VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS

Submitted by

Sharan S Pai (1BM19CS146)

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

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

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled “BIG DATA ANALYTICS” carried out by Sharan S Pai (1BM19CS146), who is bonafide student of B. M. S. College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a BIG DATA ANALYTICS - (20CS6PEBDA) work prescribed for the said degree.

Antara Roy Choudhary
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

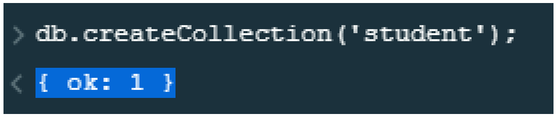
`

1. MongoDB- CRUD Demonstration

Mongo DB Expts

1. Create a collection called students

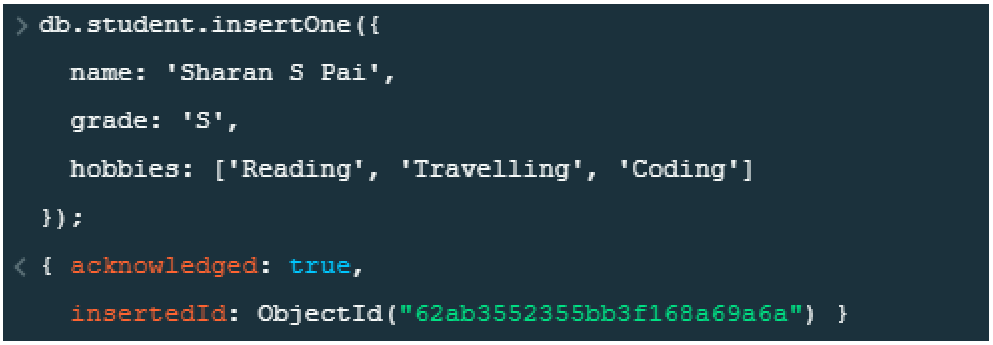
>db.createCollection('student');



```
> db.createCollection('student');
< { ok: 1 }
```

2. Insert an element into the collection which has name, grade, hobbies

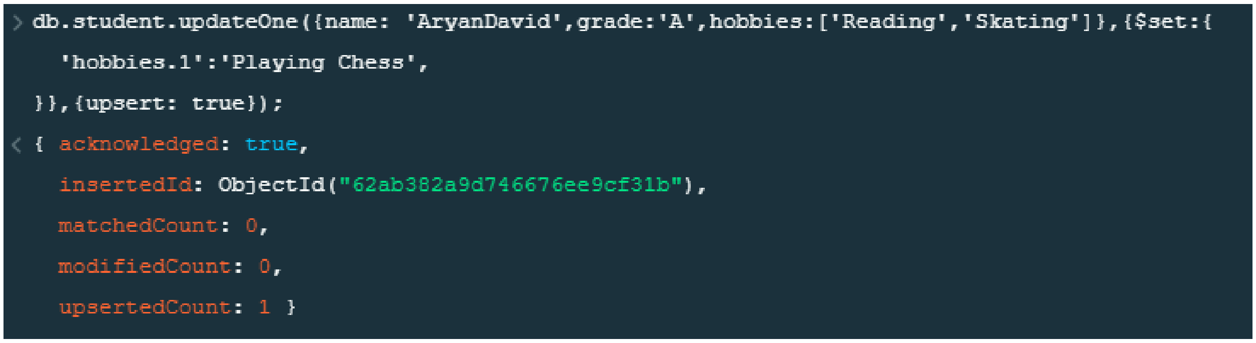
```
>db.student.insertOne({
  name: 'Sharan S Pai',
  grade: 'S',
  hobbies: ['Reading', 'Travelling', 'Coding']
});
```



```
> db.student.insertOne({
    name: 'Sharan S Pai',
    grade: 'S',
    hobbies: ['Reading', 'Travelling', 'Coding']
  });
< { acknowledged: true,
    insertedId: ObjectId("62ab3552355bb3f168a69a6a") }
```

3. 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”.

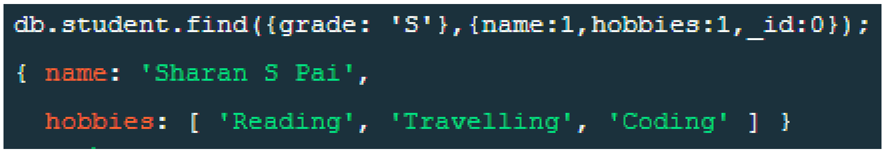
```
   >db.student.updateOne({name: 'AryanDavid',grade:'A',hobbies:['Reading','Skating']}
   ,{$set:{
 'hobbies.1':'Playing Chess',
}},{upsert: true});
```



```
> db.student.updateOne({name: 'AryanDavid',grade:'A',hobbies:['Reading','Skating']},{$set:{
    'hobbies.1':'Playing Chess',
  }},{upsert: true});
< { acknowledged: true,
    insertedId: ObjectId("62ab382a9d746676ee9cf31b"),
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 1 }
```

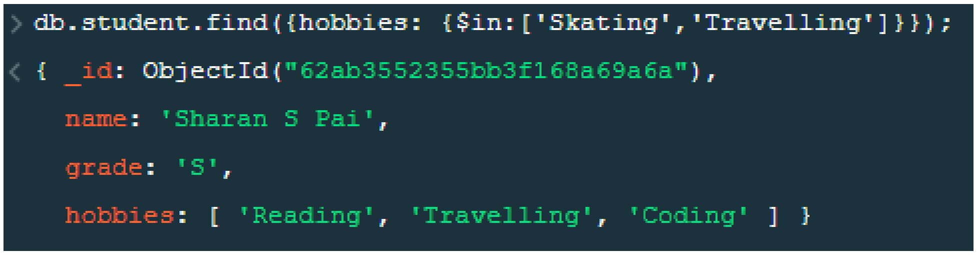
4. Find collection based on some search criteria

>db.student.find({grade: 'S'},{name:1,hobbies:1,_id:0});



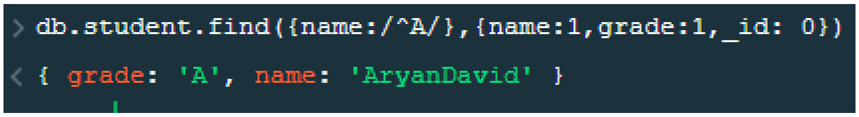
```
db.student.find({grade: 'S'},{name:1,hobbies:1,_id:0});
{ name: 'Sharan S Pai',
  hobbies: [ 'Reading', 'Travelling', 'Coding' ] }
```

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



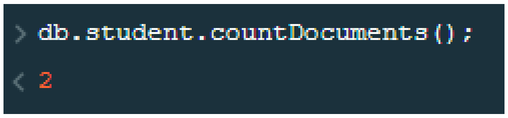
```
> db.student.find({hobbies: {$in:['Skating','Travelling']}});
< { _id: ObjectId("62ab3552355bb3f168a69a6a"),
    name: 'Sharan S Pai',
    grade: 'S',
    hobbies: [ 'Reading', 'Travelling', 'Coding' ] }
```

6. To find documents from the Students collection where the StudName begins with “A”.



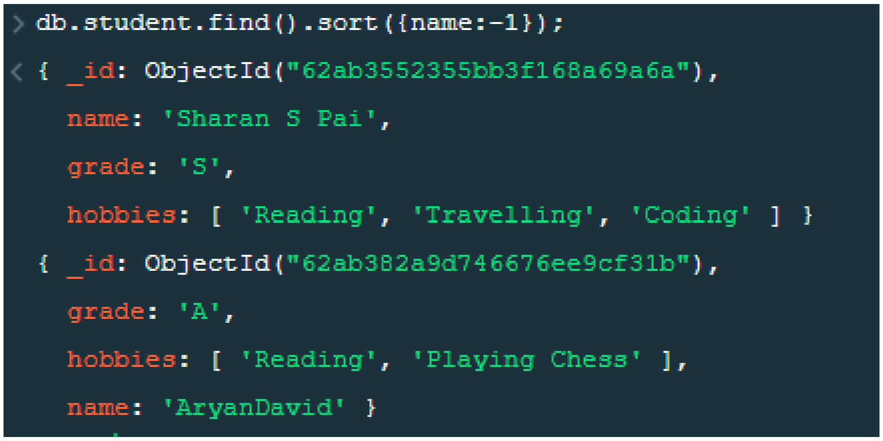
```
> db.student.find({name:/^A/},{name:1,grade:1,_id: 0})
< { grade: 'A', name: 'AryanDavid' }
```

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



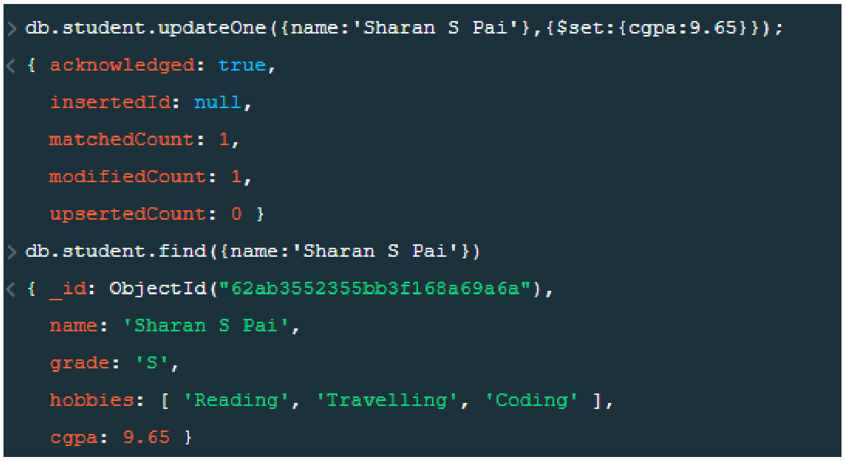
```
> db.student.countDocuments();
< 2
```

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



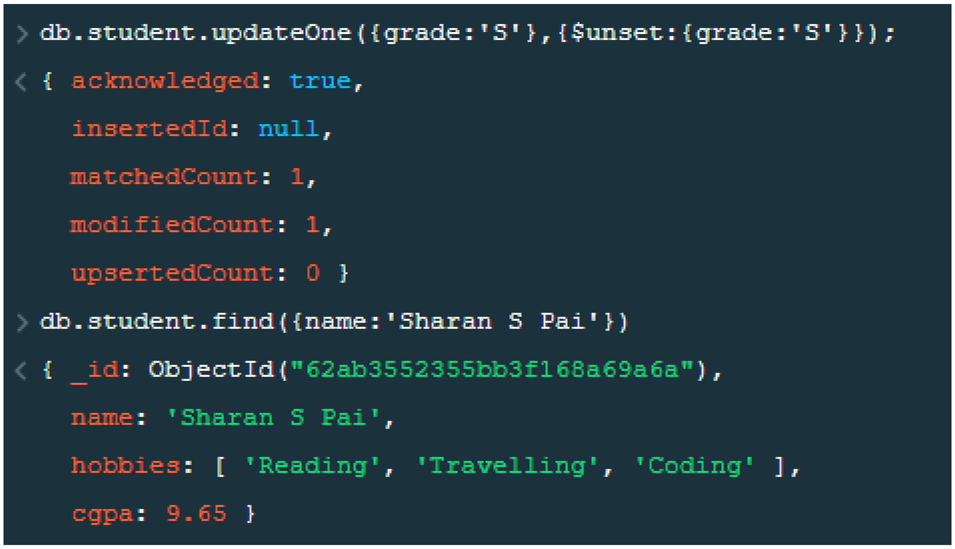
```
> db.student.find().sort({name:-1});
< { _id: ObjectId("62ab3552355bb3f168a69a6a"),
    name: 'Sharan S Pai',
    grade: 'S',
    hobbies: [ 'Reading', 'Travelling', 'Coding' ] }
  { _id: ObjectId("62ab382a9d746676ee9cf31b"),
    grade: 'A',
    hobbies: [ 'Reading', 'Playing Chess' ],
    name: 'AryanDavid' }
```

9. Add a new field to existing document



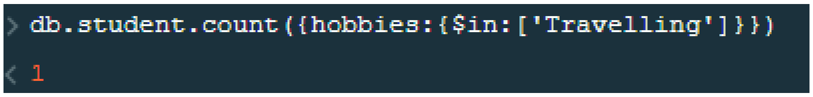
```
> db.student.updateOne({name:'Sharan S Pai'},{$set:{cgpa:9.65}});
< { acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0 }
> db.student.find({name:'Sharan S Pai'})
< { _id: ObjectId("62ab3552355bb3f168a69a6a"),
    name: 'Sharan S Pai',
    grade: 'S',
    hobbies: [ 'Reading', 'Travelling', 'Coding' ],
    cgpa: 9.65 }
```

10. Remove a field from existing document



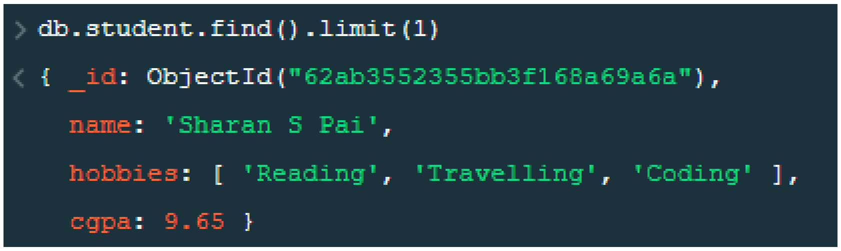
```
> db.student.updateOne({grade:'S'},{$unset:{grade:'S'}});
< { acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0 }
> db.student.find({name:'Sharan S Pai'})
< { _id: ObjectId("62ab3552355bb3f168a69a6a"),
    name: 'Sharan S Pai',
    hobbies: [ 'Reading', 'Travelling', 'Coding' ],
    cgpa: 9.65 }
```

11. Count the document that have travelling as their hobby



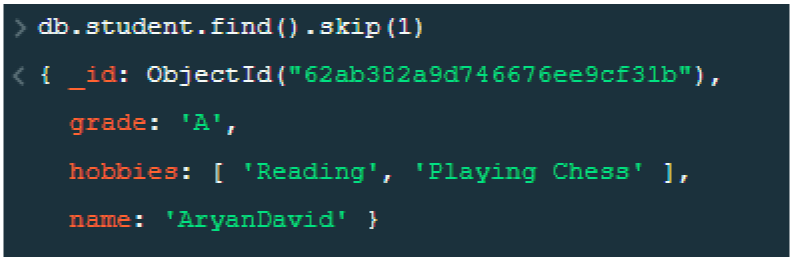
```
> db.student.count({hobbies:{$in:['Travelling']}})
< 1
```

12. Read only 1st document



```
> db.student.find().limit(1)
< { _id: ObjectId("62ab3552355bb3f168a69a6a"),
    name: 'Sharan S Pai',
    hobbies: [ 'Reading', 'Travelling', 'Coding' ],
    cgpa: 9.65 }
```

13. Read all documents by skipping over 1st document



```
> db.student.find().skip(1)
< { _id: ObjectId("62ab382a9d746676ee9cf31b"),
    grade: 'A',
    hobbies: [ 'Reading', 'Playing Chess' ],
    name: 'AryanDavid' }
```

2. Perform the following DB operations using Cassandra.

(Employee DB)

1. Create keyspace

```
CREATE KEYSPACE employee;
```

2. create employee table

```
create table employee_in( emp_id int, emp_name text, desig text, dateofjoin date, salary float, dept text,
PRIMARY KEY(emp_id,salary));
```

3. Insert data interms of batches

```
BEGIN BATCH
        ... INSERT INTO employee_in(emp_id,emp_name,desig,dateofjoin,salary,dept) VALUES (1,'Karan
        ... ','Manager','2021-06-12',1000000,'HR')
        ... INSERT INTO employee_in(emp_id,emp_name,desig,dateofjoin,salary,dept) VALUES (2,'Rajesh
Meheta','associate SE','2020-04-20',600000,'Tech')
        ... INSERT INTO employee_in(emp_id,emp_name,desig,dateofjoin,salary,dept) VALUES (3,'Vihnu
Chauhan','associate SE','2020-05-20',600000,'Tech')
        ... INSERT INTO employee_in(emp_id,emp_name,desig,dateofjoin,salary,dept) VALUES (4,'Shweta
Tripathi','associate SE','2020-05-20',600000,'Tech')
        ... APPLY BATCH
        ... ;
```

4. Update one emp_name and dept in the table

```
select * from employee_in
        ... ;

 emp_id | salary | dateofjoin | dept | desig        | emp_name
--------+--------+------------+------+--------------+-----------------
      1 |  1e+06 | 2021-06-12 |   HR |      Manager |         Karan\n
      2 |  6e+05 | 2020-04-20 | Tech | associate SE |   Rajesh Meheta
      4 |  6e+05 | 2020-05-20 | Tech | associate SE | Shweta Tripathi
      3 |  6e+05 | 2020-05-20 | Tech | associate SE |   Vihnu Chauhan
```

```
update employee_in SET emp_name='Vishnu Chauhan',dept='Technical' where emp_id=3 AND
salary=600000;
```

```
select * from employee_in ;

 emp_id | salary | dateofjoin | dept      | desig        | emp_name
--------+--------+------------+-----------+--------------+-----------------
      1 |  1e+06 | 2021-06-12 |        HR |      Manager |         Karan\n
      2 |  6e+05 | 2020-04-20 |      Tech | associate SE |   Rajesh Meheta
      4 |  6e+05 | 2020-05-20 |      Tech | associate SE | Shweta Tripathi
      3 |  6e+05 | 2020-05-20 | Technical | associate SE |  Vishnu Chauhan
```

5. SORT the entire employee table on Salary

```
select * from employee_in where emp_id IN(1,2,3,4) ORDER BY salary DESC allow filtering;

 emp_id | salary | dateofjoin | dept      | desig        | emp_name
--------+--------+------------+-----------+--------------+-----------------
      1 |  1e+06 | 2021-06-12 |        HR |      Manager |         Karan\n
      2 |  6e+05 | 2020-04-20 |      Tech | associate SE |   Rajesh Meheta
      3 |  6e+05 | 2020-05-20 | Technical | associate SE |  Vishnu Chauhan
      4 |  6e+05 | 2020-05-20 |      Tech | associate SE | Shweta Tripathi
```

6. Add projects column to the table

```
ALTER TABLE employee_in ADD projects list<text>;

select * from employee_in ;

 emp_id | salary | dateofjoin | dept      | desig        | emp_name        | projects
--------+--------+------------+-----------+--------------+-----------------+----------
      1 |  1e+06 | 2021-06-12 |        HR |      Manager |         Karan\n |     null
      2 |  6e+05 | 2020-04-20 |      Tech | associate SE |   Rajesh Meheta |     null
      4 |  6e+05 | 2020-05-20 |      Tech | associate SE | Shweta Tripathi |     null
      3 |  6e+05 | 2020-05-20 | Technical | associate SE |  Vishnu Chauhan |     null
```

7. update the projects in the table

```
>update employee_in SET projects=['CCF','CCD','KMAP'] where emp_id=3 AND salary=600000;
>update employee_in SET projects=['AAP','BJP','TMC'] where emp_id=4 AND salary=600000;
>select * from employee_in ;

 emp_id | salary | dateofjoin | dept      | desig        | emp_name        | projects
--------+--------+------------+-----------+--------------+-----------------+------------------------
      1 |  1e+06 | 2021-06-12 |        HR |      Manager |         Karan\n |                   null
      2 |  6e+05 | 2020-04-20 |      Tech | associate SE |   Rajesh Meheta |                   null
      4 |  6e+05 | 2020-05-20 |      Tech | associate SE | Shweta Tripathi |  ['AAP', 'BJP', 'TMC']
      3 |  6e+05 | 2020-05-20 | Technical | associate SE |  Vishnu Chauhan | ['CCF', 'CCD', 'KMAP']
```

8. CREATE TTL of 15 seconds to display values of employees

```
>update employee_in USING TTL 15  SET emp_name='Karan Sharma' where emp_id=1 AND
salary=600000;
//BEFORE 15 seconds
> select * from employee_in ;

 emp_id | salary | dateofjoin | dept      | desig        | emp_name        | projects
--------+--------+------------+-----------+--------------+-----------------+------------------------
      1 |  6e+05 |       null |      null |         null |    Karan Sharma |                   null
      1 |  1e+06 | 2021-06-12 |        HR |      Manager |         Karan\n |                   null
      2 |  6e+05 | 2020-04-20 |      Tech | associate SE |   Rajesh Meheta |                   null
      4 |  6e+05 | 2020-05-20 |      Tech | associate SE | Shweta Tripathi |  ['AAP', 'BJP', 'TMC']
      3 |  6e+05 | 2020-05-20 | Technical | associate SE |  Vishnu Chauhan | ['CCF', 'CCD', 'KMAP']
```

```
//AFTER 15 seconds
> select * from employee_in ;

 emp_id | salary | dateofjoin | dept      | desig        | emp_name        | projects
--------+--------+------------+-----------+--------------+-----------------+------------------------
      1 |  1e+06 | 2021-06-12 |        HR |      Manager |         Karan\n |                   null
      2 |  6e+05 | 2020-04-20 |      Tech | associate SE |   Rajesh Meheta |                   null
      4 |  6e+05 | 2020-05-20 |      Tech | associate SE | Shweta Tripathi |  ['AAP', 'BJP', 'TMC']
      3 |  6e+05 | 2020-05-20 | Technical | associate SE |  Vishnu Chauhan | ['CCF', 'CCD', 'KMAP']

(4 rows)
```

3. Perform the following DB operations usingCassandra.

(Library DB)

Perform following operation using CASSANDRA on library database

1. Create a keyspace library

```
> CREATE KEYSPACE library WITH REPLICATION={
        ‘class’: ‘SimpleStrategy’,
        ‘replication_factor’: 3
};
```

2. Create a column family by name lib_info:

```
> CREATE TABLE lib_info(
        sid int,
        c_val counter,
        sname text,
        bname text,
        bid int,
        doi date,
        PRIMARY KEY(sid,sname,bname,bid,doi)
);
```

```
> DESCRIBE lib_info;
```

OUTPUT:

```
CREATE TABLE library.lib_info (
    sid int,
    sname text,
    bname text,
    bid int,
    doi date,
    c_val counter,
    PRIMARY KEY (sid, sname, bname, bid, doi)
) WITH CLUSTERING ORDER BY (sname ASC, bname ASC, bid ASC, doi ASC)
    AND 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 crc_check_chance = 1.0
    AND default_time_to_live = 0;
```

3) Insert values in batches:

Ans. Not possible to insert (update doesn’t work with batches)

```
ERROR: InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot include
a counter statement in a logged batch"
```

```
>UPDATE lib_info SET c_val=c_val+1 WHERE sid=110 and sname=’Sharan’ AND
bname=’BDA’ AND bid=120 AND doi=’2022-01-27’ ;

> UPDATE lib_info SET c_val=c_val+1 WHERE sid=112 and sname=’Varun’ AND
bname=’CNS’ AND bid=110 AND doi=’2022-01-27’ ;

> UPDATE lib_info SET c_val=c_val+1 WHERE sid=112 and sname=’Varun’ AND
bname=’CNS’ AND bid=110 AND doi=’2022-01-27’ ;

> SELECT * FROM lib_info;

sid | sname  | bname | bid | doi        | c_val
-----+--------+-------+-----+------------+-------
 110 | Sharan |   BDA | 120 | 2022-01-27 |     1
 112 |  Varun |   CNS | 110 | 2022-01-27 |     2
```

4. Display details of table and increase counter

```
> SELECT * FROM lib_info;

sid | sname  | bname | bid | doi        | c_val
-----+--------+-------+-----+------------+-------
 110 | Sharan |   BDA | 120 | 2022-01-27 |     1
 112 |  Varun |   CNS | 110 | 2022-01-27 |     2

> UPDATE lib_info SET c_val=c_val+1 WHERE sid=112 and sname=’Varun’ AND
bname=’CNS’ AND bid=110 AND doi=’2022-01-27’ ;

> SELECT * FROM lib_info;

sid | sname  | bname | bid | doi        | c_val
-----+--------+-------+-----+------------+-------
 110 | Sharan |   BDA | 120 | 2022-01-27 |     1
 112 |  Varun |   CNS | 110 | 2022-01-27 |     3
```

5. Write a query to show student with id 112 has taken CNS 2 times

```
> SELECT * FROM lib_info WHERE sid=112 AND c_val>=2 AND sname='Karan' AND
bname='CNS' AND bid=110 AND doi='2022-01-27' ALLOW FILTERING;

sid | sname  | bname | bid | doi        | c_val
-----+--------+-------+-----+------------+-------
```

```
 112 |  Varun |    CNS | 110 | 2022-01-27 |     3
```

6. Export created column family to csv file

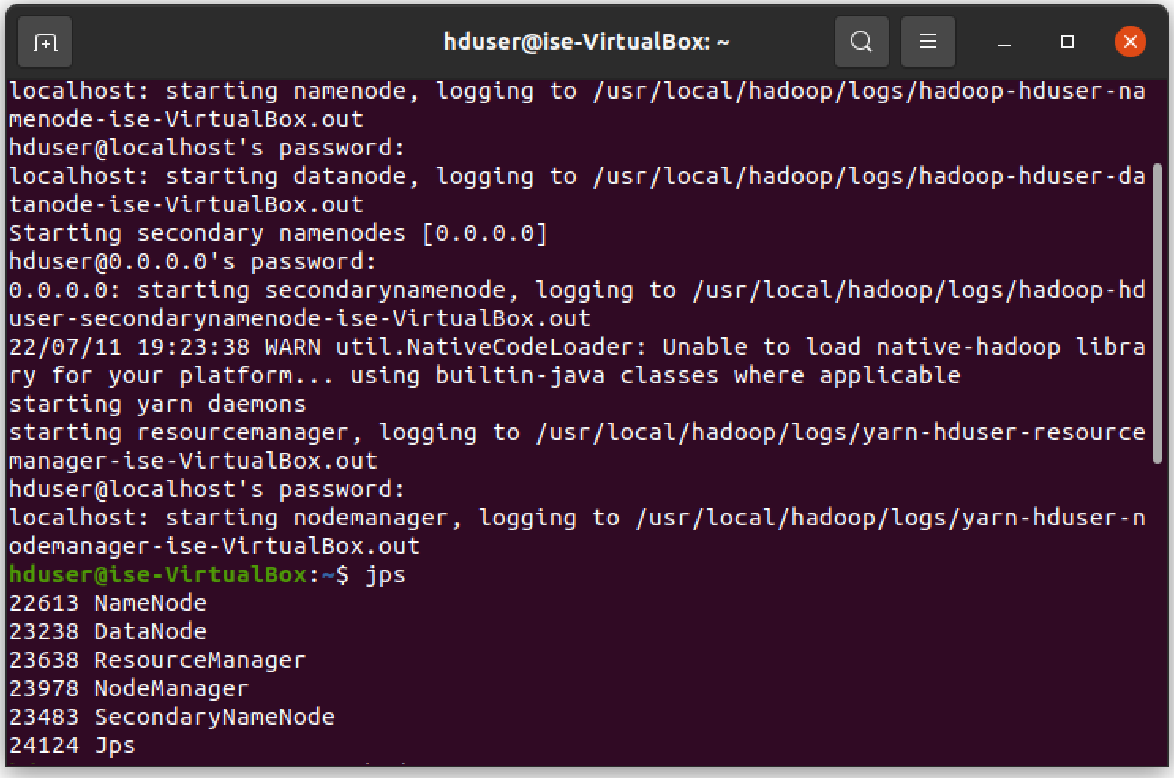
```
> COPY lib_info(sid,sname,c_val,bid,bname,doi) TO ‘./lib.csv’;

Using 1 child processes

Starting copy of library.lib_info with columns [sid, sname, c_val, bid, bname, doi].
Processed: 2 rows; Rate:      11 rows/s; Avg. rate:       3 rows/s
2 rows exported to 1 files in 0.827 seconds.
```

7. Import csv dataset from local FS to cassandra

```
> COPY lib_info(sid,sname,c_val,bid,bname,doi) FROM ‘./lib.csv’;
Using 1 child processes

Starting copy of library.lib_info with columns [sid, sname, c_val, bid, bname, doi].
Processed: 2 rows; Rate:       1 rows/s; Avg. rate:       1 rows/s
2 rows imported from 1 files in 0 day, 0 hour, 0 minute, and 1.400 seconds (0 skipped).

> SELECT * FROM LIB_INFO;
sid | sname  | bname | bid | doi        | c_val
-----+--------+-------+-----+------------+-------
 110 | Sharan |   BDA | 120 | 2022-01-27 |     2
 112 |  Varun |    CNS | 110 | 2022-01-27 |     6
```

LAB-4

4. Screenshot of Hadoop installed



```
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-na
menode-ise-VirtualBox.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-da
tanode-ise-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hd
user-secondarynamenode-ise-VirtualBox.out
22/07/11 19:23:38 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resource
manager-ise-VirtualBox.out
hduser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-n
odemanager-ise-VirtualBox.out
hduser@ise-VirtualBox:~$ jps
22613 NameNode
23238 DataNode
23638 ResourceManager
23978 NodeManager
23483 SecondaryNameNode
24124 Jps
```

LAB-5

5 Execution of HDFS Commands for interaction with Hadoop Environment.

```
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

4644 NameNode

5450 SecondaryNameNode

6666 NodeManager

4827 DataNode

5710 ResourceManager

6799 Jps

hduser@bmsce-Precision-T1700:~$ ls

 b                 'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz'

 c                  Pictures

 derby.log          pig_1564816082257.log

 Desktop            pt

 Documents          PT72Installer

 Downloads          Public

 eclipse-workspace  R

 examples.desktop   snap

 hadoop-2.6.0.tar.gz  Templates
 hive              toinstalledlist
 metastore_db       Videos
 Music
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /
Found 2 items
drwxrwxr-x   - hduser supergroup          0 2019-08-01 16:19 /tmp
drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /abc
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x   - hduser supergroup          0 2022-05-31 09:38 /abc
drwxrwxr-x   - hduser supergroup          0 2019-08-01 16:19 /tmp
drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user
hduser@bmsce-Precision-T1700:~$ hdfs dfs -touchz /abc/lab.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /abc
Found 1 items
-rw-r--r--   1 hduser supergroup          0 2022-05-31 09:39 /abc/lab.txt
hduser@bmsce-Precision-T1700:~$ ls
 b                 'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz'
 c                  Pictures
 derby.log          pig_1564816082257.log
 Desktop            pt
 Documents          PT72Installer
 Downloads          Public
 eclipse-workspace  R
 examples.desktop   snap
 hadoop-2.6.0.tar.gz Templates
 hive               toinstalledlist
 metastore_db       Videos
 Music
hduser@bmsce-Precision-T1700:~$ vi new.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put new.txt /abc/newhadoop.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/newhadoop.txt
Cbbbbb
fgggjyujyhcvdgrbghh
```

```
hduser@bmsce-Precision-T1700:~$ cd /Desktop

bash: cd: /Desktop: No such file or directory

hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /

Found 3 items

drwxr-xr-x   - hduser supergroup          0 2022-05-31 09:48 /abc

drwxrwxr-x   - hduser supergroup          0 2019-08-01 16:19 /tmp

drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/Desktop/Welcome.txt /abc/newWelcome.txt

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

nnkjkdngdmglc

hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/wc.txt /home/hduser/Downloads/wcc.txt

get: `/abc/wc.txt': No such file or directory

hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/newWelcome.txt /home/hduser/Downloads/wcc.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /abc/newWelcome.txt /home/hduser/Downloads

hduser@bmsce-Precision-T1700:~$ hadoop fs -mv /abc /FFF

hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /

Found 3 items

drwxr-xr-x   - hduser supergroup          0 2022-05-31 10:08 /FFF

drwxrwxr-x   - hduser supergroup          0 2019-08-01 16:19 /tmp

drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user

hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /FFF/new.txt /tmp

cp: `/FFF/new.txt': No such file or directory

hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /FFF

Found 3 items

-rw-r--r--   1 hduser supergroup          0 2022-05-31 09:39 /FFF/lab.txt

-rw-r--r--   1 hduser supergroup         14 2022-05-31 10:08 /FFF/newWelcome.txt

-rw-r--r--   1 hduser supergroup         27 2022-05-31 09:48 /FFF/newhadoop.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /FFF/lab.txt /tmp

hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /tmp

Found 2 items

drwx-wx-wx   - hduser supergroup          0 2019-08-01 16:19 /tmp/hive

-rw-r--r--   1 hduser supergroup          0 2022-05-31 10:19 /tmp/lab.txt

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

LAB-6

6. From the following link extract the weather data https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all. Create a Map Reduce program to

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

- **Program**

```
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(&quot;Please Enter the input and output
parameters&quot;);
System.exit(-1);
}
Job job = new Job();
job.setJarByClass(AverageDriver.class);
job.setJobName(&quot;Max temperature&quot;);
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&lt;LongWritable, Text,
Text, IntWritable&gt; {
```

```
public static final int MISSING = 9999;
public void map(LongWritable key, Text value,
Mapper&lt;LongWritable, Text, Text, IntWritable&gt;.Context context)
throws IOException, InterruptedException {
int temperature;
String line = value.toString();
String year = line.substring(15, 19);
if (line.charAt(87) == &#39;+&#39;) {
temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 &amp;&amp; quality.matches(&quot;[01459]&quot;))
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&lt;Text, IntWritable,
Text, IntWritable&gt; {
public void reduce(Text key, Iterable&lt;IntWritable&gt; values,
Reducer&lt;Text, IntWritable, Text, IntWritable&gt;.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:~$ sudo su hduser
[sudo] password for hduser:
```

```
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
7376 DataNode
8212 Jps
8090 NodeManager
3725 org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar
7758 ResourceManager
7199 NameNode
7599 SecondaryNameNode
hduser@bmsce-Precision-T1700:~$ hadoop fs -mkdir /input_kundana
hduser@bmsce-Precision-T1700:~$ hadoop fs -put Downloads/1901 /input_kundana/1901.txt
hduser@bmsce-Precision-T1700:~$ hadoop jar Desktop/temp.jar Temperature.AverageDriver
/input_kundana/1901.txt /output_1901
Exception in thread "main" java.lang.ClassNotFoundException: Temperature.AverageDriver
        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:~$ hadoop jar Desktop/temp.jar AverageDriver
/input_kundana/1901.txt /output_1901
22/06/21 10:26:05 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/21 10:26:05 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
22/06/21 10:26:05 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing
not performed. Implement the Tool interface and execute your application with ToolRunner
to remedy this.
22/06/21 10:26:05 INFO input.FileInputFormat: Total input paths to process : 1
22/06/21 10:26:05 INFO mapreduce.JobSubmitter: number of splits:1
22/06/21 10:26:05 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_local1195965365_0001
22/06/21 10:26:05 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/21 10:26:05 INFO mapreduce.Job: Running job: job_local1195965365_0001
22/06/21 10:26:05 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/21 10:26:05 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/21 10:26:05 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/21 10:26:05 INFO mapred.LocalJobRunner: Starting task:
attempt_local1195965365_0001_m_000000_0
22/06/21 10:26:05 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/21 10:26:05 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/input_kundana/1901.txt:0+888190
22/06/21 10:26:06 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/21 10:26:06 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/21 10:26:06 INFO mapred.MapTask: soft limit at 83886080
22/06/21 10:26:06 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/21 10:26:06 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/21 10:26:06 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/21 10:26:06 INFO mapred.LocalJobRunner:
22/06/21 10:26:06 INFO mapred.MapTask: Starting flush of map output
22/06/21 10:26:06 INFO mapred.MapTask: Spilling map output
22/06/21 10:26:06 INFO mapred.MapTask: bufstart = 0; bufend = 59076; bufvoid = 104857600
22/06/21 10:26:06 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend =
26188144(104752576); length = 26253/6553600
22/06/21 10:26:06 INFO mapred.MapTask: Finished spill 0
22/06/21 10:26:06 INFO mapred.Task: Task:attempt_local1195965365_0001_m_000000_0 is
done. And is in the process of committing
22/06/21 10:26:06 INFO mapred.LocalJobRunner: map
22/06/21 10:26:06 INFO mapred.Task: Task 'attempt_local1195965365_0001_m_000000_0'
done.
22/06/21 10:26:06 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1195965365_0001_m_000000_0
22/06/21 10:26:06 INFO mapred.LocalJobRunner: map task executor complete.
22/06/21 10:26:06 INFO mapred.LocalJobRunner: Waiting for reduce tasks
22/06/21 10:26:06 INFO mapred.LocalJobRunner: Starting task:
attempt_local1195965365_0001_r_000000_0
22/06/21 10:26:06 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
```

```
22/06/21 10:26:06 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@65367f35
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: MergerManager:
memoryLimit=349752512, maxSingleShuffleLimit=87438128, mergeThreshold=230836672,
ioSortFactor=10, memToMemMergeOutputsThreshold=10
22/06/21 10:26:06 INFO reduce.EventFetcher: attempt_local1195965365_0001_r_000000_0
Thread started: EventFetcher for fetching Map Completion Events
22/06/21 10:26:06 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map
attempt_local1195965365_0001_m_000000_0 decomp: 72206 len: 72210 to MEMORY
22/06/21 10:26:06 INFO reduce.InMemoryMapOutput: Read 72206 bytes from map-output
for attempt_local1195965365_0001_m_000000_0
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of
size: 72206, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->72206
22/06/21 10:26:06 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning
22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory
map-outputs and 0 on-disk map-outputs
22/06/21 10:26:06 INFO mapred.Merger: Merging 1 sorted segments
22/06/21 10:26:06 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of
total size: 72199 bytes
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merged 1 segments, 72206 bytes to disk
to satisfy reduce memory limit
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 1 files, 72210 bytes from disk
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from
memory into reduce
22/06/21 10:26:06 INFO mapred.Merger: Merging 1 sorted segments
22/06/21 10:26:06 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of
total size: 72199 bytes
22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:26:06 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead,
use mapreduce.job.skiprecords
22/06/21 10:26:06 INFO mapred.Task: Task:attempt_local1195965365_0001_r_000000_0 is
done. And is in the process of committing
22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:26:06 INFO mapred.Task: Task attempt_local1195965365_0001_r_000000_0 is
allowed to commit now
22/06/21 10:26:06 INFO output.FileOutputCommitter: Saved output of task
'attempt_local1195965365_0001_r_000000_0' to
hdfs://localhost:54310/output_1901/_temporary/0/task_local1195965365_0001_r_000000
22/06/21 10:26:06 INFO mapred.LocalJobRunner: reduce > reduce
22/06/21 10:26:06 INFO mapred.Task: Task 'attempt_local1195965365_0001_r_000000_0'
done.
22/06/21 10:26:06 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1195965365_0001_r_000000_0
22/06/21 10:26:06 INFO mapred.LocalJobRunner: reduce task executor complete.
```

```
22/06/21 10:26:06 INFO mapreduce.Job: Job job_local1195965365_0001 running in uber
mode : false
22/06/21 10:26:06 INFO mapreduce.Job:  map 100% reduce 100%
22/06/21 10:26:06 INFO mapreduce.Job: Job job_local1195965365_0001 completed
successfully
22/06/21 10:26:06 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=152940
                FILE: Number of bytes written=725372
                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=110
                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)=63
                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:~$ hadoop fs -cat /output_1901/part-r-00000
1901    46
hduser@bmsce-Precision-T1700:~$
```

b) find the mean max temperature for every month

- Program

```
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(&quot;Please Enter the input and output
parameters&quot;);
System.exit(-1);
}
Job job = new Job();
job.setJarByClass(MeanMaxDriver.class);
job.setJobName(&quot;Max temperature&quot;);
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&lt;LongWritable, Text,
Text, IntWritable&gt; {
public static final int MISSING = 9999;
public void map(LongWritable key, Text value,
Mapper&lt;LongWritable, Text, Text, IntWritable&gt;.Context context)
throws IOException, InterruptedException {
int temperature;
String line = value.toString();
String month = line.substring(19, 21);
if (line.charAt(87) == &#39;+&#39;) {
temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 &amp;&amp; quality.matches(&quot;[01459]&quot;))
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&lt;Text, IntWritable,
Text, IntWritable&gt; {
public void reduce(Text key, Iterable&lt;IntWritable&gt; values,
Reducer&lt;Text, IntWritable, Text, IntWritable&gt;.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 &gt; 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));
}
}
```

- **Output**

```
hduser@bmsce-OptiPlex-3060:~$ hadoop jar /home/hduser/Desktop/mean_max_temp.jar
meanmax.MeanMaxDriver /input_pranav/temp_1901.txt /avg_temp_output_meanmax_1901
22/06/21 10:17:01 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/21 10:17:01 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
22/06/21 10:17:01 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing
not performed. Implement the Tool interface and execute your application with ToolRunner
to remedy this.
22/06/21 10:17:01 INFO input.FileInputFormat: Total input paths to process : 1
22/06/21 10:17:01 INFO mapreduce.JobSubmitter: number of splits:1
22/06/21 10:17:01 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_local232634845_0001
22/06/21 10:17:01 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/21 10:17:01 INFO mapreduce.Job: Running job: job_local232634845_0001
22/06/21 10:17:01 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/21 10:17:01 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/21 10:17:01 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/21 10:17:01 INFO mapred.LocalJobRunner: Starting task:
attempt_local232634845_0001_m_000000_0
22/06/21 10:17:01 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/21 10:17:01 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/input_pranav/temp_1901.txt:0+888190
22/06/21 10:17:01 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/21 10:17:01 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/21 10:17:01 INFO mapred.MapTask: soft limit at 83886080
22/06/21 10:17:01 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/21 10:17:01 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/21 10:17:01 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/21 10:17:01 INFO mapred.LocalJobRunner:
22/06/21 10:17:01 INFO mapred.MapTask: Starting flush of map output
22/06/21 10:17:01 INFO mapred.MapTask: Spilling map output
22/06/21 10:17:01 INFO mapred.MapTask: bufstart = 0; bufend = 45948; bufvoid = 104857600
```

```
22/06/21 10:17:01 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend =
26188144(104752576); length = 26253/6553600
22/06/21 10:17:01 INFO mapred.MapTask: Finished spill 0
22/06/21 10:17:01 INFO mapred.Task: Task:attempt_local232634845_0001_m_000000_0 is
done. And is in the process of committing
22/06/21 10:17:01 INFO mapred.LocalJobRunner: map
22/06/21 10:17:01 INFO mapred.Task: Task 'attempt_local232634845_0001_m_000000_0'
done.
22/06/21 10:17:01 INFO mapred.LocalJobRunner: Finishing task:
attempt_local232634845_0001_m_000000_0
22/06/21 10:17:01 INFO mapred.LocalJobRunner: map task executor complete.
22/06/21 10:17:01 INFO mapred.LocalJobRunner: Waiting for reduce tasks
22/06/21 10:17:01 INFO mapred.LocalJobRunner: Starting task:
attempt_local232634845_0001_r_000000_0
22/06/21 10:17:01 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/21 10:17:01 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@1a055244
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: MergerManager:
memoryLimit=349752512, maxSingleShuffleLimit=87438128, mergeThreshold=230836672,
ioSortFactor=10, memToMemMergeOutputsThreshold=10
22/06/21 10:17:01 INFO reduce.EventFetcher: attempt_local232634845_0001_r_000000_0
Thread started: EventFetcher for fetching Map Completion Events
22/06/21 10:17:01 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map
attempt_local232634845_0001_m_000000_0 decomp: 59078 len: 59082 to MEMORY
22/06/21 10:17:01 INFO reduce.InMemoryMapOutput: Read 59078 bytes from map-output
for attempt_local232634845_0001_m_000000_0
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of
size: 59078, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->59078
22/06/21 10:17:01 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning
22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory
map-outputs and 0 on-disk map-outputs
22/06/21 10:17:01 INFO mapred.Merger: Merging 1 sorted segments
22/06/21 10:17:01 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of
total size: 59073 bytes
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merged 1 segments, 59078 bytes to disk
to satisfy reduce memory limit
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 1 files, 59082 bytes from disk
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from
memory into reduce
22/06/21 10:17:01 INFO mapred.Merger: Merging 1 sorted segments
22/06/21 10:17:01 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of
total size: 59073 bytes
22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.
```

```
22/06/21 10:17:01 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead,
use mapreduce.job.skiprecords
22/06/21 10:17:01 INFO mapred.Task: Task:attempt_local232634845_0001_r_000000_0 is
done. And is in the process of committing
22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:17:01 INFO mapred.Task: Task attempt_local232634845_0001_r_000000_0 is
allowed to commit now
22/06/21 10:17:01 INFO output.FileOutputCommitter: Saved output of task
'attempt_local232634845_0001_r_000000_0' to
hdfs://localhost:54310/avg_temp_output_meanmax_1901/_temporary/0/task_local2326348
45_0001_r_000000
22/06/21 10:17:01 INFO mapred.LocalJobRunner: reduce > reduce
22/06/21 10:17:01 INFO mapred.Task: Task 'attempt_local232634845_0001_r_000000_0'
done.
22/06/21 10:17:01 INFO mapred.LocalJobRunner: Finishing task:
attempt_local232634845_0001_r_000000_0
22/06/21 10:17:01 INFO mapred.LocalJobRunner: reduce task executor complete.
22/06/21 10:17:02 INFO mapreduce.Job: Job job_local232634845_0001 running in uber mode
: false
22/06/21 10:17:02 INFO mapreduce.Job:  map 100% reduce 100%
22/06/21 10:17:02 INFO mapreduce.Job: Job job_local232634845_0001 completed
successfully
22/06/21 10:17:02 INFO mapreduce.Job: Counters: 38
	File System Counters
		FILE: Number of bytes read=125588
		FILE: Number of bytes written=682332
		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=74
		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=45948
		Map output materialized bytes=59082
		Input split bytes=114
		Combine input records=0
		Combine output records=0
		Reduce input groups=12
		Reduce shuffle bytes=59082
		Reduce input records=6564
```

```
                Reduce output records=12
                Spilled Records=13128
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=54
                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=74
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -ls /avg_temp_meanmax_output
ls: `/avg_temp_meanmax_output': No such file or directory
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -ls /avg_temp_output_meanmax_1901
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-06-21 10:17
/avg_temp_output_meanmax_1901/_SUCCESS
-rw-r--r--   1 hduser supergroup         74 2022-06-21 10:17
/avg_temp_output_meanmax_1901/part-r-00000
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -cat /avg_temp_output_meanmax/part-r-00000
cat: `/avg_temp_output_meanmax/part-r-00000': No such file or directory
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs -cat /avg_temp_output_meanmax_1901/part-r-
00000
01      4
02      0
03      7
04      44
05      100
06      168
07      219
08      198
09      141
10      100
11      19
12      3
```

LAB-7

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

- **Program**

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(&quot;Usage: TopN &lt;in&gt; &lt;out&gt;&quot;);
System.exit(2);
}
Job job = Job.getInstance(conf);
job.setJobName(&quot;Top N&quot;);
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&lt;Object, Text,
Text, IntWritable&gt; {
private static final IntWritable one = new IntWritable(1);
private Text word = new Text();
private String tokens = &quot;[_|$#&lt;&gt;\\^=\\[\\]\\*/\\\\,;,.\\-
:()?!\&quot;&#39;]&quot;;
public void map(Object key, Text value, Mapper&lt;Object,
Text, Text, IntWritable&gt;.Context context) throws IOException,
InterruptedException {
String cleanLine =
value.toString().toLowerCase().replaceAll(this.tokens, &quot; &quot;);
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&lt;Text, IntWritable,
Text, IntWritable&gt; {
public void reduce(Text key, Iterable&lt;IntWritable&gt; values,
Reducer&lt;Text, IntWritable, Text, IntWritable&gt;.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&lt;Object, Text, Text,
IntWritable&gt; {
private static final IntWritable one = new IntWritable(1);
```

~~~~
private Text word = new Text();
private String tokens = &quot;[_|$#&lt;&gt;\\^=\\[\\]\\*/\\\\,;,.\\-
:()?!\&quot;&#39;]&quot;;
public vo```\\id map(Object key, Text value, Mapper&lt;Object,
Text, Text, IntWritable&gt;.Context context) throws IOException,
InterruptedException {
String cleanLine =
value.toString().toLowerCase().replaceAll(this.tokens, &quot; &quot;);
StringTokenizer itr = new StringTokenizer(cleanLine);
while (itr.hasMoreTokens()) {
this.word.set(itr.nextToken().trim());
context.write(this.word, one);
}
}
}
~~~~

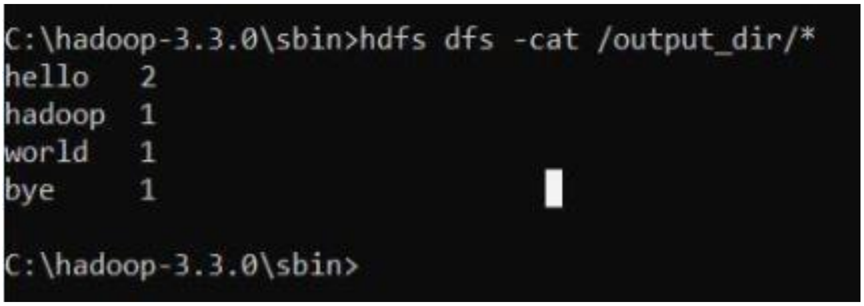
TopNReducer.class

~~~~
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&lt;Text, IntWritable,
Text, IntWritable&gt; {
private Map&lt;Text, IntWritable&gt; countMap = new HashMap&lt;&gt;();
public void reduce(Text key, Iterable&lt;IntWritable&gt; values,
Reducer&lt;Text, IntWritable, Text, IntWritable&gt;.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&lt;Text, IntWritable, Text,
IntWritable&gt;.Context context) throws IOException,
InterruptedException {
Map&lt;Text, IntWritable&gt; sortedMap =
MiscUtils.sortByValues(this.countMap);
int counter = 0;
for (Text key : sortedMap.keySet()) {
if (counter++ == 20)
~~~~

```
break;
context.write(key, sortedMap.get(key));
}
}
}
```

- **Output**



```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /output_dir/*
hello   2
hadoop  1
world   1
bye     1

C:\hadoop-3.3.0\sbin>
```

LAB-8

Create a Map Reduce program to demonstrating join operation

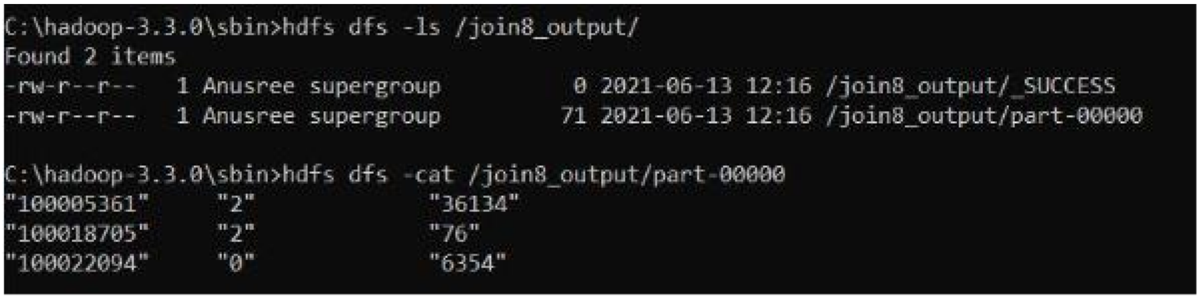
- **Program**

```
// 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&lt;TextPair,
Text&gt; {
@Override
public void configure(JobConf job) {}
@Override
public int getPartition(TextPair key, Text value, int numPartitions) {
return (key.getFirst().hashCode() &amp; Integer.MAX_VALUE) %
numPartitions;
}
}
@Override
public int run(String[] args) throws Exception {
if (args.length != 3) {
System.out.println(&quot;Usage: &lt;Department Emp Strength input&gt;
&lt;Department Name input&gt; &lt;output&gt;&quot;);
return -1;
}
JobConf conf = new JobConf(getConf(), getClass());

conf.setJobName(&quot;Join &#39;Department Emp Strength input&#39; with
&#39;Department Name
input&#39;&quot;);
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.cl
ass);
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&lt;TextPair, Text, Text,
Text&gt; {
@Override
public void reduce (TextPair key, Iterator&lt;Text&gt; values,
OutputCollector&lt;Text, Text&gt;
output, Reporter reporter)
throws IOException
{
Text nodeId = new Text(values.next());
while (values.hasNext()) {
Text node = values.next();
Text outValue = new Text(nodeId.toString() + &quot;\t\t&quot; + 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&lt;LongWritable, Text, TextPair,
Text&gt; {
@Override
public void map(LongWritable key, Text value,
OutputCollector&lt;TextPair, Text&gt; output,
Reporter reporter)
throws IOException
{
String valueString = value.toString();
String[] SingleNodeData = valueString.split(&quot;\t&quot;);
output.collect(new TextPair(SingleNodeData[0], &quot;1&quot;), 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&lt;LongWritable, Text, TextPair,
Text&gt; {
@Override
public void map(LongWritable key, Text value,
OutputCollector&lt;TextPair, Text&gt; output,
Reporter reporter)
throws IOException
{
String valueString = value.toString();
String[] SingleNodeData = valueString.split(&quot;\t&quot;);
output.collect(new TextPair(SingleNodeData[3], &quot;0&quot;), new
Text(SingleNodeData[9]));
}
}
// TextPair.java
import java.io.*;
import org.apache.hadoop.io.*;
public class TextPair implements WritableComparable&lt;TextPair&gt; {
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) &amp;&amp; second.equals(tp.second);
}
return false;
}
@Override
public String toString() {
```

```
return first + &quot;\t&quot; + 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 &amp;&amp; b instanceof TextPair) {
return ((TextPair) a).first.compareTo(((TextPair) b).first);
}
return super.compare(a, b);
}
} }
```

- **output**



```
C:\hadoop-3.3.0\sbin>hdfs dfs -ls /join8_output/
Found 2 items
-rw-r--r--   1 Anusree supergroup          0 2021-06-13 12:16 /join8_output/_SUCCESS
-rw-r--r--   1 Anusree supergroup         71 2021-06-13 12:16 /join8_output/part-00000

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /join8_output/part-00000
"100005361"     "2"             "36134"
"100018705"     "2"             "76"
"100022094"     "0"             "6354"
```

LAB-9

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

- **commands and outline:**

```
hduser@bmsce-OptiPlex-3060:~$ spark-shell
22/06/28 09:34:37 WARN Utils: Your hostname, bmsce-OptiPlex-3060 resolves to a loopback address: 127.0.1.1; using 10.124.7.72 instead (on interface enp1s0)
22/06/28 09:34:37 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
22/06/28 09:34:37 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://10.124.7.72:4040
Spark context available as 'sc' (master = local[*], app id = local-1656389082904).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.8
      /_/

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

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

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

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

scala> splitdata.collect;
res2: Array[String] = Array(hi, hw, are, ypu, how, is, your, job, how, is, your, family, how, is, your, brother, how, is, your, sister)
```

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

scala> mapdata.collect;
res3: Array[(String, Int)] = Array((hi,1), (hw,1), (are,1), (ypu,1), (how,1), (is,1), (your,1), (job,1),
(how,1), (is,1), (your,1), (family,1), (how,1), (is,1), (your,1), (brother,1), (how,1), (is,1), (your,1),
(sister,1))

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

scala> reducedata.collect;
res4: Array[(String, Int)] = Array((are,1), (brother,1), (is,4), (sister,1), (family,1), (how,4),
(ypu,1), (job,1), (hi,1), (hw,1), (your,4))

scala>
```

LAB-10

Using RDD and FlaMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark

- **commands and output:**

```
cala> val textFile=sc.textFile("/home/hduser/Desktop/sample.txt");
textFile: org.apache.spark.rdd.RDD[String] = /home/hduser/Desktop/sample.txt
MapPartitionsRDD[8] at textFile at <console>:24

scala> val counts=textFile.flatMap(line=>line.split("
")).map(word=>(word,1)).reduceByKey(_=_)
<console>:25: error: reassignment to val
    val counts=textFile.flatMap(line=>line.split(" ")).map(word=>(word,1)).reduceByKey(_=_)
                                                                                        ^

scala> val counts=textFile.flatMap(line=>line.split("
")).map(word=>(word,1)).reduceByKey(_+_)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[11] at reduceByKey at
<console>:25

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] = Map(is -> 4, how -> 4, your -> 4, are ->
1, brother -> 1, sister -> 1, family -> 1, ypu -> 1, job -> 1, hi -> 1, hw -> 1)

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

scala> for((k,v)<-sorted)
     | {
     |  if(v>4)
     |  {
     |    print(k+",")
     |     print(v)
     |     println()
     |  }
     | }
//SINCE SAMPLE TEXT FILE DOESNT HAVE WORD WITH FREQUENCY >4,NO OUTPUT
```