VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS
(20CS6PEBDA)

Submitted by

RAMYA RAMESH (1BM19CS227)

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 **RAMYA RAMESH (1BM19CS227),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 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.

Dr. Shyamala G
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

`

Index Sheet

Course Outcome

CO1	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO2	Analyze the Big Data and obtain insight using data analytics mechanisms
CO3	Design and implement Big data applications by applying NoSQL, Hadoop or Spark

LAB1 : Cassandra DB operation on Employee

Question:

1. Create a keyspace by name Employee

```
cqlsh> create keyspace Employee_227 with replication =
{'class':'SimpleStrategy','replication_factor':1};
cqlsh> use Employee_227;
```

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_227> create table Employee_info(Emp_Id int, Emp_Name text,
Designation text, Date_of_Joining date, Salary double, Dept_Name text, PRIMARY
KEY((Emp_Id), Salary)) WITH CLUSTERING ORDER BY (Salary ASC);
cqlsh:employee_227> describe table Employee_Info;
```

```
CREATE TABLE employee_227.employee_info (
    emp_id int,
    salary double,
    date_of_joining date,
    dept_name text,
    designation text,
    emp_name text,
    PRIMARY KEY (emp_id, salary)
) WITH CLUSTERING ORDER BY (salary ASC)
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

3. Insert the values into the table in batch

```
cqlsh:employee_227> begin batch
               ... insert into Employee_info(Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) values(123,'Hari','HR','2001-02-21',2500000,'HR');
               ... insert into Employee_info(Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) values(121,'Rohan','Manager','1999-11-15',3400000,'COE');
               ... insert into Employee_info(Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) values(125,'Ramesh','CEO','2005-09-06',200000000,'Admin');
               ... insert into Employee_info(Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) values(122,'Yadav','SDE','2014-05-28',1600000,'IT');
               ... apply batch;
cqlsh:employee_227> select * from Employee_info;

 emp_id | salary  | date_of_joining | dept_name | designation | emp_name
--------+---------+-----------------+-----------+-------------+----------
    123 | 2.5e+06 |      2001-02-21 |        HR |          HR |     Hari
    125 |   2e+08 |      2005-09-06 |     Admin |         CEO |   Ramesh
    122 | 1.6e+06 |      2014-05-28 |        IT |         SDE |    Yadav
    121 | 3.4e+06 |      1999-11-15 |       COE |     Manager |    Rohan

(4 rows)
```

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

```
cqlsh:employee_227> update Employee_info set Emp_Name='Mohan', Dept_Name='Marketing' where Emp_Id=121 and Salary=3400000;
cqlsh:employee_227> select * from Employee_info;

 emp_id | salary  | date_of_joining | dept_name | designation | emp_name
--------+---------+-----------------+-----------+-------------+----------
    123 | 2.5e+06 |      2001-02-21 |        HR |          HR |     Hari
    125 |   2e+08 |      2005-09-06 |     Admin |         CEO |   Ramesh
    122 | 1.6e+06 |      2014-05-28 |        IT |         SDE |    Yadav
    121 | 3.4e+06 |      1999-11-15 | Marketing |     Manager |    Mohan

(4 rows)
```

5. Sort the details of Employee records based on salary

```
cqlsh:employee_227> paging off;
Disabled Query paging.
cqlsh:employee_227> select * from Employee_info where emp_id in
(121,122,123,125) order by salary DESC allow filtering;

 emp_id | salary  | date_of_joining | dept_name | designation | emp_name
--------+---------+-----------------+-----------+-------------+----------
    125 |   2e+08 |      2005-09-06 |     Admin |         CEO |   Ramesh
    121 | 3.4e+06 |      1999-11-15 | Marketing |     Manager |    Mohan
    123 | 2.5e+06 |      2001-02-21 |        HR |          HR |     Hari
    122 | 1.6e+06 |      2014-05-28 |        IT |         SDE |    Yadav
(4 rows)
```

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 227> alter table Employee Info add Projects set<text>;
```

7. Update the altered table to add project names.

```
cqlsh:employee_227> update Employee_info set Projects={'SEO','Return on Investment Strategies'} where Emp_Id = 121 and Salary=3400000;
cqlsh:employee_227> update Employee_info set Projects={'Machine Learning', 'Web Development', 'Mobile App Development'} where Emp_Id = 122 and Salary=1600000;
cqlsh:employee_227> update Employee_info set Projects={'Risk assessment','Analytics','Cloud Computing'} where Emp_Id = 125 and Salary=200000000;
cqlsh:employee_227> update Employee_info set Projects={'Performance Appraisal System','Diversity Management','Grievance Handling'} where Emp_Id = 123 and Salary=2500000;
cqlsh:employee_227> select * from Employee_info;

 emp_id | salary  | date_of_joining | dept_name | designation | emp_name | projects
--------+---------+-----------------+-----------+-------------+----------+-----------------------------------------------------------------------------
    123 | 2.5e+06 |      2001-02-21 |        HR |          HR |     Hari | {'Diversity Management', 'Grievance Handling', 'Performance Appraisal System'}
    125 |   2e+08 |      2005-09-06 |     Admin |         CEO |   Ramesh |                           {'Analytics', 'Cloud Computing', 'Risk assessment'}
    122 | 1.6e+06 |      2014-05-28 |        IT |         SDE |    Yadav |             {'Machine Learning', 'Mobile App Development', 'Web Development'}
    121 | 3.4e+06 |      1999-11-15 | Marketing |     Manager |    Mohan |                                     {'Return on Investment Strategies', 'SEO'}

(4 rows)
```

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

```
cqlsh:employee_227> insert into Employee_info(Emp_Id, Emp_Name, Designation,
Date_of_Joining, Salary, Dept_Name,Projects)
values(124,'Karthik','Intern','2021-07-19',50000,'IT',{'CyberSecurity','Data
Science','Ethical Hacking'})using ttl 15;
cqlsh:employee_227> select * from employee_info;
```

```
 emp_id | salary  | date_of_joining | dept_name | designation | emp_name | projects
--------+---------+-----------------+-----------+-------------+----------+-----------------------------------------------------------------------------
    123 | 2.5e+06 |      2001-02-21 |        HR |          HR |     Hari | {'Diversity Management', 'Grievance Handling', 'Performance Appraisal System'}
    125 |   2e+08 |      2005-09-06 |     Admin |         CEO |   Ramesh |                           {'Analytics', 'Cloud Computing', 'Risk assessment'}
    122 | 1.6e+06 |      2014-05-28 |        IT |         SDE |    Yadav |             {'Machine Learning', 'Mobile App Development', 'Web Development'}
    121 | 3.4e+06 |      1999-11-15 | Marketing |     Manager |    Mohan |                                     {'Return on Investment Strategies', 'SEO'}
    124 |   50000 |      2021-07-19 |        IT |      Intern |  Karthik |                          {'CyberSecurity', 'Data Science', 'Ethical Hacking'}

(5 rows)
cqlsh:employee_227> select ttl(Emp_Name) from Employee_Info where Emp_Id=124;

 ttl(emp_name)
---------------
             9

(1 rows)
cqlsh:employee_227> select ttl(Emp_Name) from Employee_Info where Emp_Id=124;

 ttl(emp_name)
---------------
             4

(1 rows)
```

LAB2 : Cassandra DB operation on Library

Question:

1. Create a keyspace by name Library

```
C:\Users\Ramya>cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.8 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> create keyspace library with replication={'class':'SimpleStrategy','replication_factor':1}
   ... ;
cqlsh> describe keyspaces;

system_schema  system_auth  system  library  system_distributed  system_traces

cqlsh> use library;
```

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

```
cqlsh:library> create table libinfo(stud_id int, counter_val counter, studname text,
bookname text, book_id int, issue_date date, PRIMARY KEY(stud_id, studname, bookname,
book_id, issue_date));
```

3. Insert the values into the table in batch

```
cqlsh:library> update libinfo set counter_val = counter_val + 1 where stud_id=110 and studname='Ramya' and bookname='DBMS' and book_id=121 and issue_date='2022-05-06';
cqlsh:library> update libinfo set counter_val = counter_val + 1 where stud_id=112 and studname='Shobha' and bookname='BDA' and book_id=133 and issue_date='2022-04-13';
cqlsh:library> update libinfo set counter_val = counter_val + 1 where stud_id=134 and studname='Ramesh' and bookname='Discrete Math' and book_id=257 and issue_date='2021-12-07';
cqlsh:library> update libinfo set counter_val = counter_val + 1 where stud_id=226 and studname='Anamika' and bookname='ML' and book_id=349 and issue_date='2021-11-27';
cqlsh:library> select * from libinfo;

 stud_id | studname | bookname      | book_id | issue_date | counter_val
---------+----------+---------------+---------+------------+-------------
     110 |    Ramya |          DBMS |     121 | 2022-05-06 |           1
     134 |   Ramesh | Discrete Math |     257 | 2021-12-07 |           1
     112 |   Shobha |           BDA |     133 | 2022-04-13 |           1
     226 |  Anamika |            ML |     349 | 2021-11-27 |           1

(4 rows)
```

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

```
cqlsh:library> update libinfo set counter_val = counter_val + 1 where stud_id=112 and studname='Shobha' and bookname='BDA' and book_id=133 and issue_date='2022-04-13';
cqlsh:library> select * from libinfo;

 stud_id | studname | bookname      | book_id | issue_date | counter_val
---------+----------+---------------+---------+------------+-------------
     110 |    Ramya |          DBMS |     121 | 2022-05-06 |           1
     134 |   Ramesh | Discrete Math |     257 | 2021-12-07 |           1
     112 |   Shobha |           BDA |     133 | 2022-04-13 |           2
     226 |  Anamika |            ML |     349 | 2021-11-27 |           1

(4 rows)
```

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

```
cqlsh:library> select * from libinfo where counter_val=2 allow filtering;

 stud_id | studname | bookname | book_id | issue_date | counter_val
---------+----------+----------+---------+------------+-------------
     112 |   Shobha |      BDA |     133 | 2022-04-13 |           2

(1 rows)
```

6. Export the created column to a csv file

```
cqlsh:library> copy libinfo(stud_id,counter_val,studname, bookname, book_id, issue_date) to 'C:\Users\Ramya\libdata';
Using 7 child processes

Starting copy of library.libinfo with columns [stud_id, counter_val, studname, bookname, book_id, issue_date].
Processed: 4 rows; Rate:       2 rows/s; Avg. rate:       1 rows/s
4 rows exported to 1 files in 3.118 seconds.
```

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

```
cqlsh:library> create table libinfo_1(stud_id int, counter_val counter, studname text,
bookname text, book_id int, issue_date date, PRIMARY KEY(stud_id, studname, bookname,
book_id, issue_date));
cqlsh:library> copy libinfo_1(stud_id,counter_val,studname, bookname, book_id, issue_date) from 'C:\Users\Ramya\libdata';
Using 7 child processes

Starting copy of library.libinfo_1 with columns [stud_id, counter_val, studname, bookname, book_id, issue_date].
Processed: 4 rows; Rate:       1 rows/s; Avg. rate:       1 rows/s
4 rows imported from 1 files in 4.201 seconds (0 skipped).
cqlsh:library> select * from libinfo_1;

 stud_id | studname | bookname      | book_id | issue_date | counter_val
---------+----------+---------------+---------+------------+-------------
     110 |    Ramya |          DBMS |     121 | 2022-05-06 |           1
     134 |   Ramesh | Discrete Math |     257 | 2021-12-07 |           1
     112 |   Shobha |           BDA |     133 | 2022-04-13 |           2
     226 |  Anamika |            ML |     349 | 2021-11-27 |           1

(4 rows)
```

LAB3 : MongoDB CRUD Demonstration

Input:

```
use ramyar;

db;

show dbs;

db.createCollection("student")

db.student.insert({_id:0,StudName:"Tasmiya",Sem:"VI",Hobbies:"Singing"});
db.student.insert({_id:1,StudName:"Trisha",Sem:"IV",Hobbies:"Carrom"});
db.student.insert({_id:2,StudName:"Anagha",Sem:"V",Hobbies:"Drawing"});
db.student.insert({_id:3,StudName:"Ramya",Sem:"VI",Hobbies:"Violin"});

db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.update({_id:1},{$set:{Hobbies:"Cricket"}},{upsert:true});
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.update({StudName:'Rahul'},{$set:{Location:'BMS'}},{upsert:true});
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.update({StudName:'Meenakshi'},{$set:{Location:null}});
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.update({StudName:'Rahul'},{$unset:{Location:'BMS'}});
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({StudName:'Tasmiya'}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({Sem:'VI'}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({StudName:{$ne:'Ramya'},Sem:{$ne:'V'}}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({Sem:{$lte:'V'}}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({StudName:/^T/}).pretty();
```

```
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({StudName:/a$/}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({StudName:/r|R/}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find({Hobbies:{$in:['Drawing','Violin']}}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find().sort({Sem:1}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.find().sort({StudName:-1}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.save({StudName:'Meenakshi',Sem:'IV'});
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();

db.student.count();
db.student.count({Sem:'VI'})


db.student.find({Sem:'IV'}).limit(1).pretty();

db.student.find().skip(2).pretty();

db.student.remove({StudName:'Rahul'}).pretty();
db.student.find({},{_id:0,StudName:1,Sem:1,Hobbies:1,Location:1}).pretty();


db.student.drop();
```

Output:

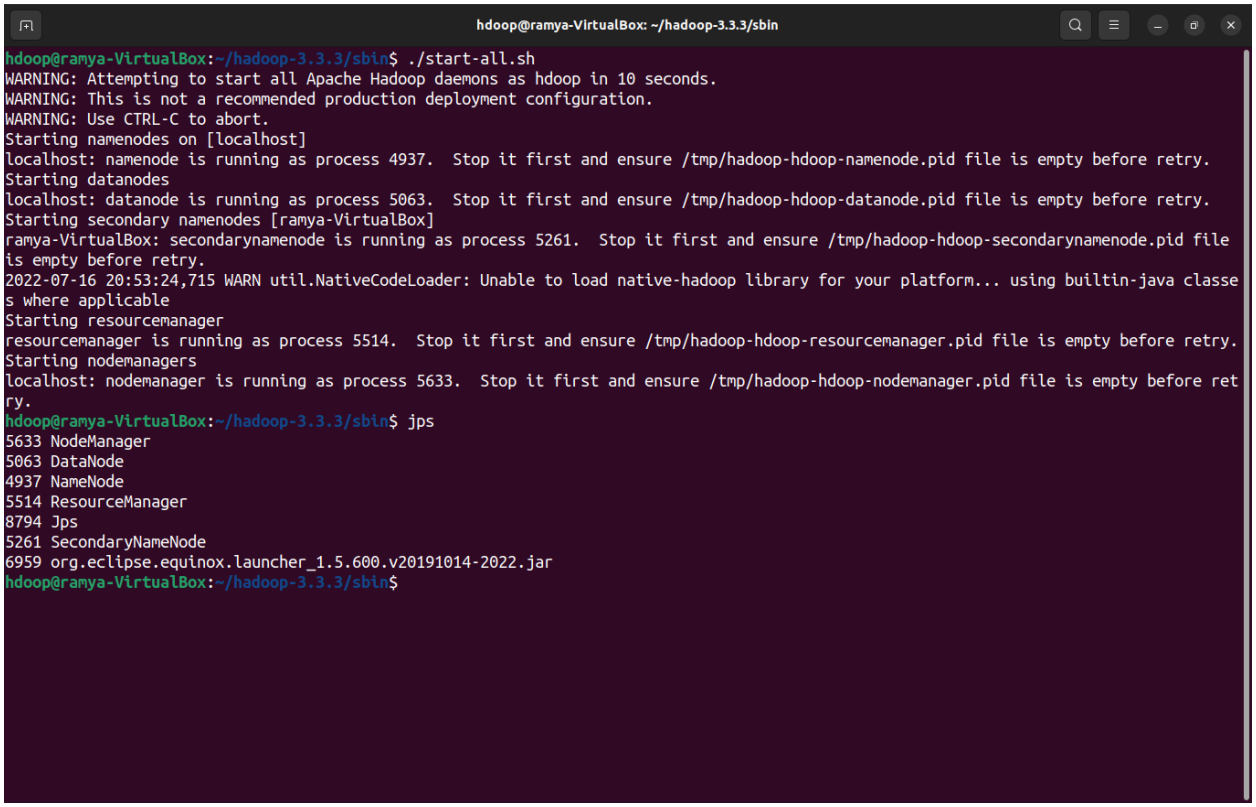
```
switched to db ramyar
ramyar
admin      0.000GB
config     0.000GB
harryKart  0.000GB
local      0.000GB
school     0.000GB
{ "ok" : 1 }
```

```
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
WriteResult({ "nInserted" : 1 })
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Carrom" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 1,
        "nModified" : 0,
        "_id" : ObjectId("62d4c66a9e01af7adee22d59")
})
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul", "Location" : "BMS" }
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul", "Location" : "BMS" }
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 0, "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 0, "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "_id" : 3, "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 0, "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
```

{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "_id" : ObjectId("62d4c66a9e01af7adee22d59"), "StudName" : "Rahul" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "_id" : 2, "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 0, "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 0, "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "_id" : 2, "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "_id" : 3, "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "_id" : 3, "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "_id" : ObjectId("62d4c66a9e01af7adee22d59"), "StudName" : "Rahul" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 2, "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "_id" : 3, "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : ObjectId("62d4c66a9e01af7adee22d59"), "StudName" : "Rahul" }
{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "_id" : 2, "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "_id" : 0, "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "_id" : 3, "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }

```
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "_id" : 0, "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "_id" : 3, "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "_id" : ObjectId("62d4c66a9e01af7adee22d59"), "StudName" : "Rahul" }
{ "_id" : 2, "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
WriteResult({ "nInserted" : 1 })
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Rahul" }
{ "StudName" : "Meenakshi", "Sem" : "IV" }
6
2
{ "_id" : 1, "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "_id" : 2, "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "_id" : 3, "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "_id" : ObjectId("62d4c66a9e01af7adee22d59"), "StudName" : "Rahul" }
{
        "_id" : ObjectId("62d4c66a7a576a20603d9453"),
        "StudName" : "Meenakshi",
        "Sem" : "IV"
}
WriteResult({ "nRemoved" : 1 })
{ "StudName" : "Tasmiya", "Sem" : "VI", "Hobbies" : "Singing" }
{ "StudName" : "Trisha", "Sem" : "IV", "Hobbies" : "Cricket" }
{ "StudName" : "Anagha", "Sem" : "V", "Hobbies" : "Drawing" }
{ "StudName" : "Ramya", "Sem" : "VI", "Hobbies" : "Violin" }
{ "StudName" : "Meenakshi", "Sem" : "IV" }
true
```

LAB4 : Hadoop Installation Screenshot



LAB5 : Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)

hduser@bmsce-Precision-T1700:~$ start-all.sh

This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh

WARNING: An illegal reflective access operation has occurred

WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()

WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil

WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

WARNING: All illegal access operations will be denied in a future release

Starting namenodes on [localhost]

hduser@localhost's password:

localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-Precision-T1700.out

localhost: WARNING: An illegal reflective access operation has occurred

localhost: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()

localhost: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil

localhost: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

localhost: WARNING: All illegal access operations will be denied in a future release

hduser@localhost's password:

localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-Precision-T1700.out

localhost: WARNING: An illegal reflective access operation has occurred

localhost: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()

localhost: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil

localhost: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

localhost: WARNING: All illegal access operations will be denied in a future release

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

0.0.0.0: WARNING: An illegal reflective access operation has occurred

0.0.0.0: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()

0.0.0.0: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil

0.0.0.0: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

0.0.0.0: WARNING: All illegal access operations will be denied in a future release

WARNING: An illegal reflective access operation has occurred

WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()

WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil

WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

WARNING: All illegal access operations will be denied in a future release

starting yarn daemons

starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-Precision-T1700.out

WARNING: An illegal reflective access operation has occurred

WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()

WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil

WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

WARNING: All illegal access operations will be denied in a future release

hduser@localhost's password:

localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-bmsce-Precision-T1700.out

localhost: WARNING: An illegal reflective access operation has occurred

localhost: WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()

localhost: WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil

localhost: WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

localhost: WARNING: All illegal access operations will be denied in a future release

hduser@bmsce-Precision-T1700:~$ jps

8386 NodeManager

7654 DataNode

7879 SecondaryNameNode

7463 NameNode

9143 Jps

8044 ResourceManager

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /227new

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /
Found 11 items
drwxr-xr-x   - hduser supergroup          0 2022-06-01 10:12 /1bm19cs186
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:27 /227new
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:20 /Copy-Secure
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:06 /Sharan
drwxr-xr-x   - hduser supergroup          0 2022-06-03 14:57 /bda
drwxr-xr-x   - hduser supergroup          0 2022-06-01 09:32 /firstlab
drwxr-xr-x   - hduser supergroup          0 2022-06-01 09:32 /lab
drwxr-xr-x   - hduser supergroup          0 2022-06-01 14:59 /nothing
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:27 /something
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 -put /home/hduser/Desktop/Welcome.txt
/227new/WC.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227new
Found 1 items
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:33 /227new/WC.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /227new/WC.txt
Hello! Welcome

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal
/home/hduser/Desktop/Welcome.txt /227new/WC1.txt
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227new
Found 2 items
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:33 /227new/WC.txt
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:37 /227new/WC1.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /227new/WC1.txt
Hello! Welcome

hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /227new/WC.txt
/home/hduser/Downloads/WWC.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /
Found 11 items
drwxr-xr-x   - hduser supergroup          0 2022-06-01 10:12 /1bm19cs186
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:37 /227new
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:20 /Copy-Secure
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:06 /Sharan
drwxr-xr-x   - hduser supergroup          0 2022-06-03 14:57 /bda
drwxr-xr-x   - hduser supergroup          0 2022-06-01 09:32 /firstlab
drwxr-xr-x   - hduser supergroup          0 2022-06-01 09:32 /lab
drwxr-xr-x   - hduser supergroup          0 2022-06-01 14:59 /nothing
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:27 /something
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 -getmerge /227new/WC.txt /227new/WC1.txt
/home/hduser/Desktop/Merge.txt
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /227new/
# file: /227new
# owner: hduser
# group: supergroup
user::rwx
group::r-x
other::r-x

hduser@bmsce-Precision-T1700:~$ sudo nano abc.txt
[sudo] password for hduser:

hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/abc.txt /227new/name.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227new
Found 3 items
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:33 /227new/WC.txt
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:37 /227new/WC1.txt
-rw-r--r--   1 hduser supergroup         20 2022-06-04 09:51 /227new/name.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /227new/name.txt
This is Ramya here!

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /227new/name.txt
/home/hduser/Desktop

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /
Found 11 items
drwxr-xr-x   - hduser supergroup          0 2022-06-01 10:12 /1bm19cs186
drwxr-xr-x   - hduser supergroup          0 2022-06-04 09:51 /227new
```

```
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:20 /Copy-Secure
drwxr-xr-x   - hduser supergroup          0 2022-06-03 12:06 /Sharan
drwxr-xr-x   - hduser supergroup          0 2022-06-03 14:57 /bda
drwxr-xr-x   - hduser supergroup          0 2022-06-01 09:32 /firstlab
drwxr-xr-x   - hduser supergroup          0 2022-06-01 09:32 /lab
drwxr-xr-x   - hduser supergroup          0 2022-06-01 14:59 /nothing
drwxr-xr-x   - hduser supergroup          0 2022-06-01 15:27 /something
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 -mv /227new /227newer

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227newer
Found 3 items
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:33 /227newer/WC.txt
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:37 /227newer/WC1.txt
-rw-r--r--   1 hduser supergroup         20 2022-06-04 09:51 /227newer/name.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /227newer/  /227new

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227new
Found 3 items
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:58 /227new/WC.txt
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:58 /227new/WC1.txt
-rw-r--r--   1 hduser supergroup         20 2022-06-04 09:58 /227new/name.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227newer
Found 3 items
```

```
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:33 /227newer/WC.txt
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:37 /227newer/WC1.txt
-rw-r--r--   1 hduser supergroup         20 2022-06-04 09:51 /227newer/name.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /227newer/name.txt  /227new
cp: `/227new/name.txt': File exists

hduser@bmsce-Precision-T1700:~$ sudo nano hello.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hduser/hello.txt /227newer/hello.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227newer
Found 4 items
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:33 /227newer/WC.txt
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:37 /227newer/WC1.txt
-rw-r--r--   1 hduser supergroup         13 2022-06-04 10:02 /227newer/hello.txt
-rw-r--r--   1 hduser supergroup         20 2022-06-04 09:51 /227newer/name.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /227newer/hello.txt
hi hello bye

hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /227newer/hello.txt  /227new

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227newer
Found 4 items
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:33 /227newer/WC.txt
-rw-r--r--   1 hduser supergroup         15 2022-06-04 09:37 /227newer/WC1.txt
-rw-r--r--   1 hduser supergroup         13 2022-06-04 10:02 /227newer/hello.txt
```

-rw-r--r-- 1 hduser supergroup 20 2022-06-04 09:51 /227newer/name.txt

hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /227new

Found 4 items

-rw-r--r-- 1 hduser supergroup 15 2022-06-04 09:58 /227new/WC.txt

-rw-r--r-- 1 hduser supergroup 15 2022-06-04 09:58 /227new/WC1.txt

-rw-r--r-- 1 hduser supergroup 13 2022-06-04 10:03 /227new/hello.txt

-rw-r--r-- 1 hduser supergroup 20 2022-06-04 09:58 /227new/name.txt

hduser@bmsce-Precision-T1700:~$

LAB6 : Create a Map Reduce program to
a) find average temperature for each year from the NCDC data set.
b) find the mean max temperature for every month

a) Average temperature

AverageDriver.java

```
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.java

```
package temp;

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

```
import org.apache.hadoop.mapreduce.Mapper;

public class AverageMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
  public static final int MISSING = 9999;

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

AverageReducer.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));
  }
}
```

OUTPUT:

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

b) Mean Max temperature

TempDriver.java

```
package temperatureMax;

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);
  }
}
```

TempMapper.java

```
packagetemperatureMax;
```

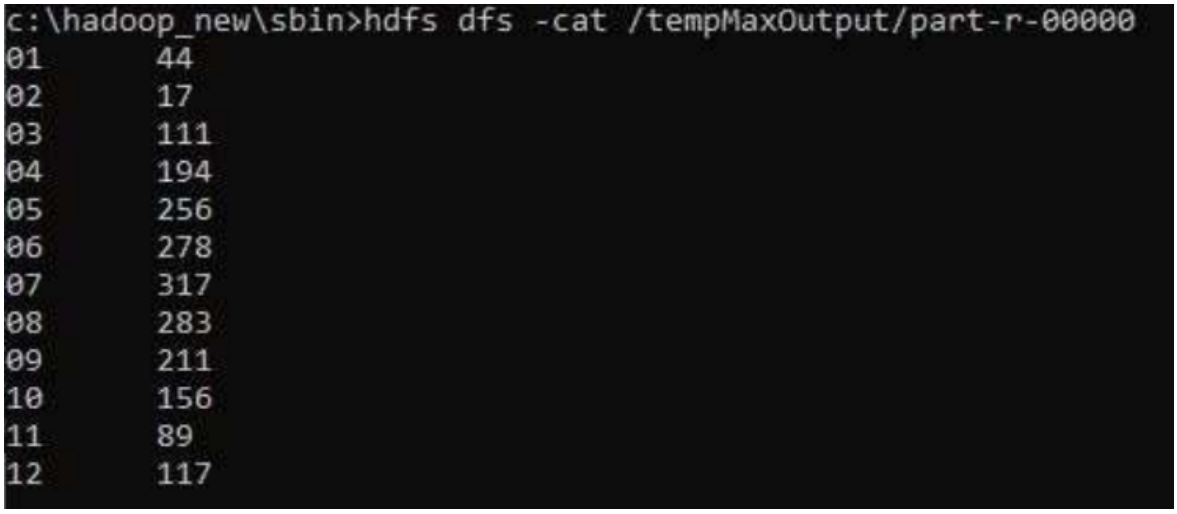
```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MeanMaxMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
  public static final int MISSING = 9999;

  public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
    int temperature;
    String line = value.toString();
    String month = line.substring(19, 21);
    if (line.charAt(87) == '+') {
      temperature = Integer.parseInt(line.substring(88, 92));
    } else {
      temperature = Integer.parseInt(line.substring(87, 92));
    }
    String quality = line.substring(92, 93);
    if (temperature != 9999 && quality.matches("[01459]"))
      context.write(new Text(month), new IntWritable(temperature));
  }
}
```

TempReducer.java

```
package temperatureMax;

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));
  }
}
```

OUTPUT:



LAB7 : For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top N maximum occurrences of words.

Driver-TopN.class

```
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);
  }
}
```

TopNMapper.class

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

```
import org.apache.hadoop.mapreduce.Mapper;

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

  private Text word = new Text();

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

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

TopNCombiner.class

```
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));
  }
}
```

TopNReducer.class

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;
```

```
public class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
  private Map<Text, IntWritable> countMap = new HashMap<>();

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

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

MiscUitls.java

```
package utils;

import java.util.*;

public class MiscUtils {

/**

* sorts the map by values. Taken from:

* http://javarevisited.blogspot.it/2012/12/how-to-sort-hashmap-java-by-key-and-value.html

*/

public static <K extends Comparable, V extends Comparable> Map<K, V>
sortByValues(Map<K, V> map) {

List<Map.Entry<K, V>> entries = new LinkedList<Map.Entry<K, V>>(map.entrySet());

Collections.sort(entries, new Comparator<Map.Entry<K, V>>() {

@Override
```

```
public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) {
return o2.getValue().compareTo(o1.getValue());
}
});
//LinkedHashMap will keep the keys in the order they are inserted

//which is currently sorted on natural ordering

Map<K, V> sortedMap = new LinkedHashMap<K, V>();
for (Map.Entry<K, V> entry : entries) {
sortedMap.put(entry.getKey(), entry.getValue());
}
return sortedMap;
}

}
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /output100
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2022-06-25 10:17 /output100/_SUCCESS
-rw-r--r--   1 hduser supergroup        125 2022-06-25 10:17 /output100/part-r-00000
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat /output100/part-r-00000
bms     4
in      2
i       2
is      2
am      2
college 2
from    2
department      1
hi      1
road    1
```

LAB8 : Create a Map Reduce program to demonstrate 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

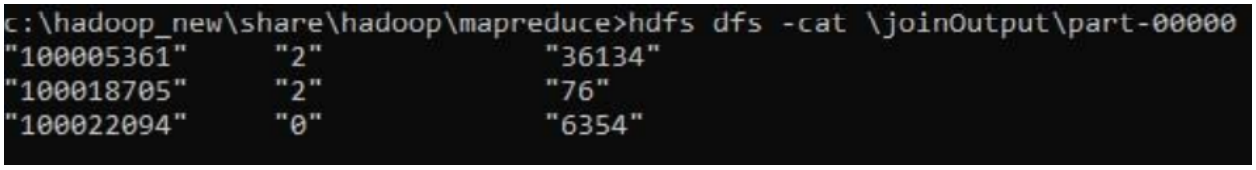
```
import java.io.*;

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);
}
}  }
```

OUTPUT:



LAB9 : Program to print word count on scala shell and print “Hello world” on scala IDE

```
(base) bmsce@bmsce-Precision-T1700:~$ spark-shell
22/07/02 09:33:57 WARN Utils: Your hostname, bmsce-Precision-T1700 resolves to a loopback address: 127.0.1.1; using 10.124.7.83 instead (on interface enp1s0)
22/07/02 09:33:57 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
22/07/02 09:33:58 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://10.124.7.83:4040
Spark context available as 'sc' (master = local[*], app id = local-1656734644119).
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_232)
Type in expressions to have them evaluated.
Type :help for more information.

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

scala> data.collect;
res0: Array[String] = Array(hi how are you, 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;
res1: Array[String] = Array(hi, how, are, you, 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;
res2: Array[(String, Int)] = Array((hi,1), (how,1), (are,1), (you,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), (family,1), (hi,1), (how,5), (is,4), (job,1), (sister,1), (you,1), (your,4))
```

Run New Format Clear Messages Worksheet Download Embed

```scala
object Hello {
    def main(args: Array[String]) = {
        println("Hello, world")
    }
}
```

Console (F3)

```
Hello, world
```

```
Microsoft Windows [Version 10.0.22000.795]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ramya>cd scala practice

C:\Users\Ramya\scala practice>scalac Hello.scala

C:\Users\Ramya\scala practice>scala Hello
Hello, world

C:\Users\Ramya\scala practice>
```

LAB10 : 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

```
scala> val textFile=sc.textFile("/home/bmsce/Desktop/sparkdata.txt")
textFile: org.apache.spark.rdd.RDD[String] = /home/bmsce/Desktop/sparkdata.txt MapPartitionsRDD[6] at textFile at <console>:24

scala> val counts=textFile.flatMap(line=>line.split(" ")).map(word=>(word,1)).reduceByKey(_+_);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[9] 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(bms -> 5, college -> 4, of -> 2, university -> 1, evening -> 1, women's -> 1, technological -> 1, engineering -> 1, architecture -> 1, id -> 1, visweswariah -> 1)

scala> println(sorted)
Map(bms -> 5, college -> 4, of -> 2, university -> 1, evening -> 1, women's -> 1, technological -> 1, engineering -> 1, architecture -> 1, id -> 1, visweswariah -> 1)

scala> for((k,v)<-sorted)
     | {
     | if(v>4)
     | {
     | print(k+",")
     | print(v)
     | println()
     | }
     | }
bms,5

scala>
```