

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

## BIG DATA ANALYTICS (20CS6PEBDA)

*Submitted by*

POOJA K(1BM19CS111)

*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 P O O J A K (1BM19CS111), 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 CHOUDURY  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

## Index Sheet

Sl. No.	Experiment Title	Page No.
1	Employee Database	5
2	Library	7
3	Mongo (CRUD)	8
4	Hadoop installation	11
5	HDFS Commands	12
6	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	15
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.	20
8	Create a Map Reduce program to demonstrating join operation	23
9	Program to print word count on scala shell and print "Hello world" on scala IDE	28
10	Using RDD and FlatMap 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	29

## 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

	Spark
--	-------

# Lab 1

**Program 1. Perform the following DB operations using Cassandra.**

1. Create a key space by name Employee
2. Create a column family by name Employee-Info with attributes Emp\_Id Primary Key, Emp\_Name, Designation, Date\_of\_Joining, Salary, Dept\_Name
3. Insert the values into the table in batch
4. Update Employee name and Department of Emp-Id 121
5. Sort the details of Employee records based on salary
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.
7. Update the altered table to add project names.
- 8 Create a TTL of 15 seconds to display the values of Employees.

1.

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

```
cqlsh> DESCRIBE KEYSPACES
```

```
USE HELP FOR HELP.
cqlsh> CREATE KEYSPACE Employee WITH REPLICATION = {
... 'class' : 'SimpleStatergy',
... 'replication_factor': 1
... };
ConfigurationException: Unable to find replication strategy class 'org.apache.cassandra.locator.SimpleStatergy'
cqlsh> CREATE KEYSPACE Employee WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor': 1 };
cqlsh> DESCRIBE KEYSPACES

system_schema system_auth system system_distributed employee system_traces

cqlsh> USE Employee
... :
```

system\_schema system\_auth system system\_distributed employee system\_traces

2.CREATE TABLE Employee\_info(

```
... emp_id int PRIMARY KEY,
... emp_name text,
... designation text,
... date_of_joining timestamp,
... salary double,
... dept_name text
... );
```

```
bmsce@bmsce-Precision-T1700: ~/cassandra/apache-cassandra-3.11.0/bin

CREATE TABLE employee.employee_info (
  emp_id int PRIMARY KEY,
  date_of_joining timestamp,
  dept_name text,
  designation text,
  emp_name text,
  salary double
) WITH 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) BEGIN BATCH

```
... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (1,'Pooja','Manager','2023-09-12',60000,'Technical')

... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (2,'Prena','Supervisor','2023-10-12',55000,'Technical')

... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (3,'Vijay','Employee','2023-10-30',45000,'HR')

... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (4,'Prajwal','General manager','2023-12-30',55000,'Research')

... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (5,'Varun','Manager','2023-12-30',55000,'Research')

... APPLY BATCH;
```

```
cqlsh:employee> BEGIN BATCH
... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (1,'Pooja','Manager','2023-09-12',60000,'Technical')
... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (2,'Prena','Supervisor','2023-10-12',55000,'Technical')
... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (3,'Vijay','Employee','2023-10-30',45000,'HR')
... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (4,'Prajwal','General manager','2023-12-30',55000,'Research')
... INSERT INTO Employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
... VALUES (5,'Varun','Manager','2023-12-30',55000,'Research')
... APPLY BATCH;
cqlsh:employee> SELECT * FROM eMPLOYEE_INFO
... ;
```

emp_id	date_of_joining	dept_name	designation	emp_name	salary
5	2023-12-29 18:30:00.000000+0000	Research	Manager	Varun	55000
1	2023-09-11 18:30:00.000000+0000	Technical	Manager	Pooja	60000
2	2023-10-11 18:30:00.000000+0000	Technical	Supervisor	Prena	55000
4	2023-12-29 18:30:00.000000+0000	Research	General manager	Prajwal	55000
3	2023-10-29 18:30:00.000000+0000	HR	Employee	Vijay	45000

### 4). SELECT \* FROM eMPLOYEE\_INFO

```
... ;
```

```
emp_id | date_of_joining          | dept_name | designation    | emp_name | salary
```

```
-----+-----+-----+-----+-----+-----
```

```

5 | 2023-12-29 18:30:00.000000+0000 | Research | Manager | Varun | 55000
1 | 2023-09-11 18:30:00.000000+0000 | Technical | Manager | Pooja | 60000
2 | 2023-10-11 18:30:00.000000+0000 | Technical | Supervisor | Prema | 55000
4 | 2023-12-29 18:30:00.000000+0000 | Research | General manager | Prajwal | 55000
3 | 2023-10-29 18:30:00.000000+0000 | HR | Employee | Vijay | 45000

```

5.)

UPDATE Employee\_info SET emp\_name = 'Tarun',dept\_name='Sales' WHERE emp\_id=5;

cqlsh:employee> SELECT \* FROM eMPLOYEE\_INFO ;

```

emp_id | date_of_joining          | dept_name | designation    | emp_name | salary
-----+-----+-----+-----+-----+-----
5 | 2023-12-29 18:30:00.000000+0000 | Sales | Manager | Tarun | 55000
1 | 2023-09-11 18:30:00.000000+0000 | Technical | Manager | Pooja | 60000
2 | 2023-10-11 18:30:00.000000+0000 | Technical | Supervisor | Prema | 55000
4 | 2023-12-29 18:30:00.000000+0000 | Research | General manager | Prajwal | 55000
3 | 2023-10-29 18:30:00.000000+0000 | HR | Employee | Vijay | 45000

```

```

(5 rows)
cqlsh:employee> UPDATE Employee_info SET emp_name = 'Tarun',dept_name='Sales' WHERE emp_id=5;
cqlsh:employee> SELECT * FROM eMPLOYEE_INFO ;

emp_id | date_of_joining          | dept_name | designation    | emp_name | salary
-----+-----+-----+-----+-----+-----
5 | 2023-12-29 18:30:00.000000+0000 | Sales | Manager | Tarun | 55000
1 | 2023-09-11 18:30:00.000000+0000 | Technical | Manager | Pooja | 60000
2 | 2023-10-11 18:30:00.000000+0000 | Technical | Supervisor | Prema | 55000
4 | 2023-12-29 18:30:00.000000+0000 | Research | General manager | Prajwal | 55000
3 | 2023-10-29 18:30:00.000000+0000 | HR | Employee | Vijay | 45000

(5 rows)
cqlsh:employee> SELECT * FROM eMPLOYEE_INFO d[];

```

6)ALTER TABLE Employee\_info

... ADD project text;

```

... ADD project text;
cqlsh:employee> SELECT * FROM eMPLOYEE_INFO ;

emp_id | date_of_joining          | dept_name | designation    | emp_name | project | salary
-----+-----+-----+-----+-----+-----+-----
5 | 2023-12-29 18:30:00.000000+0000 | Sales | Manager | Tarun | null | 55000
1 | 2023-09-11 18:30:00.000000+0000 | Technical | Manager | Pooja | null | 60000
2 | 2023-10-11 18:30:00.000000+0000 | Technical | Supervisor | Prema | null | 55000
4 | 2023-12-29 18:30:00.000000+0000 | Research | General manager | Prajwal | null | 55000
3 | 2023-10-29 18:30:00.000000+0000 | HR | Employee | Vijay | null | 45000

(5 rows)
cqlsh:employee> begin batch
...

```

7)

begin batch

... update employee\_info set project = 'abc' where emp\_id=1

```
... update employee_info set project = 'dfc' where emp_id=2
... update employee_info set project = 'dfc' where emp_id=3
... update employee_info set project = 'xyz' where emp_id=4
... update employee_info set project = 'rqz' where emp_id=5
... ;
... apply batch;
```

```
cqlsh:employee> SELECT * FROM eMPLOYEE_INFO ;
```

emp_id	date_of_joining	dept_name	designation	emp_name	project	salary
5	2023-12-29 18:30:00.000000+0000	Sales	Manager	Tarun	rqz	55000
1	2023-09-11 18:30:00.000000+0000	Technical	Manager	Pooja	abc	60000
2	2023-10-11 18:30:00.000000+0000	Technical	Supervisor	Prema	dfc	55000
4	2023-12-29 18:30:00.000000+0000	Research	General manager	Prajwal	xyz	55000
3	2023-10-29 18:30:00.000000+0000	HR	Employee	Vijay	dfc	45000



## LAB 2

1 Create a key space by name Library

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

```
cqlsh> describe keyspace library;
```

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

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

```
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));
```

```
cqlsh:library> describe table library_info;
```

```
CREATE TABLE library.library_info (  
  stud_id int,  
  stud_name text,  
  book_name text,  
  book_id int,  
  date_of_issue timestamp,  
  counter_value counter,  
  PRIMARY KEY (stud_id, stud_name, book_name, book_id, date_of_issue)  
) WITH CLUSTERING ORDER BY (stud_name ASC, book_name ASC, book_id ASC, date_of_issue ASC)  
  AND additional_write_policy = '99p'  
  AND bloom_filter_fp_chance = 0.01  
  AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}  
  AND cdc = false  
  AND comment = ''  
  AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}  
  AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}  
  AND crc_check_chance = 1.0  
  AND default_time_to_live = 0  
  AND extensions = {}  
  AND gc_grace_seconds = 864000  
  AND max_index_interval = 2048  
  AND memtable_flush_period_in_ms = 0  
  AND min_index_interval = 128  
  AND read_repair = 'BLOCKING'  
  AND speculative_retry = '99p';
```

3. Insert the values into the table in batch

```
cqlsh:library> update library_info set counter_value=counter_value+1 where stud_id=1 and stud_name = 'Raj'  
and book_name='BDA' and book_id=200 and date_of_issue='2022-04-30';
```

```
cqlsh:library> update library_info set counter_value=counter_value+1 where stud_id=2 and stud_name = 'Ravi'  
and book_name='ADA' and book_id=100 and date_of_issue='2022-04-30';
```

```
cqlsh:library> update library_info set counter_value=counter_value+1 where stud_id=1 and stud_name = 'Raj'  
and book_name='BDA' and book_id=200 and date_of_issue='2022-05-30';
```

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

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

stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
1	Raj	BDA	200	2022-04-29 18:30:00.000000+0000	1
1	Raj	BDA	200	2022-05-29 18:30:00.000000+0000	1
2	Ravi	ADA	100	2022-04-29 18:30:00.000000+0000	1

(3 rows)

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

```
cqlsh:library> update library_info set counter_value=counter_value+1 where stud_id=1 and stud_name = 'Raj'
and book_name='BDA' and book_id=200 and date_of_issue='2022-04-30';
```

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

stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
1	Raj	BDA	200	2022-04-29 18:30:00.000000+0000	2
1	Raj	BDA	200	2022-05-29 18:30:00.000000+0000	1
2	Ravi	ADA	100	2022-04-29 18:30:00.000000+0000	1

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

stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
1	Raj	BDA	200	2022-04-29 18:30:00.000000+0000	2
1	Raj	BDA	200	2022-05-29 18:30:00.000000+0000	1
2	Ravi	ADA	100	2022-04-29 18:30:00.000000+0000	1

(3 rows)

5. Write a query to show that a student with id 1 has taken a book "BDA" 2

```
times.cqlsh:library> select counter_value from library_info where stud_id = 1;
```

counter_value
2
1

```
cqlsh:library> select counter_value from library_info where stud_id = 1;
```

```
counter_value
```

```
-----
```

```
2
```

```
1
```

```
(2 rows)
```

#### 6. Export the created column to a csv file

```
cqlsh:lab2_library> copy library_info(stud_id,stud_name,book_id,date_of_issue,counter_value)to 'lib.csv';  
Using 7 child processes
```

```
Starting copy of lab2_library.library_info with columns [stud_id, stud_name, book_id, date_of_issue, counter_v  
alue].
```

```
Processed: 2 rows; Rate:      9 rows/s; Avg. rate:      9 rows/s
```

```
2 rows exported to 1 files in 0.250 seconds.
```

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

Cassandra column family

```
cqlsh:library>truncate library_info;
```

```
cqlsh:library>copy library_info(stud_id,stud_name,book_id,date_of_issue,counter_value) from  
'lib.csv';
```

## Lab3

use studentdb switched

to db studentdb

```
db.createCollection("student_details")
```

```
{ "ok" : 1 }
```

```
db.student_details.insert({'name':'abc','rollno':1,'age':19,'contactno':9090909090,'email':'abc@lab.
```

```
com'}})
```

```
WriteResult({ "nInserted" : 1 })
```

```
db.student_details.insert({'name':'mno','rollno':2,'age':20,'contactno':9999900000,'email':'mno@lab.com'}})
```

```
WriteResult({ "nInserted" : 1 })
```

```
db.student_details.insert({'name':'xyz','rollno':3,'age':21,'contactno':9999911111,'email':'xyz@lab.com'}})
```

```
WriteResult({ "nInserted" : 1 })
```

```
db.student_details.find({})
```

```
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19, "contactno" : 9090909090, "email" : "abc@lab.com" }
```

```
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20, "contactno" : 9999900000, "email" : "mno@lab.com" }
```

```
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "xyz", "rollno" : 3, "age" : 21, "contactno" : 9999911111, "email" : "xyz@lab.com" }
```

```
db.student_details.update({'rollno':3},{ $set: {'email':'update@lab.com'}})
```

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

```
db.student_details.find({'rollno':3})
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "xyz", "rollno" : 3, "age" : 21,
"contactno" : 9999911111, "email" : "update@lab.com" }
```

```
db.student_details.update({'name':'xyz'},{$set:{'name':'pqr'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
db.student_details.find({'name':'pqr'})
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "pqr", "rollno" : 3, "age" : 21,
"contactno" : 9999911111, "email" : "update@lab.com" }
```

```
mongoexport --db studentdb --collection student_details --out E:\Desktop\sample.json
2021-05-22T10:43:30.687+0530   connected to: mongodb://localhost/
2021-05-22T10:43:31.026+0530   exported 3 records
```

```
db.getCollection('student_details').drop()
true
```

```
mongoimport --db studentdb --collection student_details --type=json --file=
E:\Desktop\sample.json
2021-05-22T10:46:49.898+0530   connected to: mongodb://localhost/ 2021-05-
22T10:46:50.044+0530   3 document(s) imported successfully. 0 document(s) failed to import.
```

```
db.student_details.find({})
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "pqr", "rollno" : 3, "age" : 21,
"contactno" : 9999911111, "email" : "update@lab.com" }
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19,
"contactno" : 9090909090, "email" : "abc@lab.com" }
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20,
"contactno" : 9999900000, "email" : "mno@lab.com" }
```

```
db.student_details.remove({age:{$gt:20}})
```

```
WriteResult({ "nRemoved" : 1 })
```

```
db.student_details.find({})
```

```
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19,
"contactno" : 9090909090, "email" : "abc@lab.com" }
```

```
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20,
"contactno" : 9999900000, "email" : "mno@lab.com" }
```

```
db.student_details.find({})
```

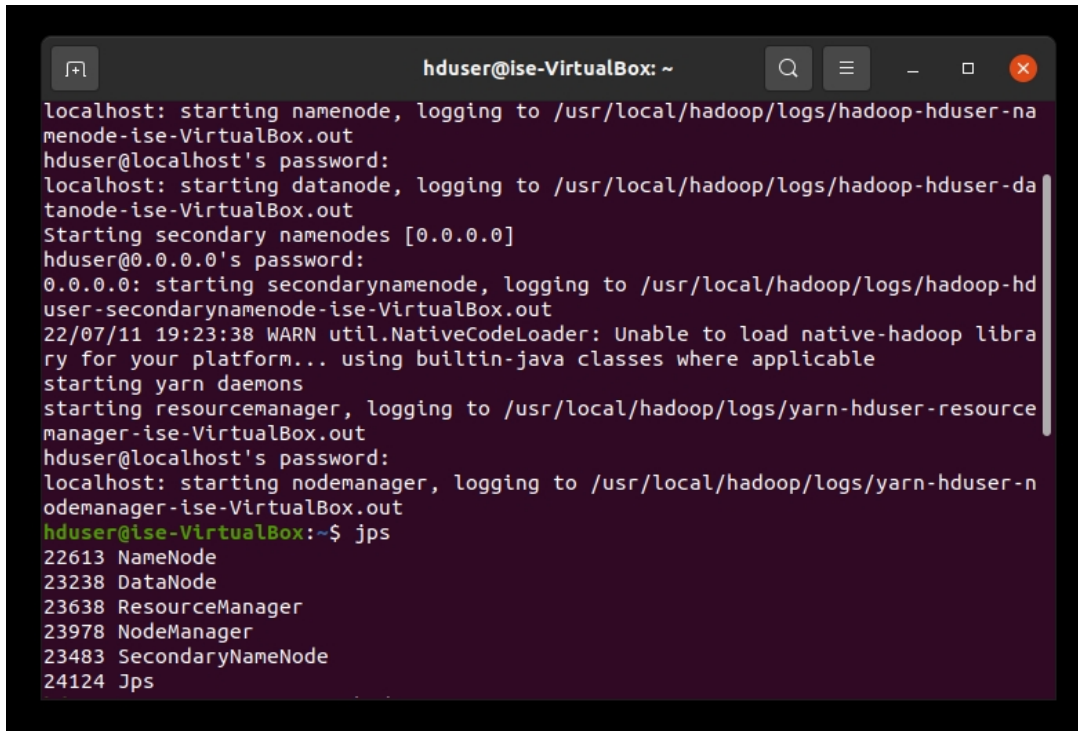
```
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19,
"contactno" : 9090909090, "email" : "abc@lab.com" }
```

```
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20, "contactno" :
9999900000, "email" : "mno@lab.com" }
```

```
> use studentdb
switched to db studentdb
> db.createCollection("student_details")
{ "ok" : 1 }
> db.student_details.insert({'name':'abc','rollno':1,'age':19,'contactno':9090909090,'email':'abc@lab.com'})
WriteResult({ "nInserted" : 1 })
> db.student_details.insert({'name':'mno','rollno':2,'age':20,'contactno':9999900000,'email':'mno@lab.com'})
WriteResult({ "nInserted" : 1 })
> db.student_details.insert({'name':'xyz','rollno':3,'age':21,'contactno':9999911111,'email':'xyz@lab.com'})
WriteResult({ "nInserted" : 1 })
> db.student_details.find({})
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19, "contactno" : 9090909090, "email" : "abc@lab.com" }
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20, "contactno" : 9999900000, "email" : "mno@lab.com" }
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "xyz", "rollno" : 3, "age" : 21, "contactno" : 9999911111, "email" : "xyz@lab.com" }
> db.student_details.update({'rollno':3},{set:{'email':'update@lab.com'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.student_details.find({'rollno':3})
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "xyz", "rollno" : 3, "age" : 21, "contactno" : 9999911111, "email" : "update@lab.com" }
> db.student_details.update({'name':'xyz',{set:{'name':'pqr'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.student_details.find({'name':'pqr'})
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "pqr", "rollno" : 3, "age" : 21, "contactno" : 9999911111, "email" : "update@lab.com" }
> db.student_details.find({})
{ "_id" : ObjectId("60a88f8ffecf7c8abe76777"), "name" : "pqr", "rollno" : 3, "age" : 21, "contactno" : 9999911111, "email" : "update@lab.com" }
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19, "contactno" : 9090909090, "email" : "abc@lab.com" }
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20, "contactno" : 9999900000, "email" : "mno@lab.com" }
> db.student_details.remove({'age':{'gt':20}})
WriteResult({ "nRemoved" : 1 })
> db.student_details.find({})
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19, "contactno" : 9090909090, "email" : "abc@lab.com" }
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20, "contactno" : 9999900000, "email" : "mno@lab.com" }
```

## LAB4

### SCREENSHOT OF HADOOP INSTALLATION



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



## LAB 5

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

```
c:\hadoop_new\sbin>hdfs dfs -mkdir /temp
```

```
c:\hadoop_new\sbin>hdfs dfs -copyFromLocal E:\Desktop\sample.txt \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 1 items

```
-rw-r--r-- 1 Admin supergroup      11 2021-06-11 21:12 /temp/sample.txt
```

```
c:\hadoop_new\sbin>hdfs dfs -cat \temp\sample.txt hello  
world
```

```
c:\hadoop_new\sbin>hdfs dfs -get \temp\sample.txt E:\Desktop\temp
```

```
c:\hadoop_new\sbin>hdfs dfs -put E:\Desktop\temp \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp
```

Found 2 items

```
-rw-r--r-- 1 Admin supergroup      11 2021-06-11 21:12 /temp/sample.txt drwxr-xr-x -  
Admin supergroup      0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -mv \lab1 \temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls \temp Found 3 items drwxr-xr-x - Admin  
supergroup      0 2021-04-19 15:07 /temp/lab1 -rw-r--r-- 1 Admin
```

```
supergroup      11 2021-06-11 21:12 /temp/sample.txt drwxr-xr-x -  
Admin supergroup    0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -rm /temp/sample.txt
```

```
Deleted /temp/sample.txt
```

```
c:\hadoop_new\sbin>hdfs dfs -ls /temp Found 2 items drwxr-xr-x - Admin
```

```
supergroup      0 2021-04-19 15:07 /temp/lab1 drwxr-xr-x - Admin
```

```
supergroup      0 2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -copyFromLocal E:\Desktop\sample.txt /temp
```

```
c:\hadoop_new\sbin>hdfs dfs -ls /temp Found 3 items drwxr-xr-x - Admin
```

```
supergroup      0 2021-04-19 15:07 /temp/lab1 -rw-r--r--  1 Admin supergroup
```

```
11 2021-06-11 21:17 /temp/sample.txt drwxr-xr-x - Admin supergroup      0
```

```
2021-06-11 21:15 /temp/temp
```

```
c:\hadoop_new\sbin>hdfs dfs -copyToLocal /temp/sample.txt E:\Desktop\sample.txt
```

```
Activities Terminal Jun 6 15:10 hdsuser@bmsce-Precision-T1700:~  
# owner: hdsuser  
# group: supergroup  
user::rwx  
group::r-x  
other::r-x  
  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar.txt /home/hdsuser/Desktop  
copyToLocal: '/Karthik/kar.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/karfile.txt /home/hdsuser/Desktop  
copyToLocal: '/Karthik/karfile.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar2.txt /home/hdsuser/Desktop  
copyToLocal: '/Karthik/kar2.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Kardir/  
# file: /Kardir  
# owner: hdsuser  
# group: supergroup  
user::rwx  
group::r-x  
other::r-x  
  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Kardir/kar.txt /home/hdsuser/Desktop  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Kardir/kar.txt  
Hello this is a sample Welcome Text File..  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -mv /Kardir /FFF  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -ls /FFF  
Found 2 items  
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 14:49 /FFF/Kardir  
-rw-r--r-- 1 hdsuser supergroup 31 2022-05-31 09:59 /FFF/sample2  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -cp /Kardir/ /LLL  
cp: '/Kardir/': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -cp /CSE/ /LLL  
cp: '/CSE/': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -cp /FFF/ /LLL  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -ls /LLL  
Found 2 items  
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 15:09 /LLL/Kardir  
-rw-r--r-- 1 hdsuser supergroup 31 2022-06-06 15:09 /LLL/sample2  
hdsuser@bmsce-Precision-T1700:~$
```

```
Activities Terminal Jun 6 15:05 hdsuser@bmsce-Precision-T1700:~  
command 'fs' from deb openafs-client (1.8.4-pre1-1ubuntu2.4)  
Try: sudo apt install <deb name>  
  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /Kardir/kar.txt /Kardir/kar2.txt /home/hdsuser/Desktop/Merge.txt  
getmerge: '/Kardir/kar2.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /Kardir/kar.txt /Kardir/kar.txt /home/hdsuser/Desktop/Merge.txt  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Karthik/  
# file: /Karthik  
# owner: hdsuser  
# group: supergroup  
user::rwx  
group::r-x  
other::r-x  
  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar.txt /home/hdsuser/Desktop  
copyToLocal: '/Karthik/kar.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/karfile.txt /home/hdsuser/Desktop  
copyToLocal: '/Karthik/karfile.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Karthik/kar2.txt /home/hdsuser/Desktop  
copyToLocal: '/Karthik/kar2.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Kardir/  
# file: /Kardir  
# owner: hdsuser  
# group: supergroup  
user::rwx  
group::r-x  
other::r-x  
  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /Kardir/kar.txt /home/hdsuser/Desktop  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Kardir/kar.txt  
Hello this is a sample Welcome Text File..  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -mv /Kardir /FFF  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -ls /FFF  
Found 2 items  
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 14:49 /FFF/Kardir  
-rw-r--r-- 1 hdsuser supergroup 31 2022-05-31 09:59 /FFF/sample2  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -cp /Kardir/ /LLL  
cp: '/Kardir/': No such file or directory
```

```
Activities Terminal Jun 6 15:05 hdsuser@bmsce-Precision-T1700:~  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 09:30 /sample  
drwxrwxr-x - hdsuser supergroup 0 2019-08-01 16:19 /tmp  
drwxr-xr-x - hdsuser supergroup 0 2019-08-01 16:03 /user  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/WC.txt  
hdfs dfs -cat /abc/WC.txt  
welcome ,hi good afternoon.  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/WC.txt  
welcome ,hi good afternoon.  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/kar.txt  
cat: '/abc/kar.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Karthik/kar.txt  
cat: '/Karthik/kar.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /Kardir/kar.txt  
Hello this is a sample Welcome Text File..  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -get /Kardir/kar.txt /home/hdsuser/Downloads/kar2.txt  
hdsuser@bmsce-Precision-T1700:~$ dfs dfs -getmerge /abc/WC.txt /abc/WC2.txt /home/hdsuser/Desktop/Merge.txt  
Command 'dfs' not found, did you mean:  
  
command 'dfc' from deb dfc (3.1.1-1)  
command 'dns' from deb anacrolix-dns (1.1.0-1)  
command 'df' from deb coreutils (8.30-3ubuntu2)  
command 'bfs' from deb bfs (1.5.2-1)  
command 'dcs' from deb drbl (2.30.5-1)  
command 'zfs' from deb zfsutils-linux (0.8.3-1ubuntu12.14)  
command 'zfs' from deb zfs-fuse (0.7.0-20)  
command 'hfs' from deb hfsutils-tcltk (3.2.6-14)  
command 'fs' from deb openafs-client (1.8.4-pre1-1ubuntu2.4)  
  
Try: sudo apt install <deb name>  
  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /Kardir/kar.txt /Kardir/kar2.txt /home/hdsuser/Desktop/Merge.txt  
getmerge: '/Kardir/kar2.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -getmerge /Kardir/kar.txt /Kardir/kar.txt /home/hdsuser/Desktop/Merge.txt  
hdsuser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /Karthik/  
# file: /Karthik  
# owner: hdsuser  
# group: supergroup  
user::rwx  
  
Activities Terminal Jun 6 15:05 hdsuser@bmsce-Precision-T1700:~  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 10:26 /hdsuser_copied  
drwxr-xr-x - hdsuser supergroup 0 2022-06-01 15:29 /new2  
drwxr-xr-x - hdsuser supergroup 0 2022-05-31 09:16 /sajjan  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 09:30 /sample  
drwxrwxr-x - hdsuser supergroup 0 2019-08-01 16:19 /tmp  
drwxr-xr-x - hdsuser supergroup 0 2019-08-01 16:03 /user  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hdsuser/Desktop/karfile.txt /Kardir/kar.txt  
put: '/Kardir/kar.txt': No such file or directory  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -put /home/hdsuser/Desktop/karfile.txt /Kardir/kar.txt  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -ls /  
Found 24 items  
drwxr-xr-x - hdsuser supergroup 0 2022-05-31 10:13 /127  
drwxr-xr-x - hdsuser supergroup 0 2022-06-03 12:51 /Desktop  
drwxr-xr-x - hdsuser supergroup 0 2022-05-31 10:13 /FFF  
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 14:49 /Kardir  
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 14:35 /Karthik  
drwxr-xr-x - hdsuser supergroup 0 2022-05-31 10:15 /NextFFF  
drwxr-xr-x - hdsuser supergroup 0 2022-06-03 15:10 /Revanth  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 09:56 /WC2.txt  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 09:48 /WC.txt  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 09:49 /Welcome.txt  
-rw-r--r-- 1 hdsuser supergroup 43 2022-06-06 14:44 /abc.txt  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 09:52 /abc  
drwxr-xr-x - hdsuser supergroup 0 2022-06-03 12:39 /akash  
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 12:23 /anant  
drwxr-xr-x - hdsuser supergroup 0 2022-06-06 12:13 /arlhant  
drwxr-xr-x - hdsuser supergroup 0 2022-06-03 15:03 /arya  
drwxr-xr-x - hdsuser supergroup 0 2022-06-01 15:31 /bindu  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 10:26 /hdsuser  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 10:26 /hdsuser_copied  
drwxr-xr-x - hdsuser supergroup 0 2022-06-01 15:29 /new2  
drwxr-xr-x - hdsuser supergroup 0 2022-05-31 09:16 /sajjan  
drwxr-xr-x - hdsuser supergroup 0 2022-06-04 09:30 /sample  
drwxrwxr-x - hdsuser supergroup 0 2019-08-01 16:19 /tmp  
drwxr-xr-x - hdsuser supergroup 0 2019-08-01 16:03 /user  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/WC.txt  
hdfs dfs -cat /abc/WC.txt  
welcome ,hi good afternoon.  
hdsuser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/WC.txt
```

```
Activities Terminal Jun 6 15:05
hduser@bmsce-Precision-T1700:~$ sudo su hduser
[sudo] password for hduser:
hduser@bmsce-Precision-T1700:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: namenode running as process 6828. Stop it first.
hduser@localhost's password:
localhost: datanode running as process 7000. Stop it first.
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: secondarynamenode running as process 7213. Stop it first.
starting yarn daemons
resourcemanager running as process 7372. Stop it first.
hduser@localhost's password:
localhost: nodemanager running as process 7707. Stop it first.
hduser@bmsce-Precision-T1700:~$ jps
7000 DataNode
7707 NodeManager
7372 ResourceManager
6828 NameNode
7213 SecondaryNameNode
9917 Jps
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /Kardir
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 24 items
drwxr-xr-x - hduser supergroup 0 2022-05-31 10:13 /127
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:51 /Desktop
drwxr-xr-x - hduser supergroup 0 2022-05-31 10:13 /FFF
drwxr-xr-x - hduser supergroup 0 2022-06-06 14:46 /Kardir
drwxr-xr-x - hduser supergroup 0 2022-06-06 14:35 /Karthik
drwxr-xr-x - hduser supergroup 0 2022-05-31 10:15 /NextFFF
drwxr-xr-x - hduser supergroup 0 2022-06-03 15:10 /Revanth
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:56 /WC2.txt
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:48 /Wc.txt
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:49 /Welcome.txt
-rw-r--r-- 1 hduser supergroup 43 2022-06-06 14:44 /abc.txt
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:52 /abc
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:39 /akash
```

```
Activities Terminal Jun 4 10:31
hduser@bmsce-Precision-T1700:~$ sbtn/start-all.sh
bash: sbtn/start-all.sh: No such file or directory
hduser@bmsce-Precision-T1700:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: namenode running as process 9999. Stop it first.
hduser@localhost's password:
localhost: datanode running as process 10175. Stop it first.
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: permission denied, please try again.
hduser@0.0.0.0's password:
0.0.0.0: secondarynamenode running as process 11044. Stop it first.
starting yarn daemons
resourcemanager running as process 11246. Stop it first.
hduser@localhost's password:
localhost: nodemanager running as process 11590. Stop it first.
hduser@bmsce-Precision-T1700:~$ jps
17169 Jps
11044 SecondaryNameNode
11590 NodeManager
11246 ResourceManager
9999 NameNode
10175 DataNode
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 17 items
drwxr-xr-x - hduser supergroup 0 2022-05-31 10:13 /127
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:51 /Desktop
drwxr-xr-x - hduser supergroup 0 2022-05-31 10:13 /FFF
drwxr-xr-x - hduser supergroup 0 2022-05-31 10:15 /NextFFF
drwxr-xr-x - hduser supergroup 0 2022-06-03 15:10 /Revanth
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:56 /WC2.txt
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:48 /Wc.txt
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:49 /Welcome.txt
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:52 /abc
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:39 /akash
drwxr-xr-x - hduser supergroup 0 2022-06-03 15:03 /arya
drwxr-xr-x - hduser supergroup 0 2022-06-01 15:31 /Dindu
drwxr-xr-x - hduser supergroup 0 2022-06-01 15:29 /newz
drwxr-xr-x - hduser supergroup 0 2022-05-31 09:10 /sajjan
drwxr-xr-x - hduser supergroup 0 2022-06-04 09:30 /sample
drwxr-xr-x - hduser supergroup 0 2019-08-01 10:19 /tmp
drwxr-xr-x - hduser supergroup 0 2019-08-01 16:03 /user
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /hduser
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /hduser2
mkdir: '/hduser': No such file or directory
mkdir: '/hduser2': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -touchz /hduser/myfile.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put ../Desktop/AI.txt /hduser
put: '../Desktop/AI.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put ../Desktop/AI.txt /hduser
put: '../Desktop/AI.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put /Desktop/AI.txt /hduser
put: '/Desktop/AI.txt': No such file or directory
```



## LAB 6

**For the given file, Create a Map Reduce program to**

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

```
// AverageDriver.java package temperature;
```

```
import org.apache.hadoop.io.*; import org.apache.hadoop.fs.*; import
org.apache.hadoop.mapreduce.*; 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 temperature;
```

```
import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.*; import java.io.IOException;
```

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

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

```
public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
```

```
{
    String line = value.toString();    String year = line.substring(15,19);    int temperature;
    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 != MISSING && quality.matches("[01459]"))
        context.write(new Text(year),new IntWritable(temperature)); }
```

```

}

//AverageReducer.java package temperature;

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

public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable>
{
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException,InterruptedException
    {
        int max_temp = 0;          int count = 0;
        for (IntWritable value : values)
        {
            max_temp += value.get();
            count+=1;
        }
        context.write(key, new IntWritable(max_temp/count));
    }
}

```

```

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

```

```

//TempDriver.java package
temperatureMax;

import org.apache.hadoop.io.*; import org.apache.hadoop.fs.*; import
org.apache.hadoop.mapreduce.*; import
org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class TempDriver
{
    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(TempDriver.class);          job.setJobName("Max
        temperature");

        FileInputFormat.addInputPath(job,new Path(args[0]));

        FileOutputFormat.setOutputPath(job,new Path (args[1]));

        job.setMapperClass(TempMapper.class);
        job.setReducerClass(TempReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        System.exit(job.waitForCompletion(true)?0:1);
    }
}

```

//TempMapper.java package

temperatureMax;

```

import org.apache.hadoop.io.*; import
org.apache.hadoop.mapreduce.*; import
java.io.IOException;

```

```

public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>

```

```

{ public static final int MISSING = 9999;

```

```

    public void map(LongWritable key, Text value, Context context) throws IOException,
    InterruptedException

```

```

{

```



```

        String line = value.toString();    String month = line.substring(19,21);
int temperature;        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 != MISSING &&
quality.matches("[01459]"))        context.write(new Text(month),new
IntWritable(temperature)); }

}

//TempReducer.java package
temperatureMax;

import org.apache.hadoop.io.*; import
org.apache.hadoop.mapreduce.*; import
java.io.IOException;

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

public void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException
{
        String line = value.toString();    String month = line.substring(19,21);
int temperature;        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 != MISSING &&

```

```
quality.matches("[01459]"))          context.write(new Text(month),new  
IntWritable(temperature));  
    }  
}
```

```
c:\hadoop_new\sbin>hdfs dfs -cat /tempMaxOutput/part-r-00000  
01      44  
02      17  
03     111  
04     194  
05     256  
06     278  
07     317  
08     283  
09     211  
10     156  
11      89  
12     117
```

## LAB 7

**For a given Text file, create a Map Reduce program to sort the content in an alphabetic order listing only top 'n' maximum occurrence of words.**

```
// TopN.java package sortWords;

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.Reducer; import
org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import
org.apache.hadoop.util.GenericOptionsParser; import utils.MiscUtils;

import java.io.IOException; import java.util.*;

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.setCombinerClass(TopNReducer.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);
    }

    /**
     * The mapper reads one line at the time, splits it into an array of single words and emits every
     * word to the reducers with the value of 1.
     */
    public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {

        private final static IntWritable one = new IntWritable(1);    private Text word = new Text();
        private String tokens = "[_ $#<>\\^=\\[\\]\\|\\*\\/\\\\\\.,;\\.:\\-\\!\\?\\'\""]";

        @Override
```

```

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

/**
 * The reducer retrieves every word and puts it into a Map: if the word already exists in the      * map,
increments its value, otherwise sets it to 1.
 */
public static class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

    private Map<Text, IntWritable> countMap = new HashMap<>();

    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {

        // computes the number of occurrences of a single word      int sum = 0;      for
(IntWritable val : values) {      sum += val.get();
        }
        // puts the number of occurrences of this word into the map.
        // We need to create another Text object because the Text instance
        // we receive is the same for all the words      countMap.put(new Text(key), new
IntWritable(sum));
    }
    @Override
    protected void cleanup(Context context) throws IOException, InterruptedException {

        Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(countMap);

        int counter = 0;      for (Text key : sortedMap.keySet()) {      if (counter++ == 3) {
break;
        }
        context.write(key, sortedMap.get(key));
    }
    }
}

/**
 * The combiner retrieves every word and puts it into a Map: if the word already exists in the      *
map, increments its value, otherwise sets it to 1.
 */
public static class TopNCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {

```

```

    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
    InterruptedException {

        // computes the number of occurrences of a single word
        (IntWritable val : values) {
            int sum = 0;
            for (
                sum += val.get();
            }
            context.write(key, new IntWritable(sum));
        }
    }
}

// MiscUtils.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;
    }
}

```

```
C:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \sortwordsOutput\part-r-00000  
car      7  
deer     6  
bear     3
```

## LAB 8

**Create a Hadoop Map Reduce program to combine information from the users file along with Information from the posts file by using the concept of join and display user\_id, Reputation and Score.**

```
// 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);
    }
}
}

```

```

c:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \joinOutput\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

```
scala> println("Hello World!");  
Hello World!
```

```
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;
```

```
hadoop@wave-ubu: ~/hadoop_files/scalacountwords$ spark-shell -i countwords.scala  
21/06/14 13:01:47 WARN Utils: Your hostname, wave-ubu resolves to a loopback address: 127.0.1.1; using  
21/06/14 13:01:47 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
21/06/14 13:01:47 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... usi  
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://192.168.2.7:4040  
Spark context available as 'sc' (master = local[*], app id = local-1623655911213).  
Spark session available as 'spark'.  
wasn't: 6  
what: 5  
as: 7  
she: 13  
it: 23  
he: 5  
for: 6  
her: 12  
the: 30  
was: 19  
be: 8  
It: 7  
but: 11  
had: 5  
would: 7  
in: 9  
you: 6  
that: 8  
a: 9  
or: 5  
to: 20  
I: 5  
of: 6  
and: 16  
Welcome to
```

## LAB 10

Using RDD and Flat Map 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/sam/Desktop/abc.txt")
textfile: org.apache.spark.rdd.RDD[String] = /home/sam/Desktop/abc.txt MapPartitionsRDD[8] at textFile at <console>:25

scala> val counts = textfile.flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey(_+_ )
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[11] at reduceByKey at <console>:26

scala> import scala.collection.immutable.ListMap
import scala.collection.immutable.ListMap

scala> val sorted = ListMap(counts.collect.sortWith(_. _2 >= _. _2):_*)
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(hello -> 3, apple -> 2, unicorn -> 1, world -> 1)

scala> println(sorted)
ListMap(hello -> 3, apple -> 2, unicorn -> 1, world -> 1)
```