VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

BIG DATA ANALYTICS

Submitted by

VARUN RAJ S (1BM21CS264)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING

in
COMPUTER SCIENCE AND ENGINEERING

Under the guidance of
Prof. Saritha A N
Assistant Professor
Department of CSE
B.M.S.C.E., Bengaluru

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

Mar-2024 to July-2024

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

CERTIFICATE

This is to certify that the Lab work entitled "LAB COURSE **BIG DATA ANALYTICS**" carried out by **VARUN RAJ S (1BM21CS264),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics - (22CS6PEBDA)** work prescribed for the said degree.

Name of the Lab-Incharge:

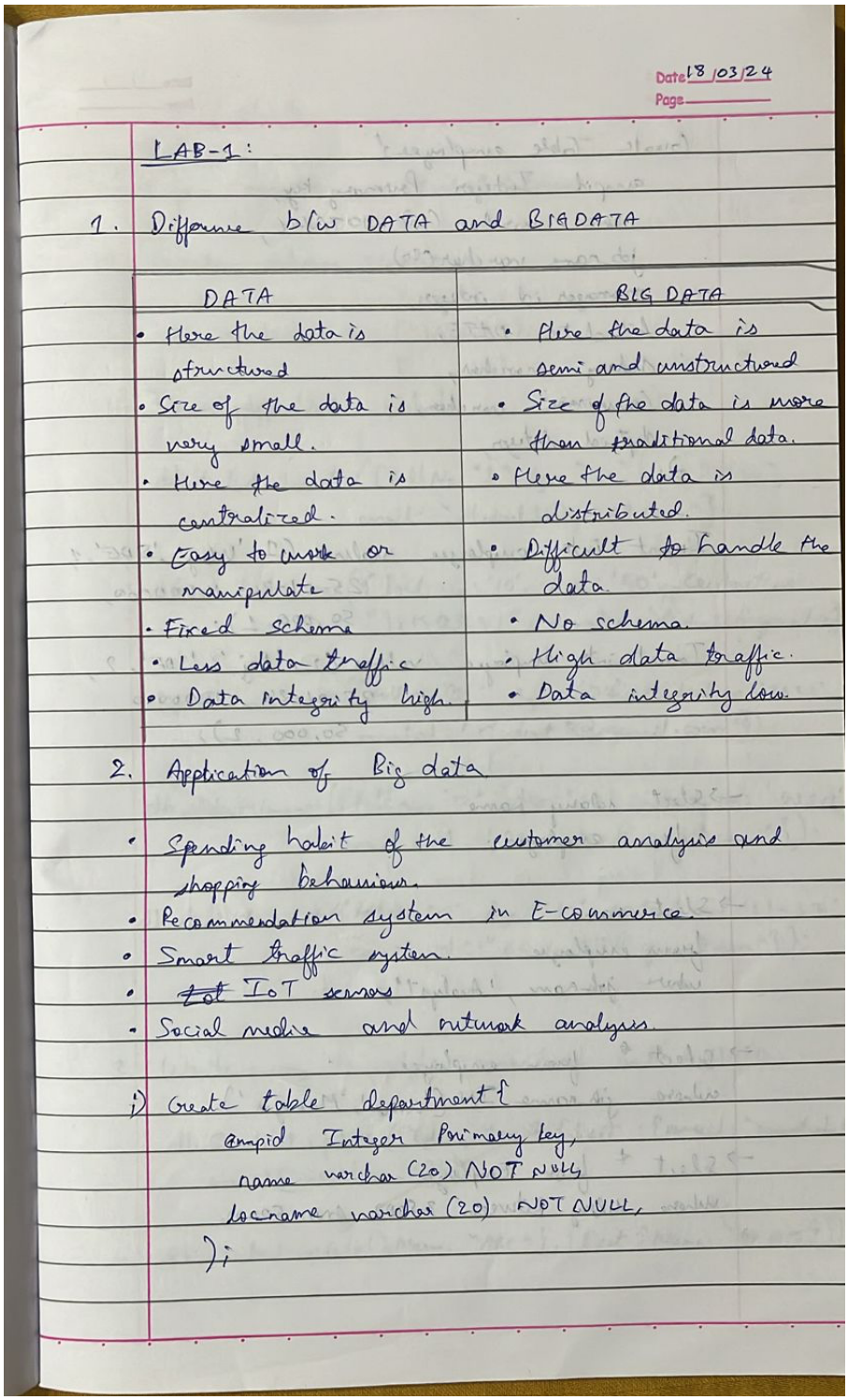
Saritha A N	**Dr. Jyothi S Nayak**
Assistant Professor.	Professor and Head of Dept
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

Index Sheet

Sl. No.	Experiment Title	Page No.
1	Question and answer & SQL	4
2	MongoDB- CRUD	5
3	Perform the following DB operations using Cassandra-Student Database	8
4	Cassandra-Employee Database	12
5	Hadoop installation	14
6	HDFS Commands	15
7	Implement WordCount Program on Hadoop framework	19
8	Create a Map Reduce program to a) find average temperature for each year from NCDC data set. b) find the mean max temperature for every month	24
9	For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words	32

1. QUESTION AND ANSWER

Fig1.: Lab 1 observation



Date 18/03/24
Page

LAB-1:

1. Diffrence b/w DATA and BIGDATA

DATA	BIG DATA
• Here the data is structured	• Here the data is semi and unstructured
• Size of the data is very small.	• Size of the data is more than traditional data.
• Here the data is centralized.	• Here the data is distributed.
• Easy to work or manipulate	• Difficult to handle the data
• Fixed schema	• No schema.
• Less data traffic	• High data traffic.
• Data integrity high.	• Data integrity low.

2. Application of Big data

- Spending habit of the customer analysis and shopping behaviour.
- Recommendation system in E-commerce.
- Smart traffic system.
- ~~IoT~~ IoT ~~sensors~~
- Social media and network analysis.

i) Create table department {
deptid Integer Primary key,
name varchar (20) NOT NULL
locname varchar (20) NOT NULL,
);

2. MongoDB- CRUD Demonstration

Fig.2: Inserting into database:

```
test> use Student
switched to db Student
Student> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
```

Fig.3: Displaying inserted values

```
}
Student> db.Student.find()
[
  {
    _id: ObjectId('660a86053f257f0a2b66fd9b'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86063f257f0a2b66fd9c'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86063f257f0a2b66fd9d'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86063f257f0a2b66fd9e'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86083f257f0a2b66fd9f'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'abhinav@gmail.com'
  }
```

Fig. 4: Updating values:

```
Student> db.Student.update({RollNo:10},{$set:{email:"abhinav@gmail.com"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Student> db.Student.update({RollNo:11, Name:"ABC"},{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

Fig.5: Creating Customers database and inserting.

```
Student> db.createCollection("Customers");
{ ok: 1 }
Student> db.Customers.insert({cust_id:1,Balance:200, Type:"S"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda0') }
}
Student>

Student> db.Customers.insert({cust_id:1,Balance:1000, Type:"Z"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda1') }
}
Student>

Student> db.Customers.insert({cust_id:2,Balance:100, Type:"Z"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda2') }
}
Student>

Student> db.Customers.insert({cust_id:2,Balance:1000, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda3') }
}
Student>

Student> db.Customers.insert({cust_id:2,Balance:500, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda4') }
}
```

Fig.6: Updating

```
Student> db.Customers.aggregate (
...
... {$group : { _id : "$cust_id",
...
... minAccBal :{$min:"$Balance"},
... maxAccBal :{$max:"$Balance"} }});
[
  { _id: 3, minAccBal: 500, maxAccBal: 500 },
  { _id: 2, minAccBal: 50, maxAccBal: 1000 },
  { _id: 1, minAccBal: 200, maxAccBal: 1000 }
]
Student> db.Customers.aggregate(
... {$match:{Type:"Z"}},
... {$group:{_id:"$cust_id",
... TotAccBal:{$sum:"$Balance"}}},
... {$match:{TotAccBal:{$gt:1200}}});
```

3. Perform the following DB operations using Cassandra.

bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Students WITH REPLICATION={
'class':'SimpleStrategy','replication_factor':1
};

cqlsh> DESCRIBE KEYSPACES
students system_auth, system_schema system_views system
system_distributed system_traces system_virtual_schema

cqlsh> SELECT * FROM system.schema_keyspaces;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table schema_keyspaces does not exist"

cqlsh> use Students;
cqlsh:students> create table Students_info(Roll_No int Primary key,StudName text,DateOfJoining timestamp,last_exam_Percent double);

cqlsh:students> describe tables;
students_info

cqlsh:students> describe table students;
Table 'students' not found in keyspace 'students'

cqlsh:students> describe table students_info;
CREATE TABLE students.students_info (
roll_no int PRIMARY KEY,
dateofjoining timestamp,
last_exam_percent double,
studname text
) WITH additional_write_policy = '99p'
AND bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND cdc = false
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}

```
        AND compression = {'chunk_length_in_kb': '16', 'class':
'org.apache.cassandra.io.compress.LZ4Compressor'}
        AND memtable = 'default'
        AND crc_check_chance = 1.
       AND default_time_to_live = 0
       AND extensions = {}
        AND gc_grace_seconds = 864000
        AND max_index_interval = 2048
        AND memtable_flush_period_in_ms = 0
        AND min_index_interval = 128
        AND read_repair = 'BLOCKING'
        AND speculative_retry = '99p';
```

cqlsh:students> Begin batch

```
insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent)
values(1,'Srivatsa','2023-10-09', 98)
insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent)
values(2,'Varun','2023-10-10', 97)
insert into Students_info(Roll_no, StudName,DateOfJoining,
last_exam_Percent)
values(3,'Prithvi','2023-10-10', 97.5)
insert into Students_info(Roll_no, StudName,DateOfJoining,
last_exam_Percent)
values(4,'Uday','2023-10-06', 96.5)
apply batch;
```

cqlsh:students> select * from students_info;

```
 roll_no | dateofjoining                  | last_exam_percent | studname
        +                               +                   +
        1 | 2023-10-08 18:30:00.000000+0000 |              98 | Srivatsa
        2 | 2023-10-09 18:30:00.000000+0000 |              97 | Varun
        4 | 2023-10-05 18:30:00.000000+0000 |              96.5 | Uday
        3 | 2023-10-09 18:30:00.000000+0000 |              97.5 | Prithvi
       (4 rows)
```

cqlsh:students> select * from students_info where roll_no in (1,2,3);

```
 roll_no | dateofjoining                  | last_exam_percent | studname
        +                               +                   +
        1 | 2023-10-08 18:30:00.000000+0000 |              98 | Srivatsa
        2 | 2023-10-09 18:30:00.000000+0000 |              97 |Varun
        3 | 2023-10-09 18:30:00.000000+0000 |              97.5 | Prithvi
```

cqlsh:students> select * from students_info where Studname='Uday';

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute
```

this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"

cqlsh:students> create index on Students_info(StudName);
cqlsh:students> select * from students_info where Studname=Uday;

```
 roll_no | dateofjoining                  | last_exam_percent | studname
         +                              +                   +
       4 | 2023-10-05 18:30:00.000000+0000 |              96.5 | Uday

(1 rows)
```

cqlsh:students> select Roll_no,StudName from students_info LIMIT 2;

```
 roll_no | studname
         +
       1 | Srivatsa
       2 |Rohan
      (2 rows)
```

cqlsh:students> SELECT Roll_no as "USN" from Students_info;

```
 USN
   1
   2
   4
   3

(4 rows)
```

cqlsh:students> update students_info set StudName='Shreya' where Roll_no=3;
cqlsh:students> select * from students_info;

```
 roll_no | dateofjoining                  | last_exam_percent | studname
         +                              +                   +
       1 | 2023-10-08 18:30:00.000000+0000 |                98 | Srivatsa
       2 | 2023-10-09 18:30:00.000000+0000 |                97 | Varun
       4 | 2023-10-05 18:30:00.000000+0000 |               96.5 | Uday
       3 | 2023-10-09 18:30:00.000000+0000 |              97.5 | Prithvi

(4 rows)
```

cqlsh:students> update students_info set roll_no=8 where Roll_no=3;
InvalidRequest: Error from server: code=2200 [Invalid query] message="PRIMARY KEY part roll_no found in SET part"

cqlsh:students> delete last_exam_percent from students_info where roll_no=2;
cqlsh:students> select * from students_info;

```
 roll_no | dateofjoining                  | last_exam_percent | studname
        +                               +                   +
       1 | 2023-10-08 18:30:00.000000+0000 |              98 | Srivatsa
       2 | 2023-10-09 18:30:00.000000+0000 |            null | Varun
       4 | 2023-10-05 18:30:00.000000+0000 |            96.5 |  Uday
       3 | 2023-10-09 18:30:00.000000+0000 |            97.5 |  Prithvi
(4 rows)
```

cqlsh:students> delete from students_info where roll_no=2;
cqlsh:students> select * from students_info;

```
 roll_no | dateofjoining                  | last_exam_percent | studname
        +                               +                   +
       1 | 2023-10-08 18:30:00.000000+0000 |              98 | Srivatsa
       4 | 2023-10-05 18:30:00.000000+0000 |            96.5 |  Uday
       3 | 2023-10-09 18:30:00.000000+0000 |            97.5 |  Prithvi

      (3 rows
```

4. Employee Database

Fig.7: Employee commands and queries SS-1:

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> create keyspace Employee with replication = {'class':'SimpleStrategy;,;replicationfactor':1};
SyntaxException: line 1:89 mismatched input ':' expecting '}' (...with replication = {'class':'SimpleStrategy;,;replicationfactor'[:]1...)
cqlsh> create keyspace Employee WITH replication={'class':'SimpleStrategy','replicationfactor':1};
ConfigurationException: Unrecognized strategy option {replicationfactor} passed to SimpleStrategy for keyspace employee
cqlsh> create keyspace Employee WITH replication={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES

employee  system_auth         system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

cqlsh> CREATE TABLE IF NOT EXISTS Employee_Info(
   ... Emp_Id INT PRIMARY KEY,
   ... Emp_name TEXT,
   ... designation TEXT,
   ... date_of_joining DATE,
   ... Salary FLOAT,
   ... Dep_name TEXT,
   ... Projects SET<TEXT>);
InvalidRequest: Error from server: code=2200 [Invalid query] message="No keyspace has been specified. USE a keyspace, or explicitly specify keyspace.tablename"
cqlsh> USE eMPLOYEE
   ...
cqlsh> USE Employee
   ...
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_Info( Emp_Id INT PRIMARY KEY, Emp_name TEXT, designation TEXT, date_of_joining DATE, Salary FLOAT, Dep_name TEXT, Projects SET<TEXT>);
cqlsh:employee> describe keyspace Employee

CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'}  AND durable_writes = true;

CREATE TABLE employee.employee_info (
    emp_id int PRIMARY KEY,
    date_of_joining date,
    dep_name text,
    designation text,
    emp_name text,
    salary float,
    projects set<text>
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
```

Fig.8: Employee commands and queries SS-2:

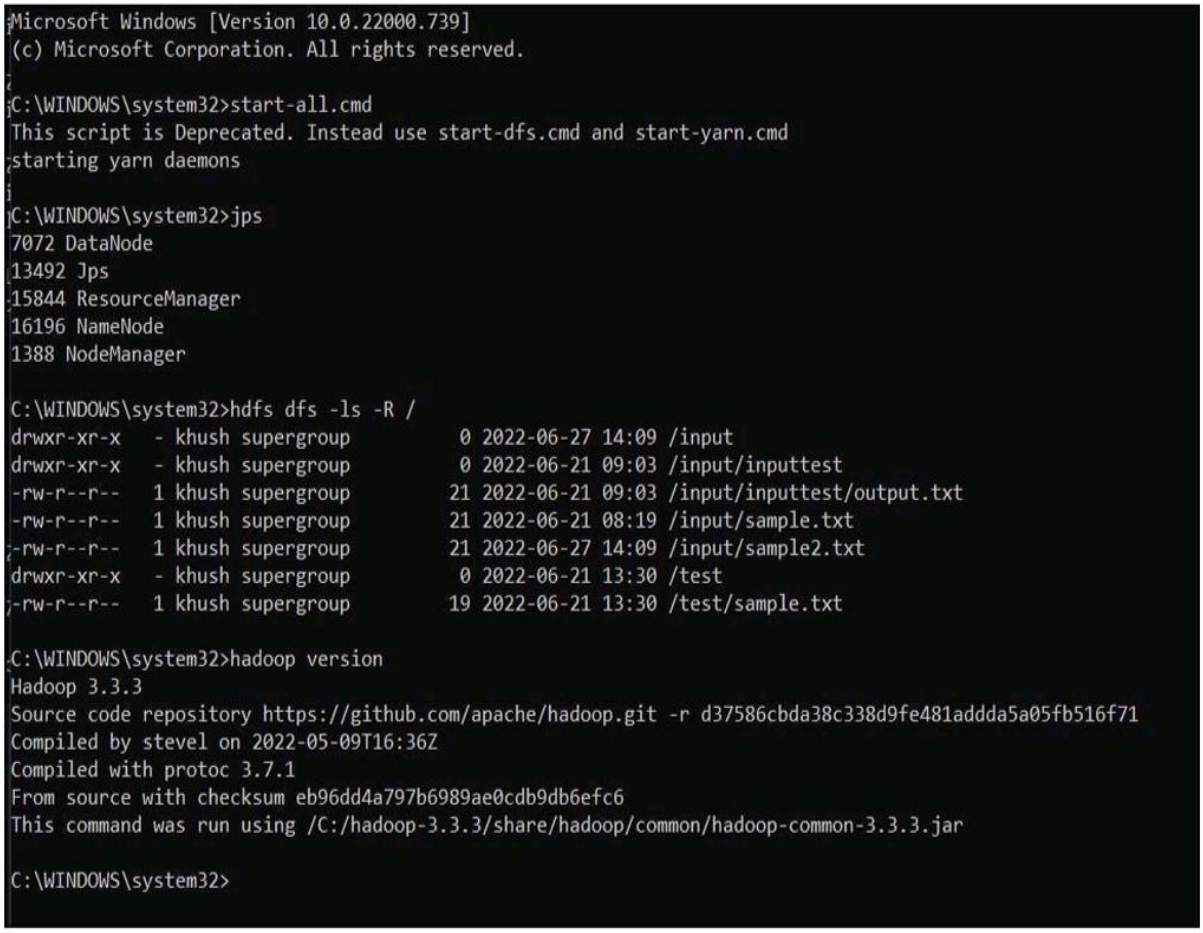
```
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqlsh:employee> select * from employee_info;

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                    | salary
--------+-------+-----------------+-------------+-------------+-------------+-----------------------------+---------
    120 | 12000 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 | 11000 |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |       0

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

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                    | salary
--------+-------+-----------------+-------------+-------------+-------------+-----------------------------+---------
    120 | 12000 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 | 11000 |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |    null

(4 rows)
cqlsh:employee>
```

Fig.9: Employee commands and queries SS-3:

```
   AND speculative_retry = '99p';
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining | dep_name    | designation | emp_name | projects                  | salary
--------+-----------------+-------------+-------------+----------+---------------------------+---------
    120 |      2024-05-06 | Engineering |   Developer | Priyanka |  {'Project B', 'ProjectA'} |   1e+06
    123 |      2024-05-07 | Engineering |    Engineer |  Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |      2024-05-06 |  Management |          HR |  Rachana | {'Project C', 'Project M'} |   9e+05
    121 |      2024-05-06 |  Management |   Developer |   Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id = '120';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid STRING constant (120) for "emp_id" of type int"
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id=120;
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining | dep_name    | designation | emp_name    | projects                  | salary
--------+-----------------+-------------+-------------+-------------+---------------------------+---------
    120 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> select * from employee_info order by salary;
InvalidRequest: Error from server: code=2200 [Invalid query] message="ORDER BY is only supported when the partition key is restricted by an EQ or an IN."
cqlsh:employee> alter table employee_info add bonus INT;
cqlsh:employee> select * from employee_info;

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                  | salary
--------+-------+-----------------+-------------+-------------+-------------+---------------------------+---------
    120 |  null |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 |  null |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> update employee_info set bonus = 12000 where emp_id = 120;
cqlsh:employee> select * from employee_info;

 emp_id | bonus | date_of_joining | dep_name    | designation | emp_name    | projects                  | salary
--------+-------+-----------------+-------------+-------------+-------------+---------------------------+---------
    120 | 12000 |      2024-05-06 | Engineering |   Developer | Priyanka GH |  {'Project B', 'ProjectA'} |   1e+06
    123 |  null |      2024-05-07 | Engineering |    Engineer |     Sadhana | {'Project M', 'Project P'} | 1.2e+06
    122 |  null |      2024-05-06 |  Management |          HR |     Rachana | {'Project C', 'Project M'} |   9e+05
    121 |  null |      2024-05-06 |  Management |   Developer |      Shreya |  {'Project C', 'ProjectA'} |   9e+05

(4 rows)
cqlsh:employee> update employee_info set bonus = 11000 where emp_id = 121;
cqlsh:employee> select * from employee_info using ttl 15 where emp_id = 123;
SyntaxException: line 1:28 mismatched input 'using' expecting EOF (select * from employee_info [using] ttl...)
cqlsh:employee> select * from employee_info where emp_id = 121 using ttl 15;
SyntaxException: line 1:47 no viable alternative at input 'using' (...employee_info where emp_id = 121 [using]...)
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqlsh:employee> select * from employee_info;
```

5. Hadoop Installation

Fig.10: Hadoop installation commands screenshot:



6. Hadoop Hdfs commands

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh

WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.

WARNING: This is not a recommended production deployment configuration.

WARNING: Use CTRL-C to abort.

Starting namenodes on [localhost]

Starting datanodes

Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]

Starting resourcemanager

Starting nodemanagers

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop dfs -mkdir /sadh WARNING: Use of this script to execute dfs is deprecated.

WARNING: Attempting to execute replacement "hdfs dfs" instead.

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -mkdir /sadh

mkdir: `/sadh': File exists

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /

Found 1 items

drwxr-xr-x - hadoop supergroup 0 2024-05-13 14:27 /sadh

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /sadh

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/example/Welcome.txt /sadh/WC.txt

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /sadh/WC.txt hiiii

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -get /sadh/WC.txt /home/hadoop/Desktop/example/WWC.txt

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -get /sadh/WC.txt /home/hadoop/Desktop/example/WWC2.txt

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/example/Welcome.txt /sadh/WC2.txt

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -getmerge /sadh/WC.txt /sadh/WC2.txt /home/hadoop/Desktop/example/Merge.txt

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -getfacl

/sadh/ # file: /sadh

owner: hadoop

group: supergroup

user::rwx

group::r-x

other::r-x

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mv /sadh /WC2.txt

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /sadh /WC2.txt

ls: `/sadh': No such file or directory

Found 2 items

-rw-r--r-- 1 hadoop supergroup 6 2024-05-13 14:51 /WC2.txt/WC.txt
-rw-r--r-- 1 hadoop supergroup 6 2024-05-13 15:03 /WC2.txt/WC2.txt

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cp /WC2.txt/ /WC.txt

Fig.11: Hadoop commands screenshot-1:

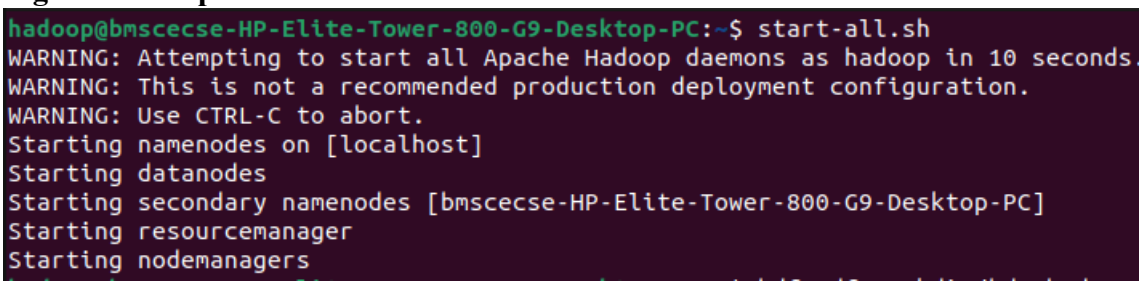


Fig.12: Hadoop commands screenshot-2:

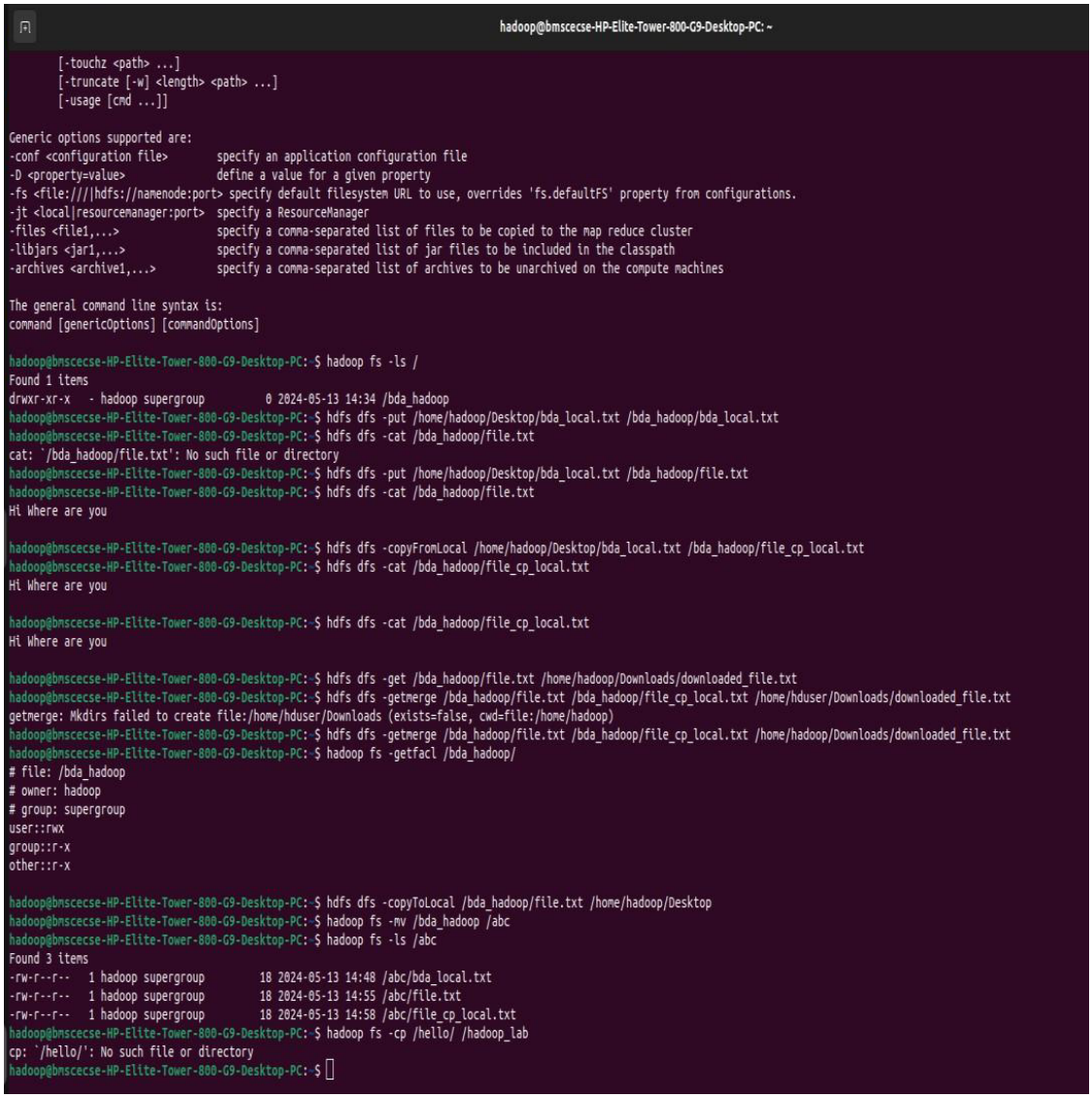
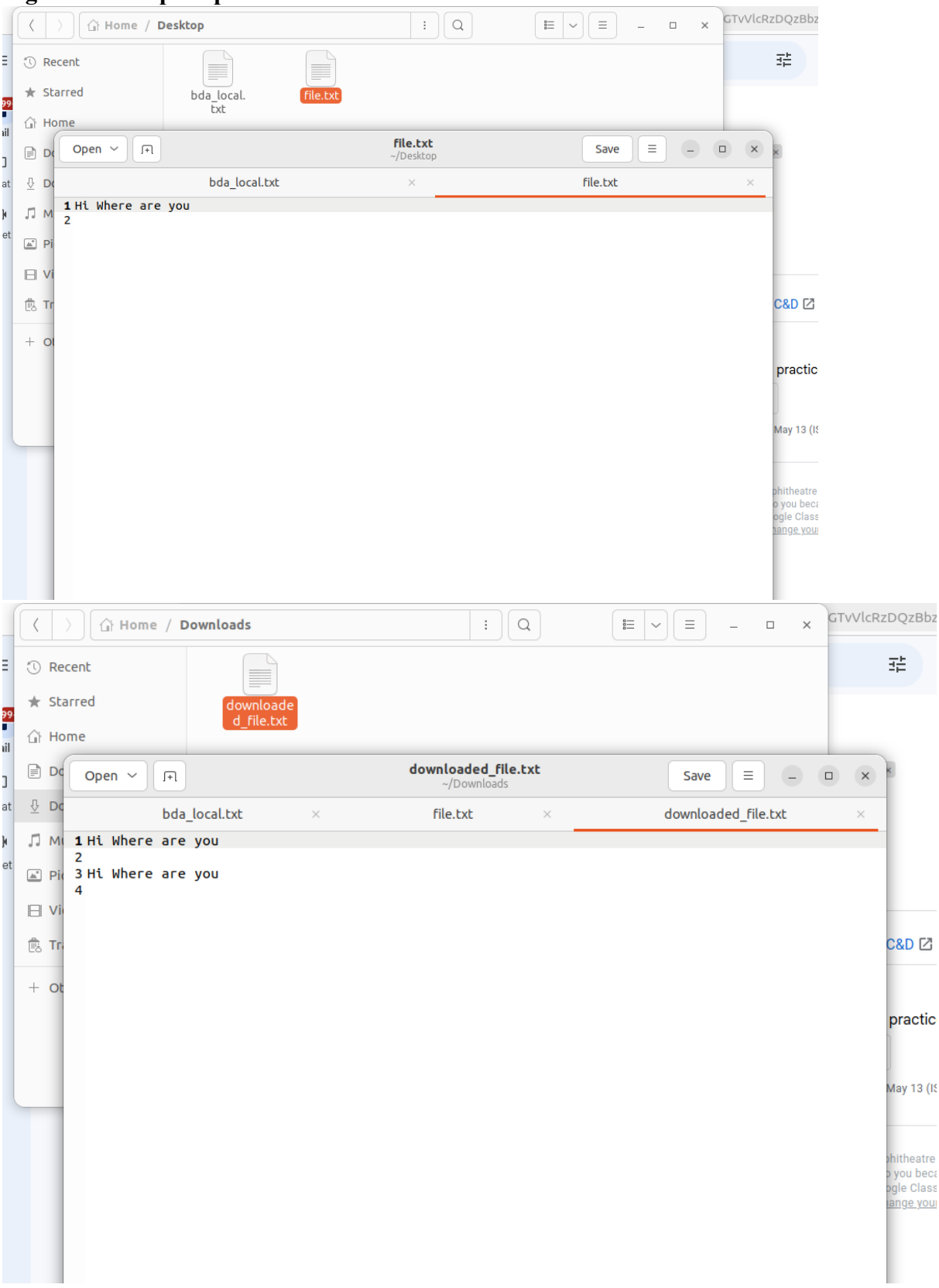


Fig.12: Hadoop output files:



7. Implement WordCount Program on Hadoop framework

Mapper Code:

```
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> {

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

String line = value.toString();
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();
```

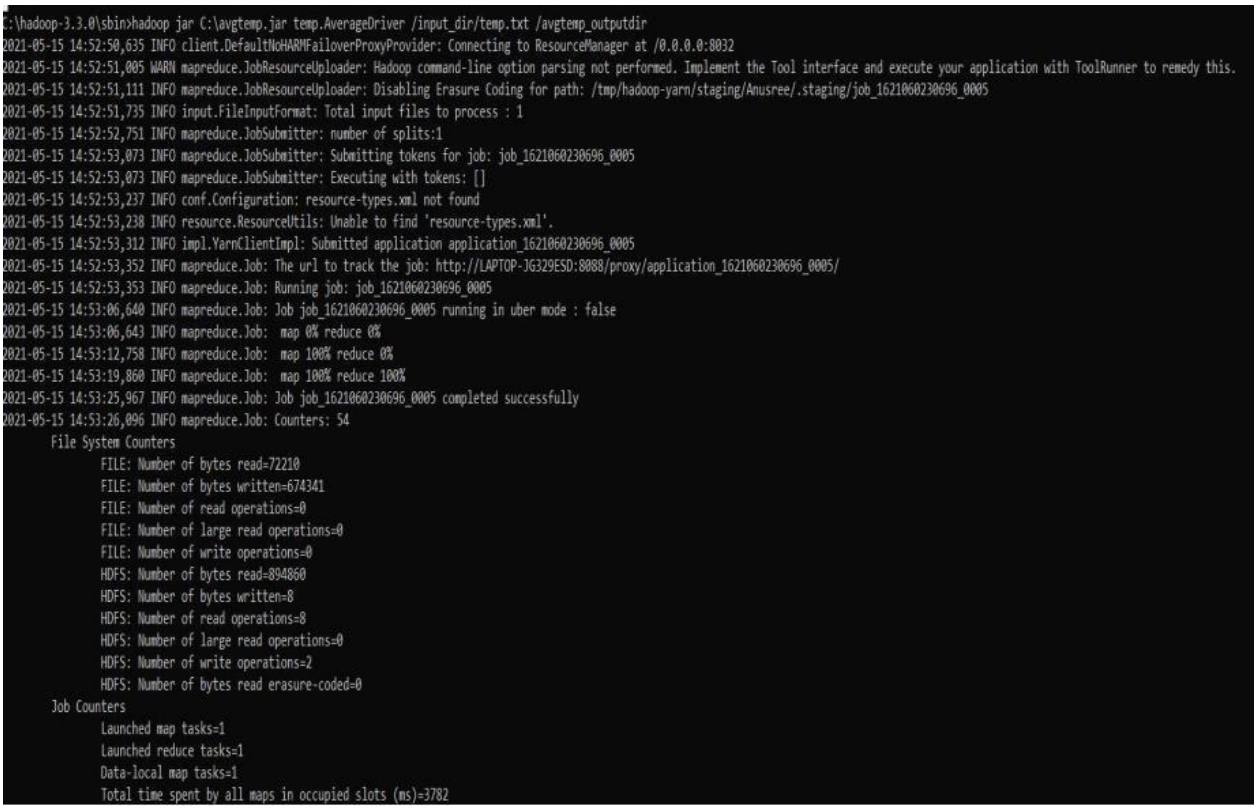
```
}

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

} }
```

Driver Code: You have to copy paste this program into the WCDriver Java Class file.

```
// 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);
  }
}
```

Fig.13: Output:



8.From the following link extract the weather data

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

Create a Map Reduce program to

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

Average Driver

```
package temp;

import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.IntWritable; import
org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class AverageDriver {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("Please Enter the input and output
parameters"); System.exit(-1);
}

Job job = new Job();
job.setJarByClass(AverageDriver.class);
job.setJobName("Max temperature");
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]);
job.setMapperClass(AverageMapper.class);
job.setReducerClass(AverageReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
  }
```

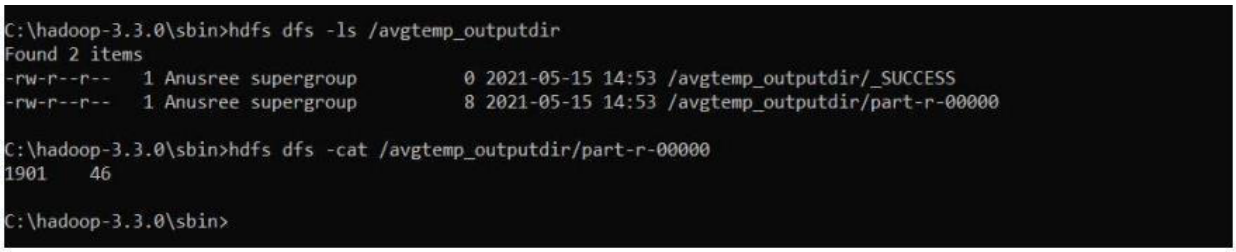
Average Mapper

```
package temp;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class AverageMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
int temperature;

String line = value.toString();
String year = line.substring(15,
19); if (line.charAt(87) == '+') {
temperature = Integer.parseInt(line.substring(88, 92));

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

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

if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(year), new IntWritable(temperature));
}

}
```

Average
Reducer:

```
package temp;
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class AverageReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int count = 0;
for (IntWritable value : values) {
max_temp += value.get();
count++;
}

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

}}
```

Fig.14: Program a) output:



b) Find the mean max temperature for every month

MeanMaxDriver.class:

```
package meanmax;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;

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

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

public class MeanMaxDriver {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("Please Enter the input and output
parameters"); System.exit(-1);
}

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

}
```

MeanMaxMapper.class

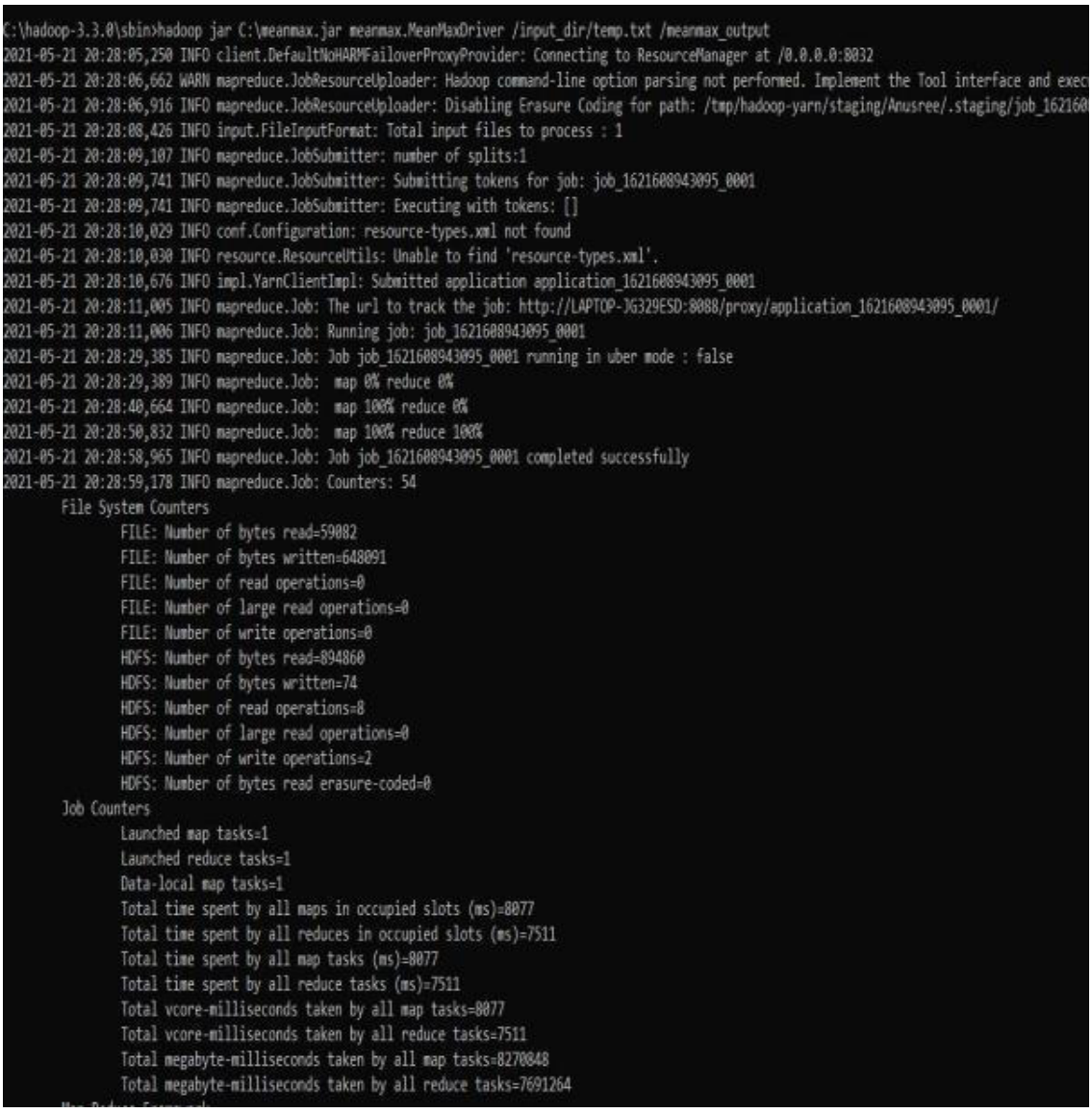
```
package meanmax;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.LongWritable
 import org.apache.hadoop.io.Text;

 import org.apache.hadoop.mapreduce.Mapper;

 public class MeanMaxMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

 public static final int MISSING = 9999;

 public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,

 IntWritable>.Context context) throws IOException, InterruptedException {

 int temperature;

 String line = value.toString();

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

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

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

 } else {

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

 }

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

 if (temperature != 9999 && quality.matches("[01459]"))

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

 }
```

MeanMaxReducer.class

```
package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class MeanMaxReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int total_temp = 0;
int count = 0;
int days = 0;
for (IntWritable value : values) {
int temp = value.get();
if (temp > max_temp)
max_temp = temp;
count++;
if (count == 3) {
total_temp += max_temp;
max_temp = 0;
count = 0;
days++;
}
```

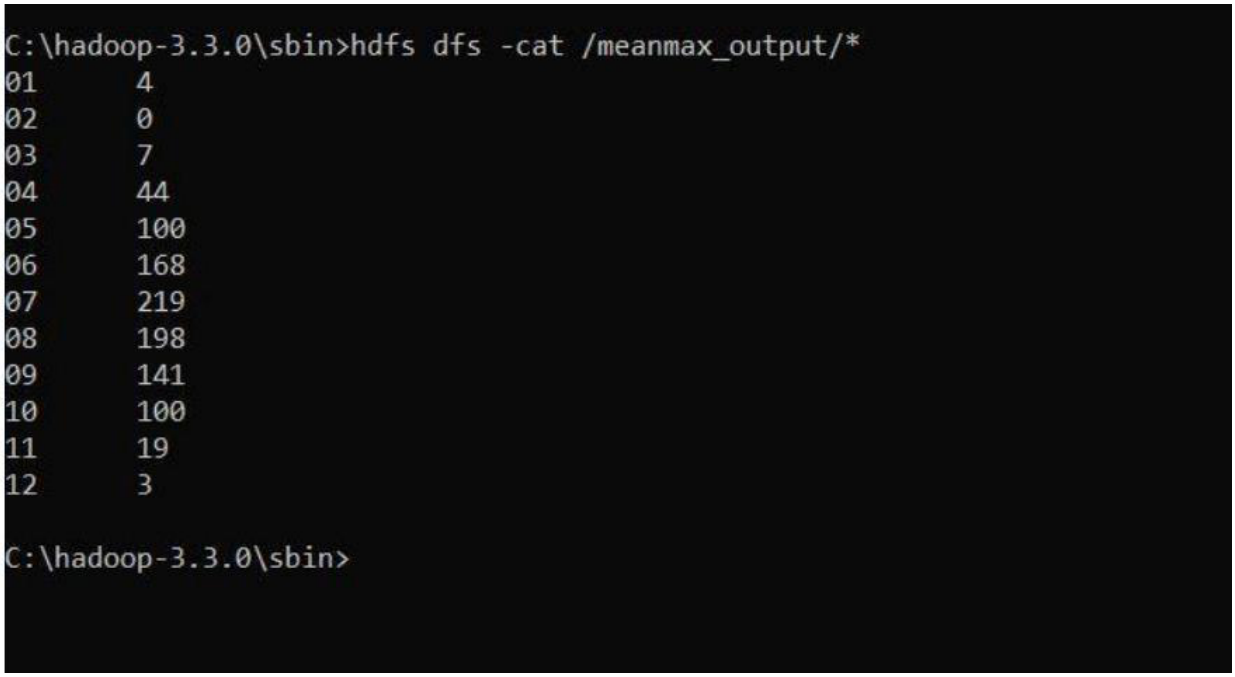
```
}
context.write(key, new IntWritable(total_temp / days));
}
}
```

Fig.15: Program b) Output:



```
C:\hadoop-3.3.0\sbin>hadoop jar C:\meanmax.jar meanmax.MeanMaxDriver /input_dir/temp.txt /meanmax_output
2021-05-21 20:28:05,250 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-21 20:28:06,662 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and exec
2021-05-21 20:28:06,916 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_162160
2021-05-21 20:28:08,426 INFO input.FileInputFormat: Total input files to process : 1
2021-05-21 20:28:09,107 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621608943095_0001
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-21 20:28:10,029 INFO conf.Configuration: resource-types.xml not found
2021-05-21 20:28:10,030 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-21 20:28:10,676 INFO impl.YarnClientImpl: Submitted application application_1621608943095_0001
2021-05-21 20:28:11,005 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621608943095_0001/
2021-05-21 20:28:11,006 INFO mapreduce.Job: Running job: job_1621608943095_0001
2021-05-21 20:28:29,385 INFO mapreduce.Job: Job job_1621608943095_0001 running in uber mode : false
2021-05-21 20:28:29,389 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-21 20:28:40,664 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-21 20:28:50,832 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-21 20:28:58,965 INFO mapreduce.Job: Job job_1621608943095_0001 completed successfully
2021-05-21 20:28:59,178 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=59082
                FILE: Number of bytes written=648091
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=894860
                HDFS: Number of bytes written=74
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=8077
                Total time spent by all reduces in occupied slots (ms)=7511
                Total time spent by all map tasks (ms)=8077
                Total time spent by all reduce tasks (ms)=7511
                Total vcore-milliseconds taken by all map tasks=8077
                Total vcore-milliseconds taken by all reduce tasks=7511
                Total megabyte-milliseconds taken by all map tasks=8270848
                Total megabyte-milliseconds taken by all reduce tasks=7691264
```

Fig.16: output continued:



9. For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words.

Driver-TopN.class

```
package samples.topn;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class TopN {
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
if (otherArgs.length != 2) {
System.err.println("Usage: TopN <in> <out>");
System.exit(2);
}
Job job = Job.getInstance(conf);
```

```
job.setJobName("Top N");
job.setJarByClass(TopN.class);
job.setMapperClass(TopNMapper.class);
job.setReducerClass(TopNReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
private static final IntWritable one = new IntWritable(1);
private Text word = new Text();
private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";
public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
StringTokenizer itr = new StringTokenizer(cleanLine);
while (itr.hasMoreTokens()) {
this.word.set(itr.nextToken().trim());
context.write(this.word, one);
}
}
}
```

```
}
```

TopNCombiner.class

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

TopNMapper.class

```
package samples.topn;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
```

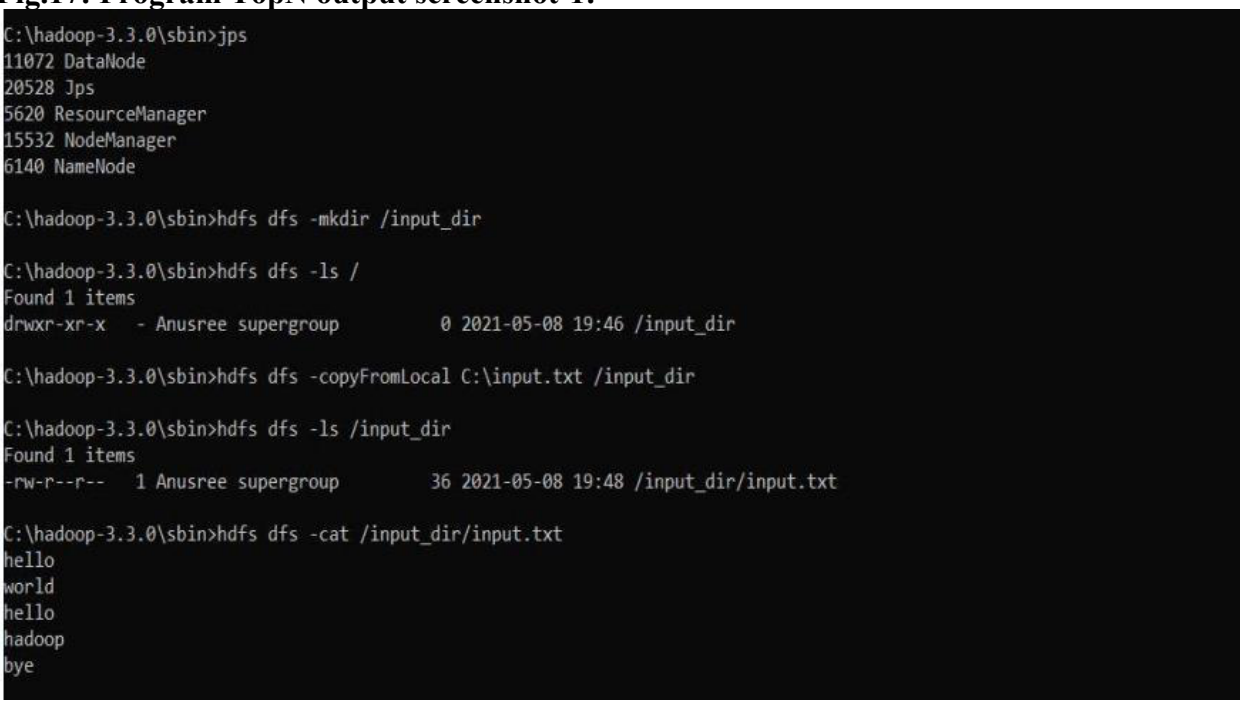
```
private static final IntWritable one = new IntWritable(1);
private Text word = new Text();
private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";
public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
StringTokenizer itr = new StringTokenizer(cleanLine);
while (itr.hasMoreTokens()) {
this.word.set(itr.nextToken().trim());
context.write(this.word, one);
}
}
}
```

TopNReducer.class

```
package samples.topn;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;
public class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
private Map<Text, IntWritable> countMap = new HashMap<>();
```

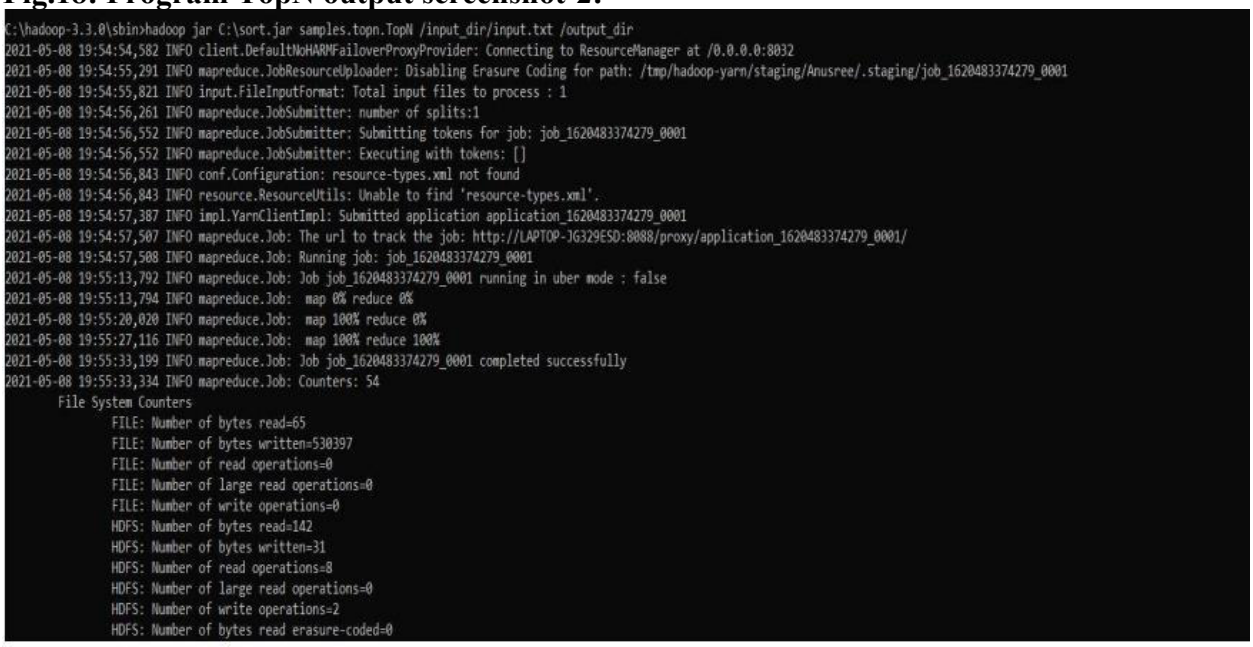
```
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values)
sum += val.get();
this.countMap.put(new Text(key), new IntWritable(sum));
}
protected void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

Fig.17: Program TopN output screenshot-1:



```
C:\hadoop-3.3.0\sbin>jps
11072 DataNode
20528 Jps
5620 ResourceManager
15532 NodeManager
6140 NameNode

C:\hadoop-3.3.0\sbin>hdfs dfs -mkdir /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - Anusree supergroup          0 2021-05-08 19:46 /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -copyFromLocal C:\input.txt /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /input_dir
Found 1 items
-rw-r--r--   1 Anusree supergroup         36 2021-05-08 19:48 /input_dir/input.txt

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/input.txt
hello
world
hello
hadoop
bye
```

Fig.18: Program TopN output screenshot-2:



```
C:\hadoop-3.3.0\sbin>hadoop jar C:\sort.jar samples.topn.TopN /input_dir/input.txt /output_dir
2021-05-08 19:54:54,582 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-08 19:54:55,291 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1620483374279_0001
2021-05-08 19:54:55,821 INFO input.FileInputFormat: Total input files to process : 1
2021-05-08 19:54:56,261 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1620483374279_0001
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-08 19:54:56,843 INFO conf.Configuration: resource-types.xml not found
2021-05-08 19:54:56,843 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-08 19:54:57,387 INFO impl.YarnClientImpl: Submitted application application_1620483374279_0001
2021-05-08 19:54:57,507 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1620483374279_0001/
2021-05-08 19:54:57,508 INFO mapreduce.Job: Running job: job_1620483374279_0001
2021-05-08 19:55:13,792 INFO mapreduce.Job: Job job_1620483374279_0001 running in uber mode : false
2021-05-08 19:55:13,794 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-08 19:55:20,020 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-08 19:55:27,116 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-08 19:55:33,199 INFO mapreduce.Job: Job job_1620483374279_0001 completed successfully
2021-05-08 19:55:33,334 INFO mapreduce.Job: Counters: 54
       File System Counters
              FILE: Number of bytes read=65
              FILE: Number of bytes written=530397
              FILE: Number of read operations=0
              FILE: Number of large read operations=0
              FILE: Number of write operations=0
              HDFS: Number of bytes read=142
              HDFS: Number of bytes written=31
              HDFS: Number of read operations=8
              HDFS: Number of large read operations=0
              HDFS: Number of write operations=2
              HDFS: Number of bytes read erasure-coded=0
```