VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS
(20CS6PEBDA)

Submitted by

PRATIBHA JAMKHANDI(1BM19CS119)

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 **PRATIBHA JAMKHANDI(1BM19CS119),** 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 Choudhury
Assistant Professor

Department of CSE
BMSCE, Bengaluru

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

`

Index Sheet

Course Outcome

CO 1	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO 2	Analyze the Big Data and obtain insight using data analytics mechanisms.
CO 3	Design and implement Big data applications by applying NoSQL, Hadoop or Spark

LAB 1

CREATE DATABASE IN MONGODB.

use myDB;

```
> use myDB;
switched to db myDB
> db;
myDB
```

CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS

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

db.createCollection(“Student”);

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

2. To drop a collection by the name “Student”.

db.Student.drop();

```
> db.Student.drop();
true
```

3. Create a collection by the name “Students” and store the following data in it.

db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing" });

```
> db.Student.insert({_id:1,StudName:"pratibha",Grade:"vii",Hobbies:"Chess"});
WriteResult({ "nInserted" : 1 })
```

4. Insert the document for “Rahul” into the Students collection only if it does not already exist in the collection. However, if it is already present in the collection, then update the document with new values. (Update his Hobbies from “Skating” to “Chess”.) Use “Update else insert” (if there is an existing document, it will attempt to update it, if there is no existing document then it will insert it).

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

```
> db.Student.update({_id:3,StudName:"rahul",Grade:"vii"},{$set:{Hobbies:"Skating"}},{upsert:true});
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 3 })
```

5. FIND METHOD

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

db.Student.find({StudName:"pratibha"});

```
> db.Student.find({StudName:"pratibha"});
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
```

B. To display only the StudName and Grade from all the documents of the Students collection. The identifier_id should be suppressed and NOT displayed.

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

```
> db.Student.find({},{StudName:1,Grade:1,_id:0});
{ "StudName" : "pratibha", "Grade" : "vii" }
{ "StudName" : "prathiksha", "Grade" : "viii" }
{ "Grade" : "vii", "StudName" : "rahul" }
```

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

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

```
> db.Student.find({Grade:{$eq:"vii"}}).pretty();
{
        "_id" : 1,
        "StudName" : "pratibha",
        "Grade" : "vii",
        "Hobbies" : "Chess"
}
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating" }
```

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

db.Student.find({Hobbies :{ $in: ['Chess','Skating']}}).pretty ();

```
> db.Student.find({Hobbies:{$in:['Chess','Skating']}}).pretty();
{
        "_id" : 1,
        "StudName" : "pratibha",
        "Grade" : "vii",
        "Hobbies" : "Chess"
}
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating" }
```

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

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

```
> db.Student.find({StudName:/^r/}).pretty();
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating" }
```

F. To find documents from the Students collection where the StudName has an “u” in any position.

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

```
> db.Student.find({StudName:/u/}).pretty();
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating" }
```

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

db.Student.count();

```
> db.Student.count();
3
```

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

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

```
> db.Student.find().sort({StudName:-1});
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating" }
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
{ "_id" : 2, "StudName" : "prathiksha", "Grade" : "viii", "Hobbies" : "cycling" }
```

6. Save Method :

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

db.Students.save({StudName:”Vamsi”, Grade:”VI”});

```
> db.Student.save({StudName:"Prasansa",Grade:"viii"});
WriteResult({ "nInserted" : 1 })
> db.Student.find();
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
{ "_id" : 2, "StudName" : "prathiksha", "Grade" : "viii", "Hobbies" : "cycling" }
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating" }
{ "_id" : ObjectId("629e2c835e84878fe9a0aea0"), "StudName" : "Prasansa", "Grade" : "viii" }
```

7. Add a new field to existing Document:

db.Students.update({_id:3},{$set:{Location:”Network”}});

```
> db.Student.update({_id:3},{$set:{Location:"Network"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find();
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
{ "_id" : 2, "StudName" : "prathiksha", "Grade" : "viii", "Hobbies" : "cycling" }
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating", "Location" : "Network" }
{ "_id" : ObjectId("629e2c835e84878fe9a0aea0"), "StudName" : "Prasansa", "Grade" : "viii" }
```

8. Remove the field in an existing Document

db.Students.update({_id:3},{$unset:{Location:"Network"}});

```
> db.Student.update({_id:3},{$unset:{Location:"Network"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find();
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
{ "_id" : 2, "StudName" : "prathiksha", "Grade" : "viii", "Hobbies" : "cycling" }
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating" }
{ "_id" : ObjectId("629e2c835e84878fe9a0aea0"), "StudName" : "Prasansa", "Grade" : "viii" }
```

9.Finding Document based on search criteria suppressing few fields

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

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

10. To find those documents where the Grade is not set to 'VII'

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

```
> db.Student.find({Grade:{$ne:'vii'}});
{ "_id" : 2, "StudName" : "prathiksha", "Grade" : "viii", "Hobbies" : "cycling" }
{ "_id" : ObjectId("629e2c835e84878fe9a0aea0"), "StudName" : "Prasansa", "Grade" : "viii" }
```

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

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

```
> db.Student.find({StudName:/a$/});
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
{ "_id" : 2, "StudName" : "prathiksha", "Grade" : "viii", "Hobbies" : "cycling" }
{ "_id" : ObjectId("629e2c835e84878fe9a0aea0"), "StudName" : "Prasansa", "Grade" : "viii" }
```

12. to set a particular field value to NULL

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

```
> db.Student.update({_id:3},{$set:{Location:null}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find();
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
{ "_id" : 2, "StudName" : "prathiksha", "Grade" : "viii", "Hobbies" : "cycling" }
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating", "Location" : null }
{ "_id" : ObjectId("629e2c835e84878fe9a0aea0"), "StudName" : "Prasansa", "Grade" : "viii" }
```

13.Retrieve first 3 documents

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

```
> db.Student.find({Grade:'vii'}).limit(3)
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating", "Location" : null }
> db.Student.find({Grade:'vii'}).limit(1)
{ "_id" : 1, "StudName" : "pratibha", "Grade" : "vii", "Hobbies" : "Chess" }
```

14.To Skip the 1 st two documents from the Students Collections

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

```
> db.Student.find().skip(2)
{ "_id" : 3, "Grade" : "vii", "StudName" : "rahul", "Hobbies" : "Skating", "Location" : null }
{ "_id" : ObjectId("629e2c835e84878fe9a0aea0"), "StudName" : "Prasansa", "Grade" : "viii" }
```

LAB -2

1.Create a keyspace by name Employee

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

2. Create a column family by name

Employee-Info with attributes

Emp_Id Primary Key, Emp_Name,

Designation, Date_of_Joining, Salary,

Dept_Name

```
cqlsh:employee> create table employee_info(
            ... Emp_id int PRIMARY KEY,
            ... Emp_name text,
            ... Designation text,
            ... Date_of_joining timestamp,
            ... Salary double,
            ... Dept_name text
            ... );
```

3. Insert the values into the table in batch

```
cqlsh:employee> begin batch
           ... insert into employee_info (emp_id,emp_name,designation,date_of_joining,salary,dept_name) values (1,'prathiksha','HR','2020-03-01',50000,'HR dept')
           ... apply batch;
cqlsh:employee> select * from Employee_info;

 emp_id | date_of_joining                 | dept_name | designation | emp_name   | salary
--------+---------------------------------+-----------+-------------+------------+--------
      1 | 2020-02-29 18:30:00.000000+0000 |   HR dept |          HR | prathiksha |  50000

(1 rows)
cqlsh:employee> begin batch
           ... insert into employee_info (emp_id,emp_name,designation,date_of_joining,salary,dept_name) values (2,'pranav','Editor','2020-04-01',40000,'Marketing dept')
           ... insert into employee_info (emp_id,emp_name,designation,date_of_joining,salary,dept_name) values (3,'rahul','Software Engineer','2020-05-01',60000,'technical')
           ... insert into employee_info (emp_id,emp_name,designation,date_of_joining,salary,dept_name) values (4,'anuradha','Security Manager','2020-05-01',60000,'security')
           ... insert into employee_info (emp_id,emp_name,designation,date_of_joining,salary,dept_name) values (5,'sonal','HR employee','2020-05-01',60000,'HR dept')
           ... apply batch;
```

```
 emp_id | date_of_joining                 | dept_name      | designation       | emp_name   | salary
--------+---------------------------------+----------------+-------------------+------------+--------
      5 | 2020-04-30 18:30:00.000000+0000 |        HR dept |       HR employee |      sonal |  60000
      1 | 2020-02-29 18:30:00.000000+0000 |        HR dept |                HR | prathiksha |  50000
      2 | 2020-03-31 18:30:00.000000+0000 | Marketing dept |            Editor |     pranav |  40000
      4 | 2020-04-30 18:30:00.000000+0000 |       security |  Security Manager |   anuradha |  60000
      3 | 2020-04-30 18:30:00.000000+0000 |      technical | Software Engineer |      rahul |  60000
```

4. Update Employee name and Department of Emp-Id 121

```
cqlsh:employee> update employee_info set emp_name='prashansa',dept_name='Marketing' where emp_id=1;
cqlsh:employee> select * from Employee_info;

 emp_id | date_of_joining                 | dept_name      | designation       | emp_name   | salary
--------+---------------------------------+----------------+-------------------+------------+--------
      5 | 2020-04-30 18:30:00.000000+0000 |        HR dept |       HR employee |      sonal |  60000
      1 | 2020-02-29 18:30:00.000000+0000 |      Marketing |                HR | prashansa |  50000
      2 | 2020-03-31 18:30:00.000000+0000 | Marketing dept |            Editor |     pranav |  40000
      4 | 2020-04-30 18:30:00.000000+0000 |       security |  Security Manager |   anuradha |  60000
      3 | 2020-04-30 18:30:00.000000+0000 |      technical | Software Engineer |      rahul |  60000
```

5. Sort the details of Employee records based on salary

```
create table emp( id int, salary int, name text,primary key(id,salary) );
cqlsh:employee> begin batch
            ... insert into emp(id,salary,name)values (1,10000,'prathiksha')
            ... insert into emp(id,salary,name)values (2,10000,'pooja')
            ... insert into emp(id,salary,name)values (3,10000,'prema')
            ... insert into emp(id,salary,name)values (3,20000,'rahul')
            ... insert into emp(id,salary,name)values (4,30000,'raghu')
            ... apply batch
            ... ;

cqlsh:employee> paging off;
Disabled Query paging.
cqlsh:employee> select *from emp where id in (1,2,3,4) order by salary;

 id | salary | name
----+--------+------------
  1 |  10000 | prathiksha
  2 |  10000 |      pooja
  3 |  10000 |      prema
  3 |  20000 |      rahul
  4 |  30000 |      raghu
```

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

```
cqlsh:employee> alter table employee_info add projects text;
cqlsh:employee> describe table Employee_info;

CREATE TABLE employee.employee_info (
    emp_id int PRIMARY KEY,
    date_of_joining timestamp,
    dept_name text,
    designation text,
    emp_name text,
    projects text,
    salary double
```

7. Update the altered table to add project names.

```
cqlsh:employee> begin batch
            ... update employee_info set projects='abc' where emp_id=1
            ... update employee_info set projects='def' where emp_id=2
            ... update employee_info set projects='ghi' where emp_id=3
            ... update employee_info set projects='jkl' where emp_id=4
            ... update employee_info set projects='mno' where emp_id=5
            ... apply batch;
```

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

 emp_id | date_of_joining                 | dept_name      | designation       | emp_name  | projects | salary
--------+---------------------------------+----------------+-------------------+-----------+----------+--------
      5 | 2020-04-30 18:30:00.000000+0000 |        HR dept |       HR employee |     sonal |      mno |  60000
      1 | 2020-02-29 18:30:00.000000+0000 |      Marketing |                HR | prashansa |      abc |  50000
      2 | 2020-03-31 18:30:00.000000+0000 | Marketing dept |            Editor |    pranav |      def |  40000
      4 | 2020-04-30 18:30:00.000000+0000 |       security |  Security Manager |  anuradha |      jkl |  60000
      3 | 2020-04-30 18:30:00.000000+0000 |      technical | Software Engineer |     rahul |      ghi |  60000
```

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

```
cqlsh:employee> insert into Employee_info (emp_id,emp_name,designation,date_of_joining,salary,dept_name)
                values (19,'prithvi',senior_developer','2022-08-09',40000,'Developing') using TTL 50;
```

```
cqlsh:employee> insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name) values(171,'Tyax','CEO','2023-08-29',57000,'Managing') USING TTL 700
;
cqlsh:employee> select ttl(emp_name) from employee_info where emp_id=171;

 ttl(emp_name)
---------------
           634

(1 rows)
```

LAB -3

3. Perform the following DB operations using Cassandra.

1. Create a keyspace by name Library

```
CREATE KEYSPACE LIBRARY1 WITH REPLICATION = {
   ... 'class':'SimpleStrategy',
   ... 'replication_factor':1};
```

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

```
create table library_info( stud_id int, counter_value counter, stud_name text,
book_name text, book_id int, date_of_issue timestamp,PRIMARY
KEY(stud_id,stud_name,book_name,book_id,date_of_issue));
```

3. Insert the values into the table in

```
batch update library_info
          ... set counter_value = counter_value +1 where stud_id=121 and
stud_name='Prema' and book_name='cns' and book_id=113 and
date_of_issue='2022-06-29';
select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                    |
counter_value
_______+________+________+_______+__________________________+___________
     121 |      Prema |        cns |      113 | 2022-06-28 18:30:00.000000+0000 |             1
```

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

.

```
 update library_info set counter_value = counter_value +1 where stud_id=121 and
stud_name='Prema' and book_name='cns' and book_id=113 and
date_of_issue='2022-06-29';
cqlsh:library1> select * from library_info;
```

```
 stud_id | stud_name | book_name | book_id | date_of_issue                    |
counter_value
_______+________+________+______+_________________________+__________
    121 |     Prema |       cns |     113 | 2022-06-28 18:30:00.000000+0000 |             2
```

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

```
cqlsh:library1> update library_info set counter_value = counter_value +2 where
stud_id=111 and stud_name='Pooja' and book_name='bda' and book_id=112 and
date_of_issue='202
2-06-29';
select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                    |
counter_value
_______+________+________+______+_________________________+__________
    111 |     Pooja |       bda |     112 | 2022-06-28 18:30:00.000000+0000 |             2
    121 |     Prema |       cns |     113 | 2022-06-28 18:30:00.000000+0000 |             2
```

6. Export the created column to a csv file COPY library_info(stud_id,counter_value,stud_name,book_name,book_id,date_of_issue) TO 'lib1.csv'
... ;

```
Using 7 child processes

Starting copy of library1.library_info with columns [stud_id, counter_value,
stud_name, book_name, book_id, date_of_issue].
Processed: 2 rows; Rate:      17 rows/s; Avg. rate:      17 rows/s
2 rows exported to 1 files in 0.143 seconds.
```

7. Import a given csv dataset from local file system into Cassandra column family TRUNCATE library_info;

```
cqlsh:library1> select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue | counter_value
```

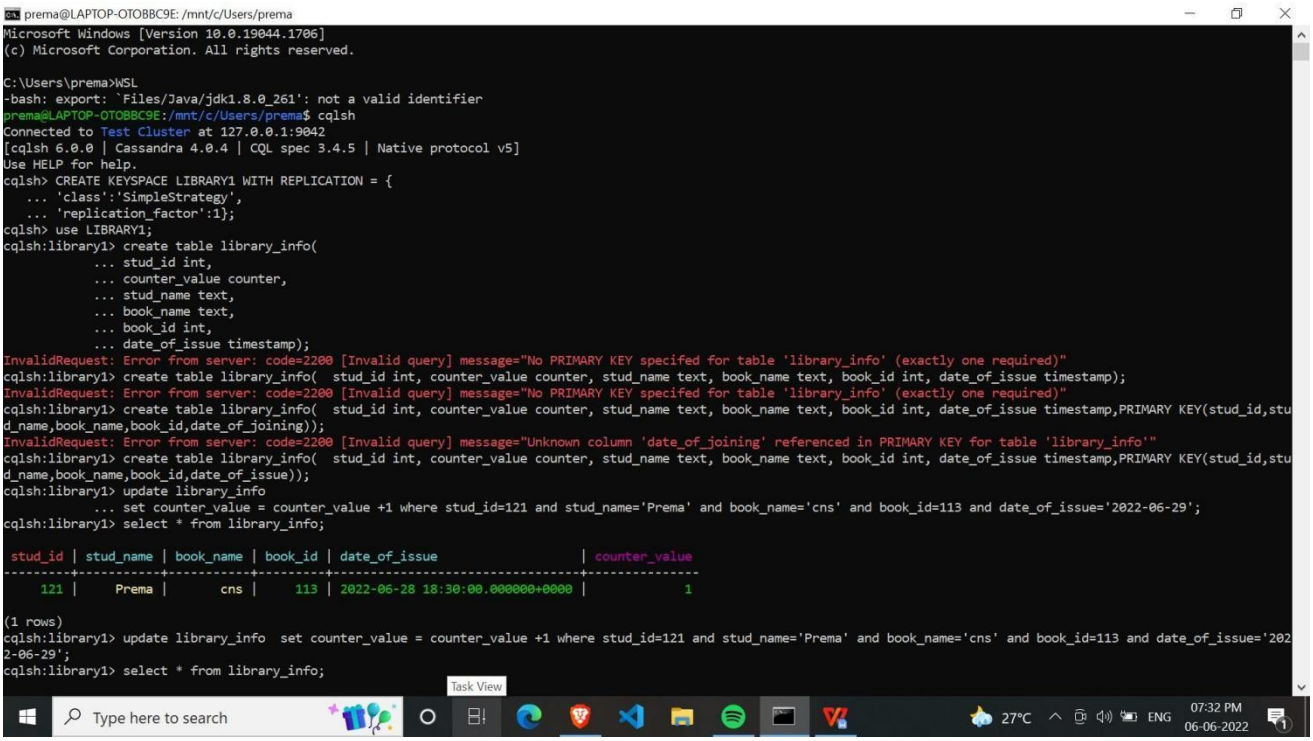
_____+_____+_____+_____+_____+_____

(0 rows)
cqlsh:library1>
COPY
library_info(stud_id,counter_value,stud_name,book_name,book_id,date_of_issue)
FROM 'lib1.csv' ;
Using 7 child processes

Starting copy of library1.library_info with columns [stud_id, counter_value, stud_name, book_name, book_id, date_of_issue].
Processed: 2 rows; Rate: 4 rows/s; Avg. rate: 6 rows/s
2 rows imported from 1 files in 0.364 seconds (0 skipped).
cqlsh:library1> select * from library_info;

```
 stud_id | stud_name | book_name | book_id | date_of_issue                   |
counter_value
_______+__________+__________+________+_________________________________+______________
     111 |      Pooja |       bda |     112 | 2022-06-28 18:30:00.000000+0000 |             2
     121 |      Prema |       cns |     113 | 2022-06-28 18:30:00.000000+0000 |             2
```

Output screenshots:



prema@LAPTOP-OTOBBC9E: /mnt/c/Users/prema

```
cqlsh:library1> select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     121 |     Prema |       cns |     113 | 2022-06-28 18:30:00.000000+0000 |             2

(1 rows)
cqlsh:library1> update library_info  set counter_value = counter_value +2 where stud_id=121 and stud_name='Pooja' and book_name='bda' and book_id=112 and date_of_issue='202
2-06-29';
cqlsh:library1> update library_info  set counter_value = counter_value +2 where stud_id=111 and stud_name='Pooja' and book_name='bda' and book_id=112 and date_of_issue='202
2-06-29';
cqlsh:library1> select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     111 |     Pooja |       bda |     112 | 2022-06-28 18:30:00.000000+0000 |             2
     121 |     Pooja |       bda |     112 | 2022-06-28 18:30:00.000000+0000 |             2
     121 |     Prema |       cns |     113 | 2022-06-28 18:30:00.000000+0000 |             2

(3 rows)
cqlsh:library1> delete from library_info where stud_id = 121 and stud_name = 'Pooja';
cqlsh:library1> select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     111 |     Pooja |       bda |     112 | 2022-06-28 18:30:00.000000+0000 |             2
     121 |     Prema |       cns |     113 | 2022-06-28 18:30:00.000000+0000 |             2

(2 rows)
cqlsh:library1>
```

Type here to search 27°C ENG 07:33 PM 06-06-2022

LAB 4

Screenshot of Hadoop installed

pratibha@LAPTOP-4C433GMJ: ~

```
pratibha@LAPTOP-4C433GMJ:/mnt/c/WINDOWS/system32$ cd
pratibha@LAPTOP-4C433GMJ:~$
pratibha@LAPTOP-4C433GMJ:~$ cd  ~/hadoop/hadoop-3.3.0/
pratibha@LAPTOP-4C433GMJ:~/hadoop/hadoop-3.3.0$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [LAPTOP-4C433GMJ]
pratibha@LAPTOP-4C433GMJ:~/hadoop/hadoop-3.3.0$ ssh localhost
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Jul 12 08:15:37 IST 2022

  System load:  0.36               Processes:             21
  Usage of /:   1.2% of 250.98GB   Users logged in:       0
  Memory usage: 24%                IPv4 address for eth0: 172.30.189.139
  Swap usage:   0%


299 updates can be installed immediately.
183 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Jul 12 08:13:03 2022 from 127.0.0.1
pratibha@LAPTOP-4C433GMJ:~$ jps
823 NameNode
1420 Jps
973 DataNode
1215 SecondaryNameNode
pratibha@LAPTOP-4C433GMJ:~$ hdfs dfs -mkdir /pratibha
mkdir: `/pratibha': File exists
pratibha@LAPTOP-4C433GMJ:~$ hdfs dfs -mkdir /pratibhaJ
```

LAB 5

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.
b) find the mean max temperature for every month

```
package temp;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class AverageDriver {
  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Please Enter the input and output
parameters");
      System.exit(-1);
    }
    Job job = new Job();
    job.setJarByClass(AverageDriver.class);
    job.setJobName("Max temperature");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.setMapperClass(AverageMapper.class);
    job.setReducerClass(AverageReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

AverageMapper

```
package temp;

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

public class AverageMapper extends Mapper<LongWritable, Text, Text,
IntWritable> {
```

```java
  public static final int MISSING = 9999;

  public void map(LongWritable key, Text value, Mapper<LongWritable,
Text, Text, IntWritable>.Context context) throws IOException,
InterruptedException {
    int temperature;
    String line = value.toString();
    String year = line.substring(15, 19);
    if (line.charAt(87) == '+') {
      temperature = Integer.parseInt(line.substring(88, 92));
    } else {
      temperature = Integer.parseInt(line.substring(87, 92));
    }
    String quality = line.substring(92, 93);
    if (temperature != 9999 && quality.matches("[01459]"))
      context.write(new Text(year), new IntWritable(temperature));
  }
}
```

AverageReducer

```java
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));
  }
}
```

SCREENSHOTS:

```
hduser@bmsce-Precision-T1700:/home/bmsce$ hadoop fs -copyFromLocal /home/bmsce/Downloads/1901 /pratibha/input.txt
hduser@bmsce-Precision-T1700:/home/bmsce$ hadoop jar /home/bmsce/eclipse-workspace/temp2.jar temp.AverageDriver /pratibha/input.txt /pratibha/outputavg
22/06/21 10:27:11 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
22/06/21 10:27:11 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
22/06/21 10:27:11 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 10:27:11 INFO input.FileInputFormat: Total input paths to process : 1
22/06/21 10:27:11 INFO mapreduce.JobSubmitter: number of splits:1
22/06/21 10:27:11 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1259082108_0001
22/06/21 10:27:11 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/21 10:27:11 INFO mapreduce.Job: Running job: job_local1259082108_0001
22/06/21 10:27:11 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/21 10:27:11 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/21 10:27:11 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/21 10:27:11 INFO mapred.LocalJobRunner: Starting task: attempt_local1259082108_0001_m_000000_0
22/06/21 10:27:11 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/21 10:27:11 INFO mapred.MapTask: Processing split: hdfs://localhost:54310/pratibha/input.txt:0+888190
22/06/21 10:27:11 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/21 10:27:11 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/21 10:27:11 INFO mapred.MapTask: soft limit at 83886080
22/06/21 10:27:11 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/21 10:27:11 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/21 10:27:11 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/21 10:27:11 INFO mapred.LocalJobRunner:
22/06/21 10:27:11 INFO mapred.MapTask: Starting flush of map output
22/06/21 10:27:11 INFO mapred.MapTask: Spilling map output
22/06/21 10:27:11 INFO mapred.MapTask: bufstart = 0; bufend = 59076; bufvoid = 104857600
22/06/21 10:27:11 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26188144(104752576); length = 26253/6553600
22/06/21 10:27:11 INFO mapred.MapTask: Finished spill 0
22/06/21 10:27:11 INFO mapred.Task: Task:attempt_local1259082108_0001_m_000000_0 is done. And is in the process of committing
22/06/21 10:27:11 INFO mapred.LocalJobRunner: map
22/06/21 10:27:11 INFO mapred.Task: Task 'attempt_local1259082108_0001_m_000000_0' done.
22/06/21 10:27:11 INFO mapred.LocalJobRunner: Finishing task: attempt_local1259082108_0001_m_000000_0
22/06/21 10:27:11 INFO mapred.LocalJobRunner: map task executor complete.
22/06/21 10:27:11 INFO mapred.LocalJobRunner: Waiting for reduce tasks
22/06/21 10:27:11 INFO mapred.LocalJobRunner: Starting task: attempt_local1259082108_0001_r_000000_0
22/06/21 10:27:11 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
22/06/21 10:27:11 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@65367f35
22/06/21 10:27:11 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=349752512, maxSingleShuffleLimit=87438128, mergeThreshold=230836672, ioSortFactor=10, memToMemMergeOutputsThreshold=10
22/06/21 10:27:11 INFO reduce.EventFetcher: attempt_local1259082108_0001_r_000000_0 Thread started: EventFetcher for fetching Map Completion Events
22/06/21 10:27:11 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map attempt_local1259082108_0001_m_000000_0 decomp: 72206 len: 72210 to MEMORY
22/06/21 10:27:11 INFO reduce.InMemoryMapOutput: Read 72206 bytes from map-output for attempt_local1259082108_0001_m_000000_0
22/06/21 10:27:11 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 72206, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->72206
22/06/21 10:27:11 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning
22/06/21 10:27:11 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:27:11 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs
```

```
22/06/21 10:27:11 INFO mapred.LocalJobRunner: Finishing task: attempt_local1259082108_0001_r_000000_0
22/06/21 10:27:11 INFO mapred.LocalJobRunner: reduce task executor complete.
22/06/21 10:27:12 INFO mapreduce.Job: Job job_local1259082108_0001 running in uber mode : false
22/06/21 10:27:12 INFO mapreduce.Job:  map 100% reduce 100%
22/06/21 10:27:12 INFO mapreduce.Job: Job job_local1259082108_0001 completed successfully
22/06/21 10:27:12 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=153100
                FILE: Number of bytes written=725600
                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=106
                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)=61
                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:/home/bmsce$ hadoop fs -ls /pratibha/outputavg/
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-06-21 10:27 /pratibha/outputavg/_SUCCESS
-rw-r--r--   1 hduser supergroup          8 2022-06-21 10:27 /pratibha/outputavg/part-r-00000
hduser@bmsce-Precision-T1700:/home/bmsce$ hadoop fs -cat /pratibha/outputavg/part-r-00000
1901    46
```

```
hduser@bmsce-Precision-T1700:/home/bmsce$ hadoop fs -ls /pratibha/outputavg1/
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-06-21 10:29 /pratibha/outputavg1/_SUCCESS
-rw-r--r--   1 hduser supergroup          8 2022-06-21 10:29 /pratibha/outputavg1/part-r-00000
hduser@bmsce-Precision-T1700:/home/bmsce$ hadoop fs -cat /pratibha/outputavg1/part-r-00000
1902    21
```

b) find the mean max temperature for every month

MeanMax
MeanMaxDriver.class

```
package meanmax;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class MeanMaxDriver {
  public static void main(String[] args) throws Exception {
    if (args.length != 2) {
      System.err.println("Please Enter the input and output
parameters");
      System.exit(-1);
    }
    Job job = new Job();
    job.setJarByClass(MeanMaxDriver.class);
    job.setJobName("Max temperature");
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    job.setMapperClass(MeanMaxMapper.class);
    job.setReducerClass(MeanMaxReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

MeanMaxMapper.class

```
package meanmax;

import java.io.IOException;
```

```java
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

```java
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));
  }
}
```

```
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 execute your application with ToolRunner to remedy this.
2021-05-21 20:28:06,916 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621608943095_0001
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
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /meanmax_output/*
01      4
02      0
03      7
04      44
05      100
06      168
07      219
08      198
09      141
10      100
11      19
12      3

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

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.

Driver-TopN.class

```
package samples.topn;

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

public class TopN {
  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = (new GenericOptionsParser(conf,
args)).getRemainingArgs();
    if (otherArgs.length != 2) {
```

```
      System.err.println("Usage: TopN <in> <out>");
      System.exit(2);
    }
    Job job = Job.getInstance(conf);
    job.setJobName("Top N");
    job.setJarByClass(TopN.class);
    job.setMapperClass(TopNMapper.class);
    job.setReducerClass(TopNReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }

  public static class TopNMapper extends Mapper<Object, Text, Text,
IntWritable> {
    private static final IntWritable one = new IntWritable(1);

    private Text word = new Text();

    private String tokens =
"[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";

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

TopNCombiner.class

```
package samples.topn;

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

public class TopNCombiner extends Reducer<Text, IntWritable, Text,
```

```
IntWritable> {
  public void reduce(Text key, Iterable<IntWritable> values,
Reducer<Text, IntWritable, Text, IntWritable>.Context context) throws
IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values)
      sum += val.get();
    context.write(key, new IntWritable(sum));
  }
}
```

TopNMapper.class

```
package samples.topn;

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class TopNMapper extends Mapper<Object, Text, Text,
IntWritable> {
  private static final IntWritable one = new IntWritable(1);

  private Text word = new Text();

  private String tokens =
"[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";

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

TopNReducer.class

```
package samples.topn;

import java.io.IOException;
import java.util.HashMap;
```

```java
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));
    }
  }
}
```

```
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
```

```
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
```

```
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

```
// JoinDriver.java
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.mapred.lib.MultipleInputs;
import org.apache.hadoop.util.*;

public class JoinDriver extends Configured implements Tool {

public static class KeyPartitioner implements Partitioner<TextPair, Text> {
@Override
public void configure(JobConf job) {}

@Override
public int getPartition(TextPair key, Text value, int numPartitions) {
return (key.getFirst().hashCode() & Integer.MAX_VALUE) %
numPartitions;
}
}

@Override

public int run(String[] args) throws Exception {

if (args.length != 3) {
System.out.println("Usage: <Department Emp Strength input>

<Department Name input> <output>");
return -1;
}

JobConf conf = new JobConf(getConf(), getClass());

conf.setJobName("Join 'Department Emp Strength input' with 'Department Name
input'");

Path AInputPath = new Path(args[0]);
```

```
Path BInputPath = new Path(args[1]);
Path outputPath = new Path(args[2]);

MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class,

Posts.class);

MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class,

User.class);

FileOutputFormat.setOutputPath(conf, outputPath);

conf.setPartitionerClass(KeyPartitioner.class);

conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);

conf.setMapOutputKeyClass(TextPair.class);

conf.setReducerClass(JoinReducer.class);

conf.setOutputKeyClass(Text.class);

JobClient.runJob(conf);

return 0;
}

public static void main(String[] args) throws Exception {

int exitCode = ToolRunner.run(new JoinDriver(), args);
System.exit(exitCode);
}
}

// JoinReducer.java
import java.io.IOException;
import java.util.Iterator;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class JoinReducer extends MapReduceBase implements Reducer<TextPair, Text, Text,
Text> {

@Override
public void reduce (TextPair key, Iterator<Text> values, OutputCollector<Text, Text>
output, Reporter reporter)

throws IOException
```

```
{

Text nodeId = new Text(values.next());
while (values.hasNext()) {

Text node = values.next();
Text outValue = new Text(nodeId.toString() + "\t\t" + node.toString());
output.collect(key.getFirst(), outValue);
}
}
}

// User.java
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FSDataOutputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.IntWritable;

public class User extends MapReduceBase implements Mapper<LongWritable, Text, TextPair,
Text> {

@Override
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reporter reporter)

throws IOException

{

String valueString = value.toString();

String[] SingleNodeData = valueString.split("\t");
output.collect(new TextPair(SingleNodeData[0], "1"), new

Text(SingleNodeData[1]));
}
}

//Posts.java
import java.io.IOException;

import org.apache.hadoop.io.*;
```

```
import org.apache.hadoop.mapred.*;

public class Posts extends MapReduceBase implements Mapper<LongWritable, Text, TextPair,
Text> {

@Override
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reporter reporter)
throws IOException
{
String valueString = value.toString();
String[] SingleNodeData = valueString.split("\t");
output.collect(new TextPair(SingleNodeData[3], "0"), new

Text(SingleNodeData[9]));
}
}

// TextPair.java
import java.io.*;

import org.apache.hadoop.io.*;

public class TextPair implements WritableComparable<TextPair> {

private Text first;
private Text second;

public TextPair() {
set(new Text(), new Text());
}

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

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

public void set(Text first, Text second) {
this.first = first;
this.second = second;
}

public Text getFirst() {
return first;
}

public Text getSecond() {
```

```
return second;
}

@Override
public void write(DataOutput out) throws IOException {
first.write(out);
second.write(out);
}

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

@Override
public int hashCode() {
return first.hashCode() * 163 + second.hashCode();
}

@Override
public boolean equals(Object o) {
if (o instanceof TextPair) {
TextPair tp = (TextPair) o;
return first.equals(tp.first) && second.equals(tp.second);
}
return false;
}

@Override
public String toString() {
return first + "\t" + second;
}

@Override
public int compareTo(TextPair tp) {
int cmp = first.compareTo(tp.first);
if (cmp != 0) {
return cmp;
}
return second.compareTo(tp.second);
}
// ^^ TextPair

// vv TextPairComparator
public static class Comparator extends WritableComparator {

private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

public Comparator() {
```

```
super(TextPair.class);
}

@Override
public int compare(byte[] b1, int s1, int l1,
byte[] b2, int s2, int l2) {

try {
int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);
int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);
int cmp = TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);
if (cmp != 0) {
return cmp;
}
return TEXT_COMPARATOR.compare(b1, s1 + firstL1, l1 - firstL1,


b2, s2 + firstL2, l2 - firstL2);
} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}
}

static {
WritableComparator.define(TextPair.class, new Comparator());
}
public static class FirstComparator extends WritableComparator {

private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

public FirstComparator() {
super(TextPair.class);
}

@Override
public int compare(byte[] b1, int s1, int l1,
byte[] b2, int s2, int l2) {

try {
int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);
int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);
return TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);
} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}

@Override
public int compare(WritableComparable a, WritableComparable b) {
```

```
if (a instanceof TextPair && b instanceof TextPair) {
return ((TextPair) a).first.compareTo(((TextPair) b).first);
}
return super.compare(a, b);
}
}  }
```

LAB8/Department_Employee_join_example/DeptName.txt

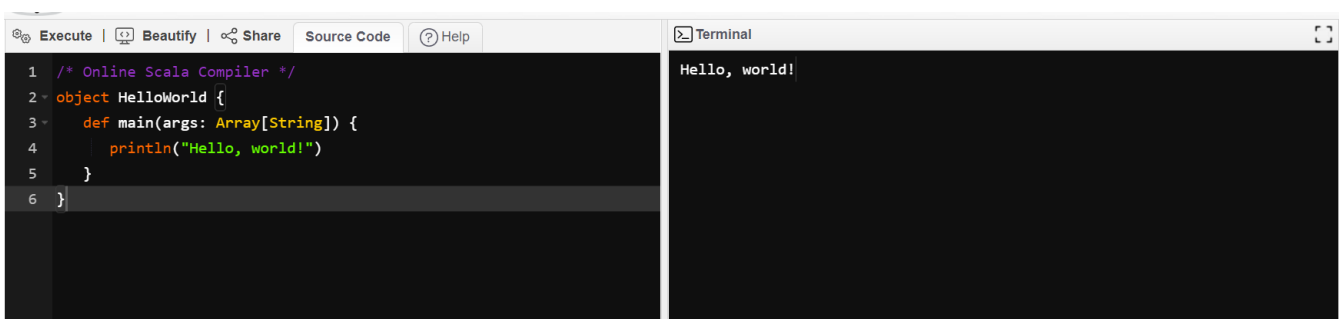
```
C:\hadoop-3.3.0\sbin>hdfs dfs -ls /join8_output/
Found 2 items
-rw-r--r--   1 Anusree supergroup          0 2021-06-13 12:16 /join8_output/_SUCCESS
-rw-r--r--   1 Anusree supergroup         71 2021-06-13 12:16 /join8_output/part-00000

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

```
val data=sc.textFile("sparkdata.txt")
data.collect;
val splitdata = data.flatMap(line => line.split(" "));
splitdata.collect;
val mapdata = splitdata.map(word => (word,1));
mapdata.collect;
val reducedata = mapdata.reduceByKey(_+_);
reducedata.collect;
```



LAB 10:

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

package scalawordcount

```
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.rdd.RDD.rddToPairRDDFunctions
import scala.collection.immutable.ListMap

object wordcount {
  def main (args: Array[String]) {
  val conf = new SparkConf().setAppName("WordCount").setMaster("local")
  val sc = new SparkContext(conf)
  val textFile = sc.textFile("input.txt")
  val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
  val sorted=ListMap(counts.collect.sortWith(_._2 > _._2):_*)// sort in descending order based on
values
  println(sorted)
  for((k,v)<-sorted)
  {
    if(v>4)
    {
      print(k+",")
      print(v)
      println()
    }
  }
  }
}
```

```
21/06/13 10:45:41 INFO DAGScheduler: ResultStage 1 (main at <unknown>:0) finished in 0.110 s
21/06/13 10:45:41 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
21/06/13 10:45:41 INFO TaskSchedulerImpl: Killing all running tasks in stage 1: Stage finished
21/06/13 10:45:41 INFO DAGScheduler: Job 0 finished: main at <unknown>:0, took 0.823276 s
ListMap(Hello -> 6, Test -> 5, Hadoop -> 3, is -> 2, This -> 2, test -> 2, The -> 1, a -> 1, bye. -> 1, to -> 1, see -> 1, World
Hello,6
Test,5
21/06/13 10:45:41 INFO SparkContext: Invoking stop() from shutdown hook
21/06/13 10:45:41 INFO SparkUI: Stopped Spark web UI at http://LAPTOP-JG329ESD:4041
21/06/13 10:45:41 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
21/06/13 10:45:41 INFO MemoryStore: MemoryStore cleared
21/06/13 10:45:41 INFO BlockManager: BlockManager stopped
21/06/13 10:45:41 INFO BlockManagerMaster: BlockManagerMaster stopped
21/06/13 10:45:41 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
```