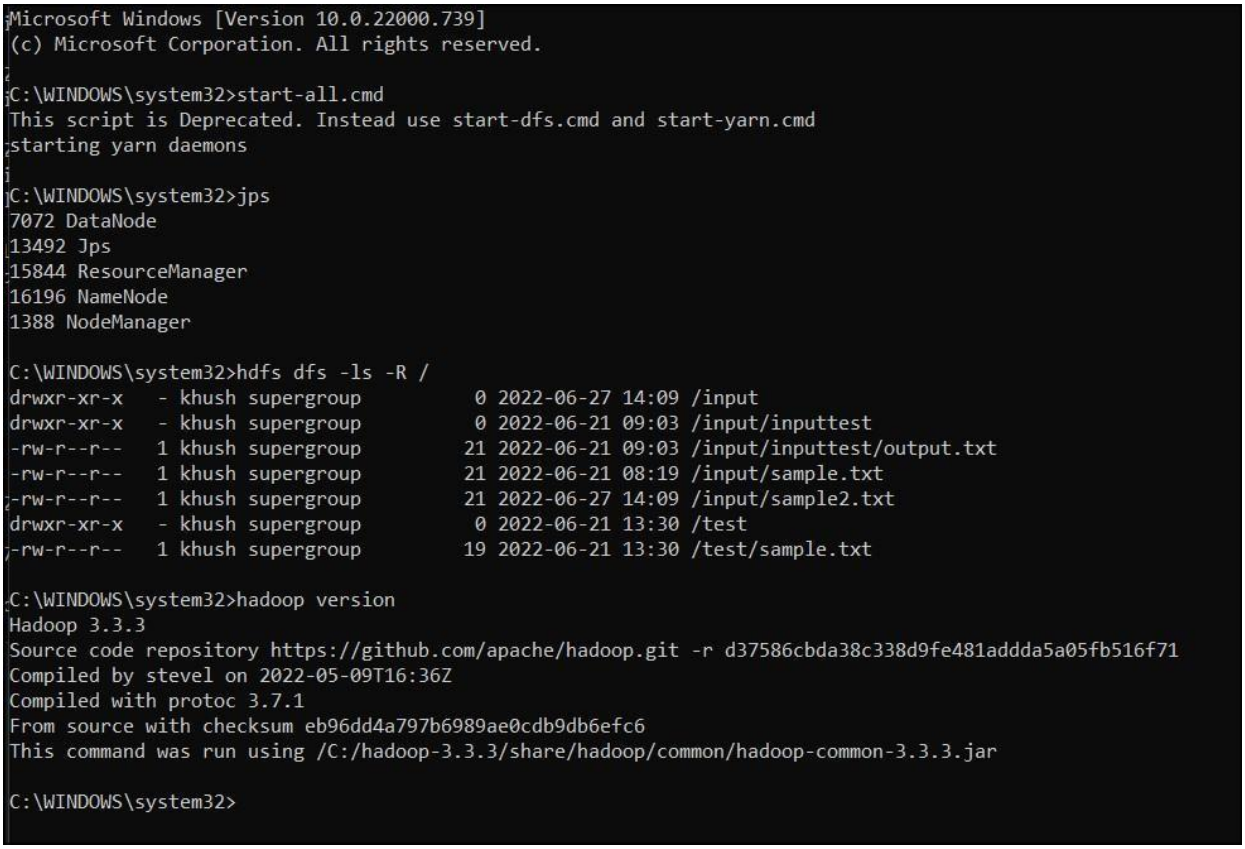
```
cqlsh:library> COPY library.library_info (Stud_Id,Book_Id,Counter_value,Stud_name,Book_name,Date_of_I
ssue) TO '/home/bmsce/CASSANDRA-NAMAN/data.csv' WITH HEADER = TRUE;
Using 11 child processes

Starting copy of library.library_info with columns [stud_id, book_id, counter_value, stud_name, book_
name, date_of_issue].
Processed: 1 rows; Rate:       0 rows/s; Avg. rate:       0 rows/s
1 rows exported to 1 files in 0.176 seconds.
```

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

Cassandra column family

```
cqlsh:library> COPY library.library_info (Stud_Id,Book_Id,Counter_value,Stud_name,Book_name,Date_of_I
ssue) FROM '/home/bmsce/CASSANDRA-NAMAN/data.csv' WITH HEADER = TRUE;
Using 11 child processes

Starting copy of library.library_info with columns [stud_id, book_id, counter_value, stud_name, book_
name, date_of_issue].
Processed: 1 rows; Rate:       2 rows/s; Avg. rate:       3 rows/s
1 rows imported from 1 files in 0.379 seconds (0 skipped).
```

Hadoop Installation




# Hadoop Commands

hdusersbmsce-OptiPlus-3000:-$ sudo su hduser
[sudo] password for hduser:

hdusersbmsce-OptiPlus-3000: $ 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: nanenade running as process 3396. Stop it first. localhost: datanode running as process 3564, Stop it first.
starting secondary nanenodes [0.0.0.0)

0.0.0.0: secondarynamenode running as process 3773. Stop it first. O22/06/06 14:43:47 WARN uttt.NativeCodeLoader: Unable to load native-hadoop library for your starting yarn daemons resource process 3932. Stop it first.
Localhost: running as process 4255. stop it
first. 6003 Jps
3932 ResourceManager
3773 SecondaryNameNode
4255 NodeManager
hdusersbmsce-OptiPlus-3060:-$ hdfs dfs -mkdir /khushil
hdusersbmsce-OptiPlus-3060: $ hdfs dfs -ls /
22/06/06 14:45:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable Found 19 itens
drwxr-xr-x hduser supergroup
02022- 06-06 11:44 /AAA
drwxr-xr-x -hduser
supergroup 2022-06-03 12:17
/Army drwxr-xr-x hduser
supergroup 02022- 06-06 11:40
/Avnit drwxr-xr-x -hduser
supergroup 02022-05-31 10:44 /88
drwxr-xr-x -hduser supergroup
02022- 06-01 15:03 /Cath
drwxr-xr-x -hduser supergroup
drwxr-xr-x hduser supergroup
drwxr-xr-x -hduser supergroup
drwxr-xr-x -hduser supergroup
drwxr-xr-x -hduser supergroup
drwxr-xr-x -hduser supergroup
drwxr-xr-x -hduser supergroup
drwxr-xr-x -hduser supergroup
drwxr-xr-x -hduser supergroup
drwxr-xr-x - hduser supergroup
drwxr-xr-x -hduser supergroup
82022-06-04 10:06 /FFF
02022-06-06 14:40 /Kmrv
02022-06-06 14:44 /Khushil
02022-06-01 15:03 /Neha
02022-06-04 09:54 /WC.txt
0 2022-06-04 09:54 /welcone.txt
02022-06-06 11:36 /abc
62022-06-03 12:13 /akash
0 2022-06-03 15:12 /darshan
0 2022-06-04 09:31 /ghh 8 2022-
06-06 11:45 /hello drwxr-xr-x -
hduser supergroup
62022-06-04 09:35 /rahul
drwxr-xr-x -hduser

supergroup 02022-06-03
12:11

/shre drwxr-xr-x .hduser
supergroup 02022-06-03
12:41
/shreshtha
hdusersbmsce-OptiPlus-3060:-$ hdfs dfs put /home/hduser/Desktop/6b.txt
/Khushil/WC.txt
22/05/06 14:46:40 WARN util.NativeCodeLoader: Unable to load native-hadoop
library for your platform... using butltin-java classes where applicable
hduserabesce-OptiPlex-3060:-$ hdfs dfs cat /Khushil/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 fron of
hdusersbmsce-OptiPlus-3040:-$ hdfs dfs-get /Khushil/WC.txt
/home/hduser/Downloads/newic.txt
22/05/06 14:51:43 WARN util.NativeCodeLoader: Unable to load nattve-hadoop
library for your platform... using builtin-java classes where applicable
hdusersbmsce-OptiPlus-3066:-$ cd Downloads
hdusersbmsce-OptiPlus-3060:-/Downloads$ cat newwMC.Ext hello from 6E
hdusersbmsce-OptiPlus-3060:-$ hdfs dfs -1s /Khushil/
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 itens
-rw-r--r-- 1 hduser supergroup
23 2822-06-06 14:46 /Khushil/MC.txt
1 hduser supergroup
23 2022-06-06 14:58 /Khushil/newwc.txt
hdusersbmsce-OptiPlus-3060:-5 hdfs drs -getmerge /Khushil/wc.txt
/Khushil/newwc.txt /bone/hduser/Desktop/newmerge.txt
22/06/06 14:55:18 NARN util.NativeCodeLoader: Unable to load nattve-hadoop
library for your platform... using butitin-Java classes where applicable
hduserabesce-OptiPlex- 3060:~$ cd Desktop
hduser@besce-OptiPlex-3060:-/Desktops cat newmerge.txt hello from 68
D B
hello from
68 D B
hdusersbmsce-OptiPlus-3060:-/Desktops hadoop fs getfacl /Khushil/ 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: /Khushil
# owner: hduser #
group:
supergroup
user::rwx
group::r-x
other::r-x
hdusersbmsce-OptiPlus-3060:-/Desktop5 hdfs dfs copyToLocal /Khushil/HC.txt
/home/hduser/Desktop
22/05/06 14:58:09 WARN util.NativeCodeLoader: Unable to load native-hadoop
Library for your platform... using butltin-java classes where applicable

hdusersbmsce-OptiPlus-3000:-
/Desktop5 cat MC.txt hello fron 68

hdusersbmsce-OptiPlus-3060:-/Desktops hdfs dfs -cat /Khushil/MC.txt 22/06/06 14:58:59 WARN util.NativeCodeLoader: Unable to load native-hadoop Library for your platform... ustng bulltin-Java classes where applicable hello from GB B
hdusersbmsce-OptiPlus-3060:-/Desktop5 hadoop fs - /Khushil /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
hduseransceOptiPlex-3060:-/Desktops hadoop fs-Ls /FFF 22/05/06 15:00:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using butltin- java classes where applicable Found 2 itens drwxrxr-x -hduser supergroup TWEE 1 hduser supergroup 02022-05-06 14:50
/FFF/Khushil 17 2022-05-04 10:06 /FFF/MC.txt
hdusersbmsce-OptiPlus-3060:-/Desktops hadoop fs cp /FFF/ /LLL
22/06/06 15:09:34 WARN util.NativeCodeLoader: Unable to load native hadoop library for your platform... using butltin-java classes where applicable
hdusersbmsce-OptiPlus-3060:-
/Desktops 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 1tens
drwxr-xr-x -hduser supergroup
hdusersbmsce-OptiPlus-3000:-/Desktops 02022-06-06 15:09
/LLL/KHUSHIL
17 2022-00-00 15:09 /LLL/MC.txt

# Hadoop Programs

## 1) Word Count

WCMapper Java Class file.

```
// Importing libraries import
java.io.IOException; import
org.apache.hadoop.io.IntWritable; import
org.apache.hadoop.io.LongWritable; import
org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper; import
org.apache.hadoop.mapred.OutputCollector; import
org.apache.hadoop.mapred.Reporter;

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

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

      String line = value.toString();

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

Reducer Code

```
// Importing libraries import java.io.IOException;
import java.util.Iterator; import
org.apache.hadoop.io.IntWritable; import
```

```
org.apache.hadoop.io.Text; import
org.apache.hadoop.mapred.MapReduceBase; import
org.apache.hadoop.mapred.OutputCollector; import
org.apache.hadoop.mapred.Reducer; import
org.apache.hadoop.mapred.Reporter;

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

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

      int count = 0;

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

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

Driver Code: // Importing libraries import
java.io.IOException; import
org.apache.hadoop.conf.Configured; import
org.apache.hadoop.fs.Path; import
org.apache.hadoop.io.IntWritable; import

```
org.apache.hadoop.io.Text; import
org.apache.hadoop.mapred.FileInputFormat; import
org.apache.hadoop.mapred.FileOutputFormat; import
org.apache.hadoop.mapred.JobClient; import
org.apache.hadoop.mapred.JobConf; import
org.apache.hadoop.util.Tool; import
org.apache.hadoop.util.ToolRunner;
public class WCDriver extends Configured implements Tool {


   public int run(String args[]) throws IOException


   {
      if (args.length < 2)
      {
         System.out.println("Please give valid inputs");
return -1;
      }

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

   // Main Method
   public static void main(String args[]) throws Exception
```

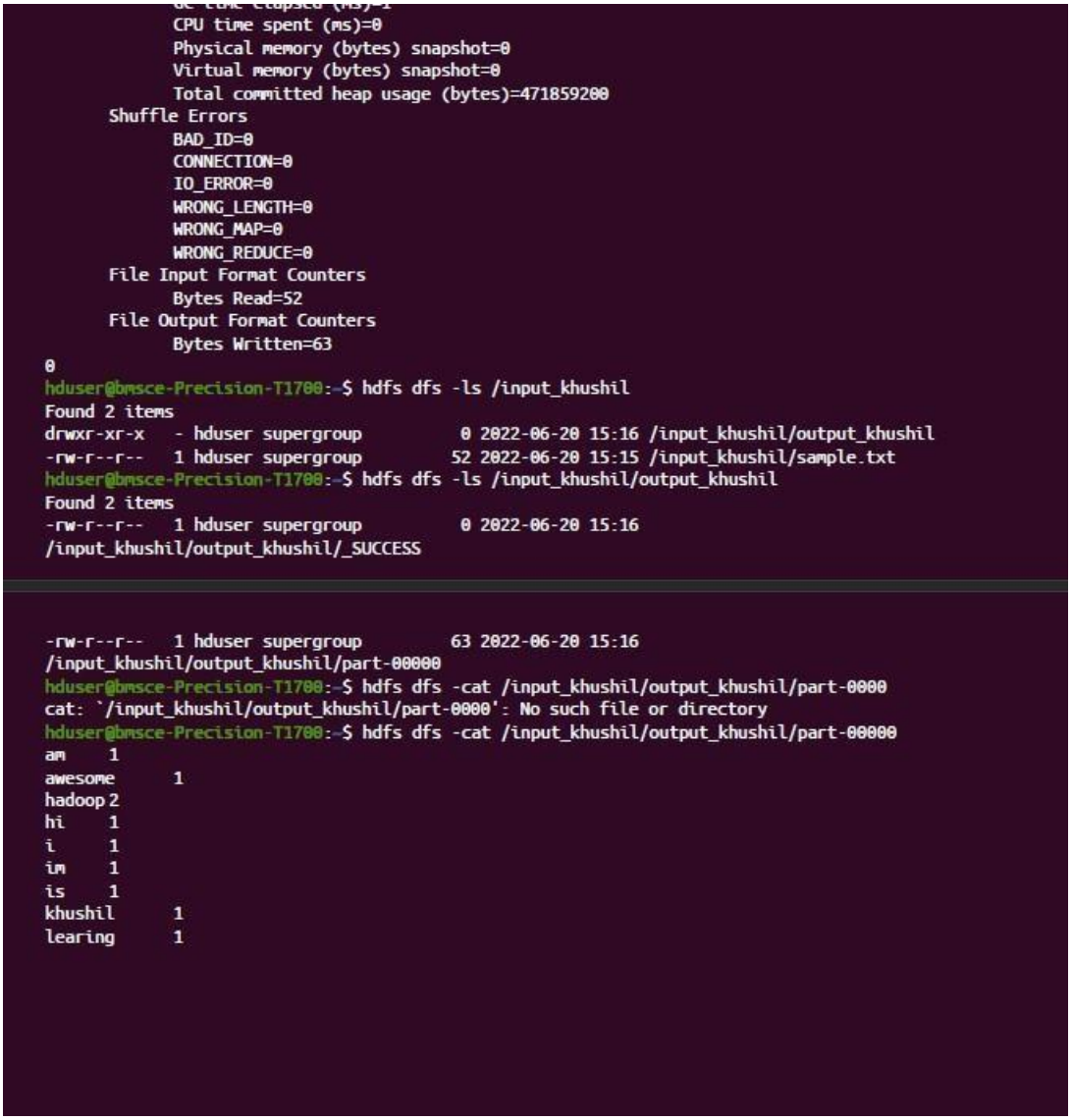
{

```
        int exitCode = ToolRunner.run(new WCDriver(), args);
        System.out.println(exitCode);
    }
}
```

Output :

```
hduser@bmsce-Precision-T1700:~$ su hduser\
> ^C
hduser@bmsce-Precision-T1700:~$ ^C
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: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-
Precision-T1700.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-
Precision-T1700.out
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-
secondarynamenode-bmsce-Precision-T1700.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-
Precision-T1700.out
hduser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-
bmsce-Precision-T1700.out
hduser@bmsce-Precision-T1700:~$ jps
7328 Jps
6497 DataNode
4372 org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar
6325 NameNode
7206 NodeManager
6872 ResourceManager
6713 SecondaryNameNode
hduser@bmsce-Precision-T1700:~$ cat > sample.txt
hi im khushil
i am learing hadoop
hadoop is awesome
^C
hduser@bmsce-Precision-T1700:~$ cat sample.txt
hi im khushil
i am learing hadoop
hadoop is awesome
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 18 items
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 2022-06-03 14:52 /bharath
drwxr-xr-x   - hduser supergroup          0 2022-06-03 14:43 /bharath035
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 2022-06-03 12:27 /irfan
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:09 /muskan
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 2022-06-20 12:17 /output
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:08 /saurab
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
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /input_khushil
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 19 items
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 2022-06-03 14:52 /bharath
drwxr-xr-x   - hduser supergroup          0 2022-06-03 14:43 /bharath035
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 15:13 /input_khushil
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:27 /irfan
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:09 /muskan
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 2022-06-20 12:17 /output
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:08 /saurab
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
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/sample.txt /input_khushil
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /input_khushil
Found 1 items
-rw-r--r--   1 hduser supergroup         52 2022-06-20 15:15 /input_khushil/sample.txt
hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/khushil/WordCount.jar WCDriver
/input_khushil /input_khushil/output_khushil
22/06/20 15:16:44 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/20 15:16:44 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker,
sessionId=
22/06/20 15:16:44 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
22/06/20 15:16:44 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not
performed. Implement the Tool interface and execute your application with ToolRunner to remedy
this.
22/06/20 15:16:44 INFO mapred.FileInputFormat: Total input paths to process : 1
22/06/20 15:16:44 INFO mapreduce.JobSubmitter: number of splits:1
22/06/20 15:16:44 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_local230197290_0001
22/06/20 15:16:44 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/20 15:16:44 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/20 15:16:44 INFO mapreduce.Job: Running job: job_local230197290_0001
22/06/20 15:16:44 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapred.FileOutputCommitter
22/06/20 15:16:44 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/20 15:16:44 INFO mapred.LocalJobRunner: Starting task:
attempt_local230197290_0001_m_000000_0
22/06/20 15:16:44 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/20 15:16:44 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/input_khushil/sample.txt:0+52
22/06/20 15:16:44 INFO mapred.MapTask: numReduceTasks: 1
22/06/20 15:16:44 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/20 15:16:44 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/20 15:16:44 INFO mapred.MapTask: soft limit at 83886080
22/06/20 15:16:44 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/20 15:16:44 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
```




## 2) Top N

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    vo```\\id    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.Reduc
er; 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));
    }
  }
}
```

Output:

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -mkdir /khushil_topn
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -put ./input.txt /khushil_topn/
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /khushil_topn/
Found 1 items
-rw-r--r--   1 hduser supergroup        103 2022-06-27 15:43 /khushil_topn/input.txt
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hadoop jar topn.jar TopNDriver
/khushil_topn/input.txt /khushil_topn/output
Exception in thread "main" java.lang.ClassNotFoundException: TopNDriver
 at 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
/khushil_topn/input.txt /khushil_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 : 1
22/06/27 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/khushil_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 in
the 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 : [ ]
```

```
 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=0
 Reduce input groups=15
 Reduce shuffle bytes=235
 Reduce input records=21
 Reduce output records=15
 Spilled Records=42
 Shuffled Maps =1
 Failed Shuffles=0
 Merged Map outputs=1
 GC time elapsed (ms)=42
 CPU 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_LENGTH=0
 WRONG_MAP=0
 WRONG_REDUCE=0
 File Input Format Counters
 Bytes Read=103
 File Output Format Counters
 Bytes Written=105
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /khushil_topn/output/
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-06-27 15:45 /khushil_topn/output/_SUCCESS
-rw-r--r--   1 hduser supergroup        105 2022-06-27 15:45 /khushil_topn/output/part-r-00000
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -cat /khushil_topn/output/part-r-00000
hadoop  4
i3
am      2
hi      1
im      1
is      1
there   1
bye     1
learing 1
awesome 1
love    1
khushil 1
cool    1
and     1
using   1
hduser@bmsce-Precision-T1700:~/Desktop/temperature$
```

## 3) Average Temperature

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));
}
}
```

Output:

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-
Precision-T1700.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-
Precision-T1700.out
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-
secondarynamenode-bmsce-Precision-T1700.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-
Precision-T1700.out
hduser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-bmsce-
Precision-T1700.out
hduser@bmsce-Precision-T1700:~/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
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_khushil
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:27 /irfan
drwxr-xr-x   - hduser supergroup          0 2022-06-22 10:44 /lwde
drwxr-xr-x   - hduser supergroup          0 2022-06-27 13:03 /mapreducejoin_amit
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
```

drvxr -xr -x - hduser supergroup druxrwxr -x - hduser supergroup druxr -xr -x - hduser supergroup druxr -xr -x - hduser supergroup
-rw-r--r-- 1 hduser Supergroup

Fou nd 2 ttems

e 26 22-e6-63 12. 08 /saura b e 2819-GB-61 16.19 /tnp
e 2619-GB-61 16. 63 *fu ser*
e 26 22-e6-61 69. 46 /user1
Z436 26 22- 6-24 IN- 17 /w. jar
S hdfs dfs -nkd-*tr* /khusht1_tenperature
S hdfs dfs - put . / 1901 /khushtT_tenperature S hdfs dfs - put . / 1902 /khushtT_tenperature S hdfs dfs -ls /khusht1_tenperature

-re- r - - r - - l hduser supergroup 88819€i 20 22- 6-27 14. 47 /k hushtt_teriperature/19e1
-re-r - - r - - l hduser superg *roup* 888978 20 22- e6 -27 14. 47 /k hushtt_terrperature/19e2
S hadoop jar . /avgtenp. jar AverageDrtver /khusht1_tetrpe rature/1P01 /khushtf_te erature/output/
ExceQtton in thread "main' )ava.lanQ.ClassMotFoundException. AverageOrt+r
at java .net .URLCtas sLoade r. ftndCtass{LIfiLCtassLoader . java. 382} at java .hang . CTassLoader . ToadC1ass(Class Loader .java.418)
at java .hang . CTassLoader . ToadC1ass(Class Loader java. 351) at java .hang . Class .forNz ee(Native Method )
at java .hang . Class .forNwe{CTa ss . java. 348)
at org. apache. hadoop. utit .Run3ar r un{Run 3 ar . j ava . 2t4) at org . apache. hadoop. uttt.Run3ar.natft{Run3ar. j ava - t36 }
S hadoop jar . /avgte p. jar temperature.AveraQe0river /khushtl_temperature/l9el /khushtl_temperature/output/
Z2/&6/27 14- 53.27 I'gFO Conftquratto n.depre atton . **session. Id ts** deprecated . **In stead. use** dfs *.net-r* c *s.* se sston -Id
22/86/27 14.53.27 MF0 j vn. *S* vrd *•ir -acs.* I ntttattztng JVfl Metrics vrtth processIJane=3obTracker, ses stonId=
22/g6/27 14.53.27 TURN napred uce . 3ob5ubn" *ther* .- Hadoop command - Ttr›e optton par stng not per fori•ted . Imp ent tke Tool interface and execute your application 'tth TootRunner to rertecly thts .
Z2/0•6/27 14:S3.27 NFO input.FileInputFormat: Total input paths to process - 1
22/&6/27 14c53- 27 MF0 napreduce 3obSulx ttter .- nunber of spltts- 1
22/06/27 14-S3:28 lWFO mapreduce.JDb5ubmitter: SubmtttinQ tokens for job-job_local254968295_0B81
22/g6/27 14.53.28 INFO napreduce 3ob. Tke *ord* to track the job http. //localhost .8080/
22/&6/27 14.53.28 MF0 napreduce 3ob. Re nntng job jod_tocal Z50968295_886t
22/96/27 14- 53.26 INFO napred Local JobRu nner. 0utputC ttter set tn conftg null
22/06/27 14.53.28 MF0 napred . Local JobRu nner. OutputCunrittter I s *org* .apache . hadoop. ropreduce .\.tb.out put . FtTeDztputCormttter
22/&6/27 14.53.28 INFO mapred LocaT3obRu nner.- batting *lar* nap tasks
Z2/06/27 14.53.28 INFO mapred.LocalJobRunner. Starting task. attempt_localZ5496B29S_Be01_m_BBO088_B
Z2/g6/27 14- 5 3 28 INFO napred Task-Using 9esourceCatculatorProces sTree [ ]
22/86/27 14-53- 28 INFO mapred RapTa sk- Processing spTtt- hdfs . //Coca\.host . 5431e/khushtt_teriperature/1961 0+B8819O
22/&6/27 14.53.28 IfiF0 napred RapTa sk. EgtlATCOt ) e kvt 26214396 104857584 } 22/06/27 14.53.28 MF0 napred . ñapTa sk. napreduce .task . to sort . nb. ITB 22/86/27 14.53.26 INFO napred . RapTa sk. soft ttmtt at 8388608e
22/86/27 14.53.28 INFO napred . RapTa sk. bufs tart = 0; bufvotd = 10485760e
22/&6/27 14 -53.28 16F0 rrapred RapTa sk. *kvs4art* -- 26214396 ; length = 6553660 22/06/27 14.53.28 IfiF0 mapred RapTa sk. flap output collector eta ss = *org* .apache . hadoop . napred .NapTask$?'IajXMtputBuffer

mapred . Star ttnq flush of

22/&6/27 MF0 mapred RapTa Spitting nap

. = =

. RapTa = =

=

. spilt 8

```
        FILE: Number of bytes written=723014
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: 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=6564
        Map 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=72210
        Reduce input records=6564
        Reduce output records=1
        Spilled Records=13128
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=55
        CPU 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_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
        File Input Format Counters
        Bytes Read=888190
        File Output Format Counters
        Bytes Written=8
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /khushil_temperature/output/
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-06-27 14:53 /khushil_temperature/output/_SUCCESS
-rw-r--r--   1 hduser supergroup          8 2022-06-27 14:53 /khushil_temperature/output/part-r-
00000
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -cat /khushil_temperature/output/part-
r-00000
1901    46
hduser@bmsce-Precision-T1700:~/Desktop/temperature$
```

4) Join

```
// 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);
   }
```

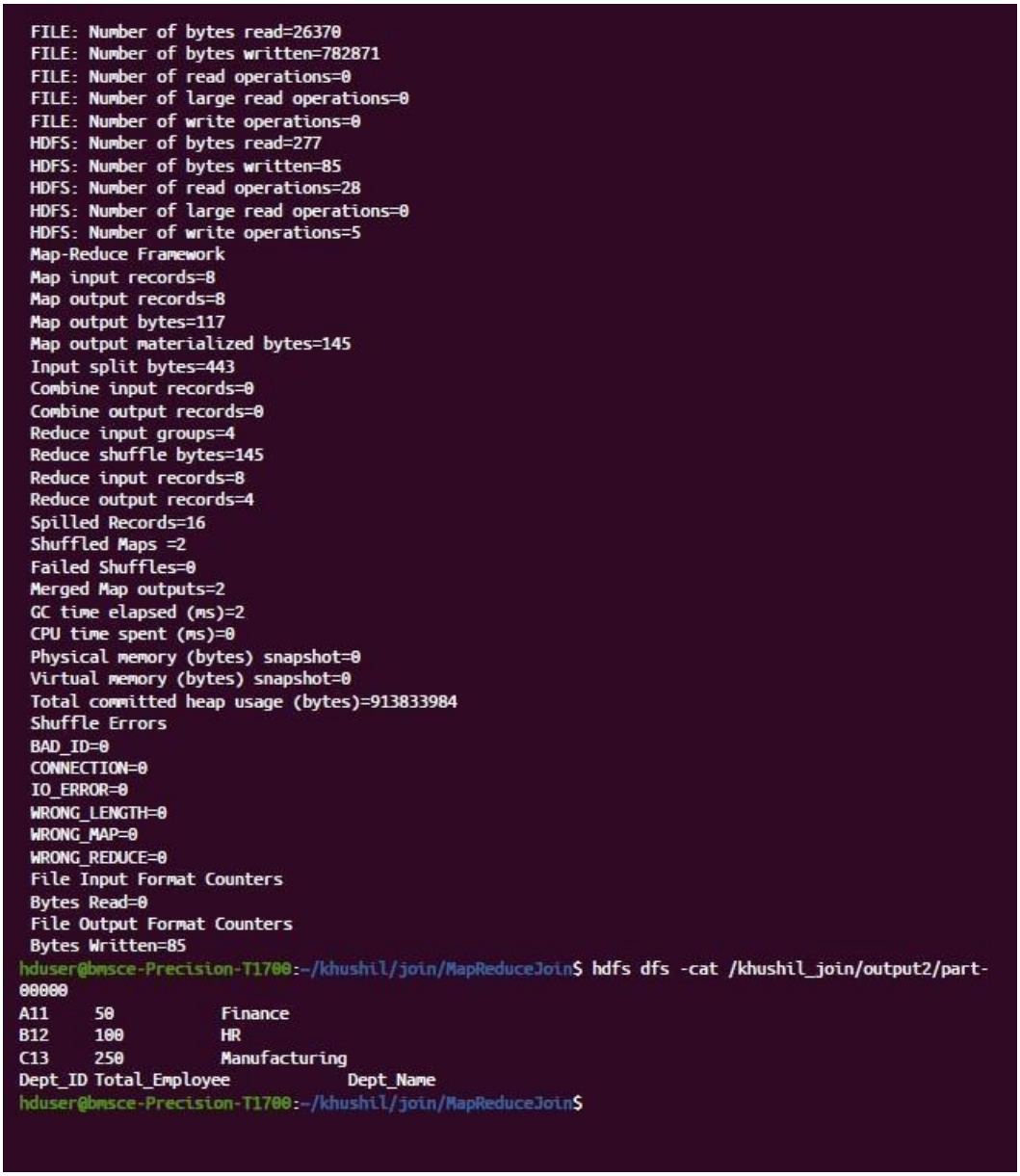
```
   @Override
   public void readFields(DataInput in) throws IOException {
first.readFields(in);
      second.readFields(in);
   }

   @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);
        }
    }
}
```

Output:

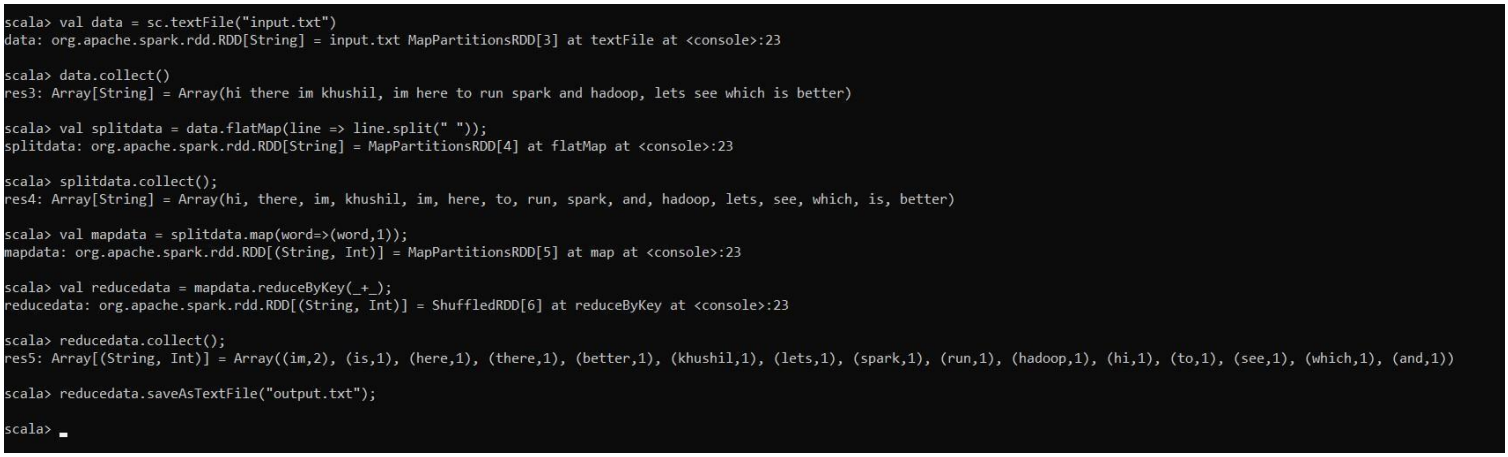
```
hduser@bmsce-Precision-T1700:~/khushil/join/MapReduceJoin$ hdfs dfs -ls /khushil_join
ls: `/khushil_join': No such file or directory
hduser@bmsce-Precision-T1700:~/khushil/join/MapReduceJoin$ hdfs dfs -mkdir /khushil_join
hduser@bmsce-Precision-T1700:~/khushil/join/MapReduceJoin$ hdfs dfs -ls /khushil_join
hduser@bmsce-Precision-T1700:~/khushil/join/MapReduceJoin$ hdfs dfs -put ./DeptName.txt
/khushil_join/
hduser@bmsce-Precision-T1700:~/khushil/join/MapReduceJoin$ hdfs dfs -put ./DeptStrength.txt
/khushil_join/
hduser@bmsce-Precision-T1700:~/khushil/join/MapReduceJoin$ hadoop jar MapReduceJoin.jar
/khushil_join/DeptName.txt /khushil_join/DeptStrength.txt /khushil_join/output/
22/06/27 15:12:24 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/27 15:12:24 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker,
sessionId=
22/06/27 15:12:24 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with processName=JobTracker,
sessionId= - already initialized
22/06/27 15:12:24 INFO mapred.FileInputFormat: Total input paths to process : 1
22/06/27 15:12:24 INFO mapred.FileInputFormat: Total input paths to process : 1
22/06/27 15:12:24 INFO mapreduce.JobSubmitter: number of splits:2
22/06/27 15:12:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1238804660_0001
22/06/27 15:12:24 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/27 15:12:24 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/27 15:12:24 INFO mapreduce.Job: Running job: job_local1238804660_0001
22/06/27 15:12:24 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapred.FileOutputCommitter
22/06/27 15:12:24 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/27 15:12:24 INFO mapred.LocalJobRunner: Starting task: attempt_local1238804660_0001_m_000000_0
22/06/27 15:12:24 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/27 15:12:24 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/khushil_join/DeptName.txt:0+59
22/06/27 15:12:24 INFO mapred.MapTask: numReduceTasks: 1
22/06/27 15:12:24 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/27 15:12:24 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
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 = 6553600
22/06/27 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 output
22/06/27 15:12:24 INFO mapred.MapTask: Spilling map output
22/06/27 15:12:24 INFO mapred.MapTask: bufstart = 0; bufend = 63; 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_000000_0 is done. And is in
the process of committing
22/06/27 15:12:24 INFO mapred.LocalJobRunner: hdfs://localhost:54310/khushil_join/DeptName.txt:0+59
22/06/27 15:12:24 INFO mapred.Task: Task 'attempt_local1238804660_0001_m_000000_0' done.
22/06/27 15:12:24 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1238804660_0001_m_000000_0
22/06/27 15:12:24 INFO mapred.LocalJobRunner: Starting task: attempt_local1238804660_0001_m_000001_0
22/06/27 15:12:24 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/27 15:12:24 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/khushil_join/DeptStrength.txt:0+50
22/06/27 15:12:24 INFO mapred.MapTask: numReduceTasks: 1
22/06/27 15:12:24 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/27 15:12:24 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
```




Scala Programming:
Lab 9:

```
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;
```




## Lab 10:

```
val textFile = sc.textFile("/home/bhoom/Desktop/wc.txt")
val counts = textFile.flatMap(line => line.split(" ")).map(word => (word,
1)).reduceByKey(_ + _)
import scala.collection.immutable.ListMap

val sorted=ListMap(counts.collect.sortWith(_._2 > _._2):_*)// sort in
descending order based on values
println(sorted) for((k,v)<-sorted)
{ if(v>4)
```

```
{ print(k+",")
print(v)
println()
}}
```

```
scala> val filerdd = sc.textFile("input.txt");
filerdd: org.apache.spark.rdd.RDD[String] = input.txt MapPartitionsRDD[13] at textFile at <console>:24

scala> val counts = filerdd.flatMap(line=>line.split(" ")).map(word=>(word,1)).reduceByKey(_+_);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[16] at reduceByKey at <console>:24

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(im -> 2, is -> 1, here -> 1, there -> 1
, better -> 1, khushil -> 1, lets -> 1, spark -> 1, run -> 1, hadoop -> 1, hi -> 1, to -> 1, see -> 1, w
hich -> 1, and -> 1)

scala> println(sorted);
ListMap(im -> 2, is -> 1, here -> 1, there -> 1, better -> 1, khushil -> 1, lets -> 1, spark -> 1, run -
> 1, hadoop -> 1, hi -> 1, to -> 1, see -> 1, which -> 1, and -> 1)

scala> for((k,v)<-sorted)
     | {
     | if(v>4)
     | {
     | print(k+",")
     | print(v)
     | println()
     | }
     | }

scala> for((k,v)<-sorted)
     | {
     | println(k+",")
     | println(v)
     | println()
     | }
im,
2

is,
1

here,
1

there,
1

better,
1

khushil,
1

lets,
1

spark,
1
```