VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BDA Lab record

Submitted by

Aisha Taffazul Chesti (1BM21CS010)

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
Feb-2024 to July-2024

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Big data Analytics lab" carried out by **Aisha Taffazul Chesti (1BM21CS010),** 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 2024. The Lab report has been approved as it satisfies the academic requirements in respect of a **BDA lab record - (22CS6PEBDA)** work prescribed for the said degree.

Dr. Pallavi G B
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

Lab 1

AIM: Perform the following DB operations using Cassandra.

1. **Create a keyspace by name Employee**

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

2. **Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name**

```
USE Employee;

CREATE TABLE Employee_Info (
   Emp_Id int PRIMARY KEY,
   Emp_Name text,
   Designation text,
   Date_of_Joining date,
   Salary decimal,
   Dept_Name text
);
```

3. **Insert the values into the table in batch**

```
BEGIN BATCH

INSERT INTO Employee_Info (Emp_Id,
Emp_Name, Designation, Date_of_Joining,
Salary, Dept_Name) VALUES (121, 'John Doe',
'Software Engineer', '2022-01-15', 60000,
'Engineering');

INSERT INTO Employee_Info (Emp_Id,
Emp_Name, Designation, Date_of_Joining,
```

Salary, Dept_Name) VALUES (122, 'Jane Smith', 'Data Scientist', '2021-06-20', 75000, 'Data Science');

INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) VALUES (123, 'Emily Davis', 'Project Manager', '2019-03-05', 85000, 'Management');

APPLY BATCH;

4. **Update Employee name and Department of Emp-Id 121 5. Sort the details of Employee records based on salary**

   UPDATE Employee_Info SET Emp_Name = 'Johnathan Doe', Dept_Name = 'Product Development' WHERE Emp_Id = 121;

5. **Sort the details of Employee records based on salary**

   SELECT * FROM Employee_Info;

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.**
   ALTER TABLE Employee_Info ADD Projects set<text>;

7. **Update the altered table to add project names.**

UPDATE Employee_Info SET Projects = {'Project Alpha', 'Project Beta'} WHERE Emp_Id = 121;

UPDATE Employee_Info SET Projects = {'Project Gamma'} WHERE Emp_Id = 122;

UPDATE Employee_Info SET Projects = {'Project Delta', 'Project Epsilon'} WHERE Emp_Id = 123;

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

INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name, Projects) VALUES (124, 'Michael Brown', 'System Analyst', '2020-09-11', 70000, 'IT', {'Project Zeta'}) USING TTL 15;

```
aisha@AishasLaptop:~$ ls
40x  apache-cassandra-4.0.13  deb  main
aisha@AishasLaptop:~$ cd apache-cassandra-4.0.13
aisha@AishasLaptop:~/apache-cassandra-4.0.13$ bin/cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.13 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> create keyspace if not exists Employee with replication={'class':'SimpleStrategy', 'replication_factor':1};
cqlsh> create table if not exists Employee.Employee_Info(Emp_id int primary key, emp_name text, designation text, date_of_joining timestamp, salary
float, dept_name text);
cqlsh> begin batch
   ... insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining, salary, dept_name) values (121, 'aisha','Software engineer', '2024-0
8-07', 3600000,'Engineering');
   ... insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining, salary, dept_name) values (122, 'fatima','prompt engineer', '202
4-11-07', 4600000,'Engineering');
   ... insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining, salary, dept_name) values (123, 'shayan','civi; engineer', '2024
-11-07', 4600000,'Engineering');
   ... apply batch;
InvalidRequest: Error from server: code=2200 [Invalid query] message="No keyspace has been specified. USE a keyspace, or explicitly specify keyspace
.tablename"
cqlsh> use Employee
   ... ;
cqlsh:employee> begin batch  insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining, salary, dept_name) values (121, 'aisha','Softw
are engineer', '2024-08-07', 3600000,'Engineering'); insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining, salary, dept_name) val
ues (122, 'fatima','prompt engineer', '2024-11-07', 4600000,'Engineering'); insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining,
 salary, dept_name) values (123, 'shayan','civi; engineer', '2024-11-07', 4600000,'Engineering'); apply batch;
cqlsh:employee> update Employee_Info set emp_name= 'zainub', dept_name='software testing' where Emp_Id =121;
cqlsh:employee> select * from employee_info
           ... ;

 emp_id | date_of_joining | dept_name        | designation       | emp_name | salary
--------+-----------------+------------------+-------------------+----------+---------
    123 |      2024-11-07 |      Engineering |    civi; engineer |   shayan | 4.6e+06
    122 |      2024-11-07 |      Engineering |   prompt engineer |   fatima | 4.6e+06
    121 |      2024-08-07 | software testing | Software engineer |   zainub | 3.6e+06

(3 rows)
cqlsh:employee>
```

```
(3 rows)
cqlsh:employee> alter table Employee
           ... _Info add projects set<text>;
Invalid syntax at char 1
  _Info add projects set<text>;
  ^
cqlsh:employee> alter table Employee_Info add projects set<text>
           ... ;
cqlsh:employee> update Employee_Info SET projects ={'alpha','beta'} wher
e emp_id =121;
cqlsh:employee> update Employee_Info SET projects ={'gamma'} where emp_i
d =122;
cqlsh:employee> update Employee_Info SET projects ={'delta'} where emp_i
d =123;
cqlsh:employee> update Employee_Info SET projects ={'delta'} where emp_id =123;
cqlsh:employee> insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining, salary, dept_name, projects) values (124, 'ateeb','ece
engineer', '2024-09-07', 5600000,'Engineering','zelta') using ttl 15;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid STRING constant (zelta) for "projects" of type set<text>"
cqlsh:employee> insert into Employee_Info(Emp_id, emp_name, designation,date_of_joining, salary, dept_name, projects) values (124, 'ateeb','ece engi
neer', '2024-09-07', 5600000,'Engineering',{'zelta'}) using ttl 15;
cqlsh:employee> select * from emplyee_info
           ... ;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table emplyee_info does not exist"
cqlsh:employee> select * from Employee_Info;

 emp_id | date_of_joining | dept_name        | designation       | emp_name | projects          | salary
--------+-----------------+------------------+-------------------+----------+-------------------+---------
    123 |      2024-11-07 |      Engineering |    civi; engineer |   shayan |         {'delta'} | 4.6e+06
    122 |      2024-11-07 |      Engineering |   prompt engineer |   fatima |         {'gamma'} | 4.6e+06
    121 |      2024-08-07 | software testing | Software engineer |   zainub | {'alpha', 'beta'} | 3.6e+06

cqlsh:employee>
cqlsh:employee>
```

LAB 2

AIM- Perform the following DB operations using Cassandra.

1.Create a keyspace by name 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

3. Insert the values into the table in batch

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

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

6. Export the created column to a csv file

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

cqlsh> CREATE KEYSPACE library5 WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1}; cqlsh> USE library5;

cqlsh:library0717> create table library_info17(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:library5> update library_info ... set counter_value=counter_value+1 ... where stud_id=112 and stud_name='shruti' and book_id=101 and book_name='os' and date_of_issue='2023-07-09';

cqlsh:library5> update library_info ... set counter_value=counter_value+1 ... where stud_id=111 and stud_name='raj' and book_id=100 and book_name='ada' and date_of_issue='2022-04-07';

stud_id | stud_name | book_name | _of_issue | book_id | counter_value ----

-----+-----------+-----------+---------------------------------+---------+------------

--- 112 |aisha | OS | 2023-07-09 18:30:00.000000+0000 | 101 | 1

111 | shayan | ada | 2022-04-07 18:30:00.000000+0000 | 100 | 1

cqlsh:library0717> select counter_value as borrow_count ... from library_info17 ... where stud_id=1;

```
cqlsh:library5> copy library5.library_info(stud_id, stud_name, book_name, book_id,
date_of_issue, counter_value) to '/home/bmsce/cassandra/data.csv' WITH
HEADER=TRUE;

cqlsh:library5> copy library5.library_info(stud_id, stud_name, book_name, book_id,
date_of_issue, counter_value) from '/home/bmsce/cassandra/data.csv' WITH
HEADER=TRUE;
```

LAB 3

AIM- MongoDB CRUD Documentation

C:\Users\bmsce>mongosh "mongodb+srv://cluster1.kipkn5v.mongodb.net/DBMS_demo1" --apiVersion 1 --username aisha

Enter password: *******

Current Mongosh Log ID: 643a27cceb99a8175b2ff0e9

Connecting to:
mongodb+srv://<credentials>@cluster1.kipkn5v.mongodb.net/DBMS_demo1?appName=mongosh+1.8.0

Using MongoDB: 6.0.5 (API Version 1)
Using Mongosh: 1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).

You can opt-out by running the disableTelemetry() command.

Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:1, Age:21. Cont:9876, email:"antara.de9@gmail.com"});

Uncaught:

SyntaxError: Unexpected token, expected "," (1:38)

> 1 | db.Student.insert({RollNo:1, Age:21. Cont:9876, email:"antara.de9@gmail.com"}); | ^
  2 |

Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:1, Age:21, Cont:9876, email:

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

Uncaught:

SyntaxError: Unexpected token, expected "," (2:79)

  1 | db.Student.insert({RollNo:1, Age:21, Cont:9876, email:**

> 2 | db.Student.insert({RollNo:1, Age:21, Cont:9876, email:"antara.de9@gmail.com"}); | ^

  3 |
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:1, Age:21, Cont:9876, email:"antara.de9@gmail.com"});

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

  acknowledged: true,

  insertedIds: { '0': ObjectId("643a2898ee0bdaa6ffbdf1dd") }

}

Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:2, Age:22, Cont:9976, email:"anuksha.de9@gmai.com"});

{

  acknowledged: true,

  insertedIds: { '0': ObjectId("643a28c8ee0bdaa6ffbdf1de") }

}

Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:3, Age:21, Cont:5576, email:

... "

Uncaught:

SyntaxError: Unterminated string constant. (2:0)

  1 | db.Student.insert({RollNo:3, Age:21, Cont:5576, email:

> 2 | "

    | ^

  3 |

Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:3, Age:21, Cont:5576, email: "anubhav.de9@gmail.com"});

{

  acknowledged: true,

  insertedIds: { '0': ObjectId("643a28f2ee0bdaa6ffbdf1df") }

}
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:4, Age:20, Cont:4476, email:"pani.de9@gmail.com"});

{

  acknowledged: true,

```
  insertedIds: { '0': ObjectId("643a291eee0bdaa6ffbdf1e0") }
}
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert( {RollNo:10,
Age:23, Cont:2276, email:"rekha.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("643a293cee0bdaa6ffbdf1e1") }
}
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.find() [
  {
    _id: ObjectId("643a2898ee0bdaa6ffbdf1dd"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("643a28c8ee0bdaa6ffbdf1de"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anuksha.de9@gmai.com'
  },
  {
    _id: ObjectId("643a28f2ee0bdaa6ffbdf1df"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("643a291eee0bdaa6ffbdf1e0"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
```

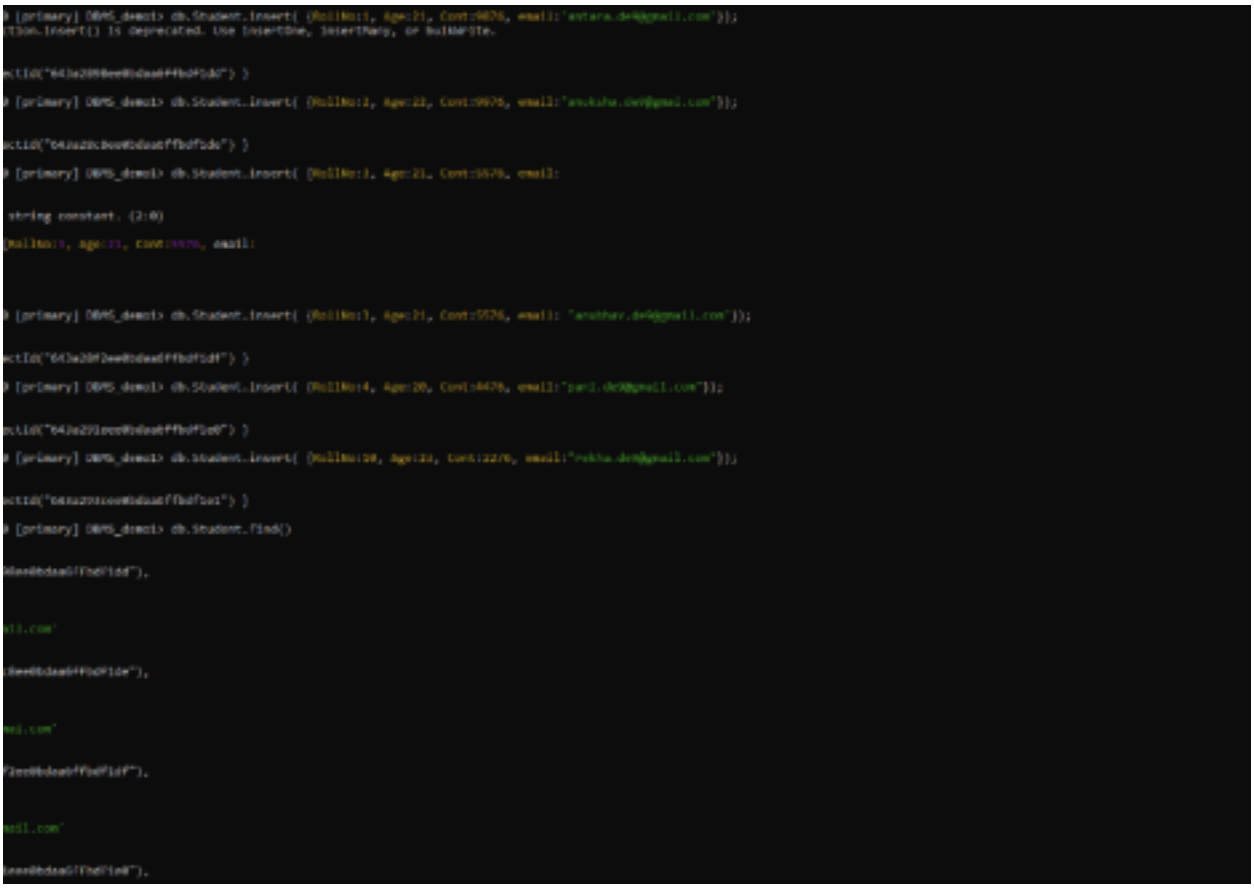
```
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("643a293cee0bdaa6ffbdf1e1"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1>
db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or
bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.insert({RollNo:11,
Age:22, Name:"ABC",Cont:2276,email:"rea.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("643a29acee0bdaa6ffbdf1e2") }
}
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.find() [
  {
    _id: ObjectId("643a2898ee0bdaa6ffbdf1dd"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
```

```
{
  _id: ObjectId("643a28c8ee0bdaa6ffbdf1de"),
  RollNo: 2,
  Age: 22,
  Cont: 9976,
  email: 'anuksha.de9@gmai.com'
},
{
  _id: ObjectId("643a28f2ee0bdaa6ffbdf1df"),
  RollNo: 3,
  Age: 21,
  Cont: 5576,
  email: 'anubhav.de9@gmail.com'
},
{
  _id: ObjectId("643a291eee0bdaa6ffbdf1e0"),
  RollNo: 4,
  Age: 20,
  Cont: 4476,
  email: 'pani.de9@gmail.com'
},
{
  _id: ObjectId("643a293cee0bdaa6ffbdf1e1"),
  RollNo: 10,
  Age: 23,
  Cont: 2276,
  email: 'Abhinav@gmail.com'
},
{
  _id: ObjectId("643a29acee0bdaa6ffbdf1e2"),
  RollNo: 11,
  Age: 22,
  Name: 'ABC',
  Cont: 2276,
```

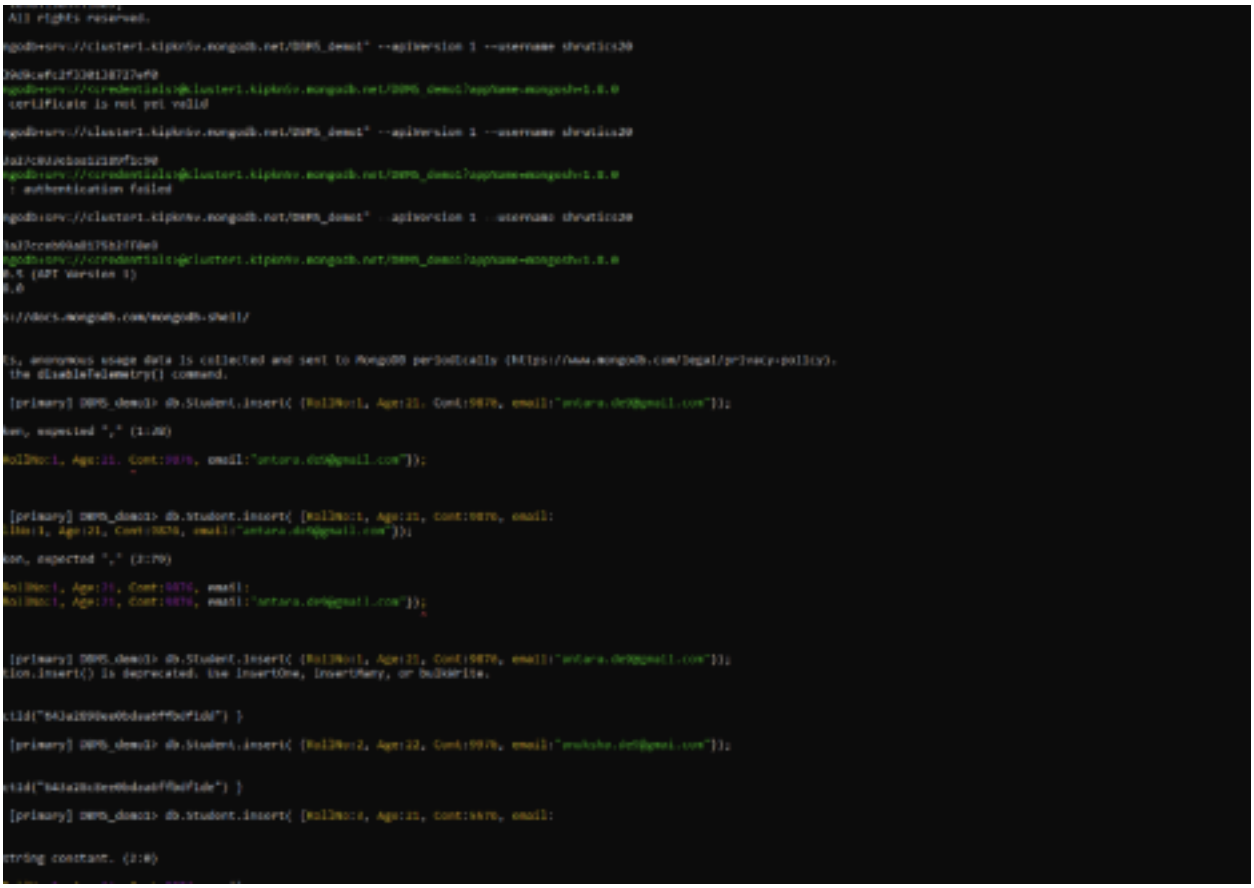
```
    email: 'rea.de9@gmail.com'
  }
]
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.update({RollNo:11,
Name:"ABC"},{$set:{Name:"FEM"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.show()
TypeError: db.Student.show is not a function
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1> db.Student.find() [
  {
    _id: ObjectId("643a2898ee0bdaa6ffbdf1dd"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("643a28c8ee0bdaa6ffbdf1de"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anuksha.de9@gmai.com'
  },
  {
    _id: ObjectId("643a28f2ee0bdaa6ffbdf1df"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
```

```
  },
  {
    _id: ObjectId("643a291eee0bdaa6ffbdf1e0"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("643a293cee0bdaa6ffbdf1e1"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  },
  {
    _id: ObjectId("643a29acee0bdaa6ffbdf1e2"),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
Atlas atlas-10724g-shard-0 [primary] DBMS_demo1>
```

```
.com'

e.ee0bdaa6ffbdf1e2"),

.com'

0 [primary] OBMS_demo1> db.Student.update({RollNo:11, Name:"ABC"},{$set:{Name:"XYZ"}});

0 [primary] OBMS_demo1> db.Student.Show()
w is not a function
0 [primary] OBMS_demo1> db.Student.Find()

90ee0bdaa6ffbdf1d2"),

ail.com'

1be0bdaa6ffbdf1de"),

mail.com'

f7ee0bdaa6ffbdf1df"),

mail.com'

3ee0bdaa6ffbdf1e0"),

1.com'

0ee0bdaa6ffbdf1e1"),

.com'
```

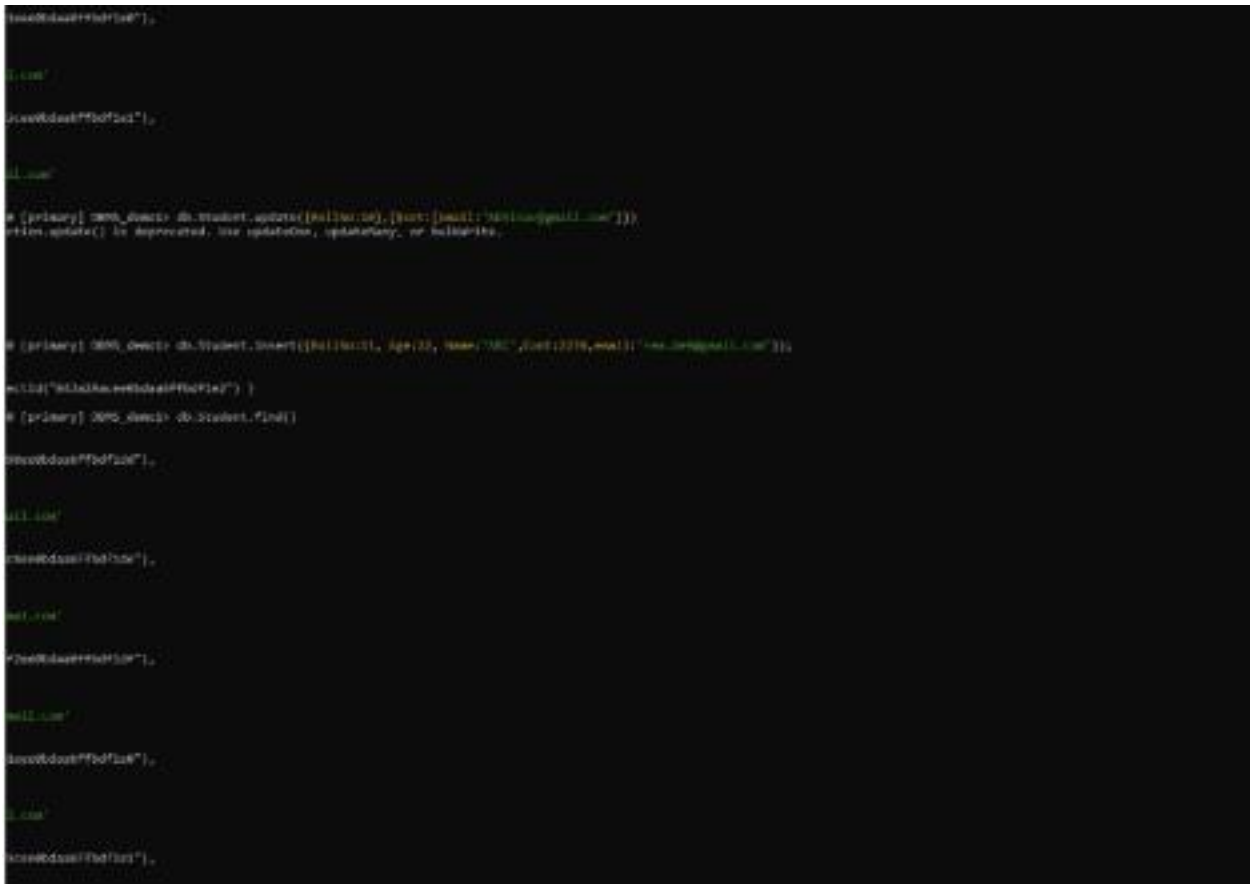


```
0 [primary] OBMS_demo1> db.Student.Insert( {RollNo:1, Age:21, Cont:9876, email:'antara.de4@gmail.com'});
rTion.Insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

ctId("661a2090ee0bdaa6ffbdf1d0") }

0 [primary] OBMS_demo1> db.Student.Insert( {RollNo:2, Age:22, Cont:9878, email:'anukshe.sen@gmail.com'});

ctId("661a220c0ee0bdaa6ffbdf1d6") }

0 [primary] OBMS_demo1> db.Student.Insert( {RollNo:3, Age:21, Cont:5678, email:

 string constant. (2:0)

{RollNo:3, Age:21, Cont:5678, email:

0 [primary] OBMS_demo1> db.Student.Insert( {RollNo:7, Age:21, Cont:5526, email: 'anushav.de4@gmail.com'});

ctId("661a20f2ee0bdaa6ffbdf1df") }

0 [primary] OBMS_demo1> db.Student.Insert( {RollNo:4, Age:20, Cont:4478, email:'pari.de4@gmail.com'});

ctId("661a21beee0bdaa6ffbdf1e0") }

0 [primary] OBMS_demo1> db.Student.Insert( {RollNo:10, Age:21, Cont:2276, email:"rekha.de4@gmail.com"});

ctId("661a23feee0bdaa6ffbdf1e1") }

0 [primary] OBMS_demo1> db.Student.Find()

0ee0bdaa6ffbdf1dd"),

ail.com'

1ee0bdaa64ffbdf1de"),

mail.com'

f1ee0bdaa6ffbdf1df"),

mail.com'

3ee0bdaa6ffbdf1e0"),
```



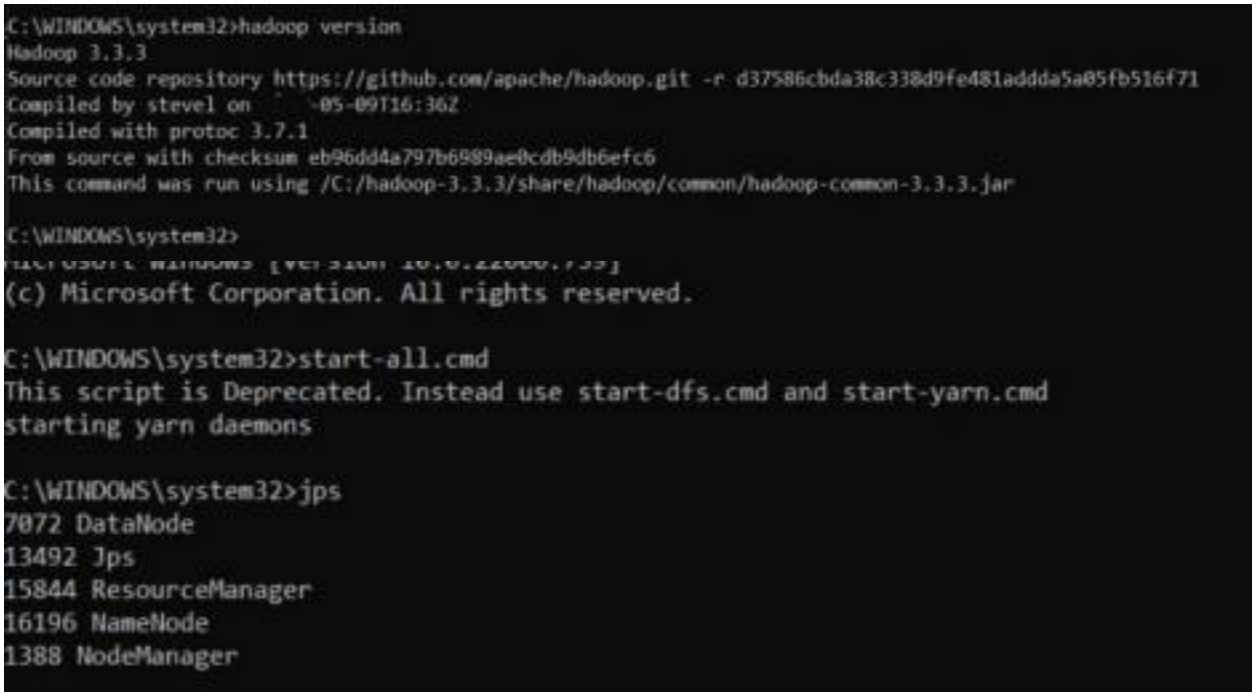
```
All rights reserved.

mongodb+srv://cluster1.kipknv.mongodb.net/DBMS_demo1" --apiVersion 1 --username shruti:s20

mongodb+srv://credentials:@cluster1.kipknv.mongodb.net/DBMS_demo1?appName=mongosh+1.8.0
certificate is not yet valid

mongodb+srv://cluster1.kipknv.mongodb.net/DBMS_demo1" --apiVersion 1 --username shruti:s20

mongodb+srv://credentials:@cluster1.kipknv.mongodb.net/DBMS_demo1?appName=mongosh+1.8.0
: authentication failed

mongodb+srv://cluster1.kipknv.mongodb.net/DBMS_demo1" --apiVersion 1 --username shruti:s20

mongodb+srv://credentials:@cluster1.kipknv.mongodb.net/DBMS_demo1?appName=mongosh+1.8.0
8.5 (API Version 1)
8.0

s://docs.mongodb.com/mongodb-shell/


), anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
 the disableTelemetry() command.

 [primary] DBMS_demo1> db.Student.insert( {RollNo:1, Age:21, Cont:9870, email:"antara.de9@gmail.com"});

on, expected "," (1:28)

RollNo:1, Age:21, Cont:9870, email:"antara.de9@gmail.com"});
                             ^


 [primary] DBMS_demo1> db.student.insert( {RollNo:1, Age:21, cont:9870, email:
:1, Age:21, Cont:9870, email:"antara.de9@gmail.com"});

on, expected "," (2:79)

RollNo:1, Age:21, Cont:9870, email:
RollNo:1, Age:21, Cont:9870, email:"antara.de9@gmail.com"});
                                  ^


 [primary] DBMS_demo1> db.Student.insert( {RollNo:1, Age:21, Cont:9870, email:"antara.de9@gmail.com"});
tion.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.


tId("64ja8990ee0bdaa6ffbdf1dd") }

 [primary] DBMS_demo1> db.Student.insert( {RollNo:2, Age:22, Cont:9876, email:"prakshe.de9@gmai.com"});


tId("64ja8bc0ee0bdaa6ffbdf1de") }

 [primary] DBMS_demo1> db.student.insert( {RollNo:3, Age:21, cont:9876, email:


tring constant. (2:0)
```

```
ail.com'


:0ee0bdaa6ffbdf1de"),


mail.com'


f1ee0bdaa6ffbdf1df"),


mail.com'


0ee0bdaa6ffbdf1e0"),


).com'


3ee0bdaa6ffbdf1e3"),


.com'


0ee0bdaa6ffbdf1e2"),



.com'


0 [primary] DBMS_demo1>
xit or (Ctrl-D) or type .exit)
0 [primary] DBMS_demo1> mongoimport
rt is not defined
0 [primary] DBMS_demo1>
```



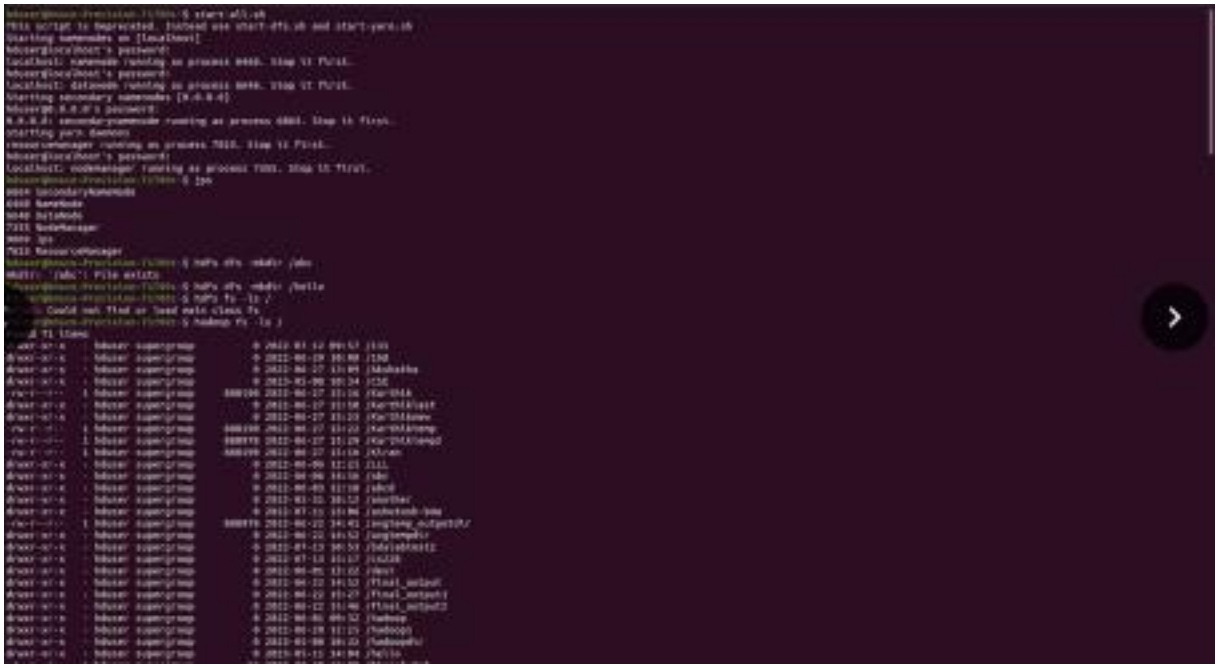
LAB 4

AIM- Screenshot of Hadoop Installed



LAB 5

AIM- Execution of HDFS Commands for Interaction with Hadoop Environment



LAB 6

AIM- Implement WordCount Program on Hadoop framework

Mapper Code:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class WCMapper extends MapReduceBase implements Mapper<LongWritable,
Text, Text,
IntWritable> {
public void map(LongWritable key, Text value, OutputCollector<Text,
IntWritable> output, Reporter rep) throws IOException
{
String line = value.toString();
for (String word : line.split(" "))
{
if (word.length() > 0)
{
output.collect(new Text(word), new IntWritable(1));
} } } }
```

Reducer Code:

```
// Importing libraries
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class WCReducer extends MapReduceBase implements Reducer<Text,
IntWritable, Text, IntWritable> {
// Reduce function
public void reduce(Text key, Iterator<IntWritable> value,
OutputCollector<Text, IntWritable> output,
Reporter rep) throws IOException
{
int count = 0;
// Counting the frequency of each words
while (value.hasNext())
{
IntWritable i = value.next();
count += i.get();
}
output.collect(key, new IntWritable(count));
} }
```

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

```
// Importing libraries
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
```

```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
public class WCDriver extends Configured implements Tool {
public int run(String args[]) throws IOException
{
if (args.length < 2)
{
System.out.println("Please give valid inputs");
return -1;
}
JobConf conf = new JobConf(WCDriver.class);
FileInputFormat.setInputPaths(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));
conf.setMapperClass(WCMapper.class);
conf.setReducerClass(WCReducer.class);
conf.setMapOutputKeyClass(Text.class);
conf.setMapOutputValueClass(IntWritable.class);
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
JobClient.runJob(conf);
return 0;
}
// Main Method
public static void main(String args[]) throws Exception
{
int exitCode = ToolRunner.run(new WCDriver(), args);
System.out.println(exitCode);
}
                        }
```

LAB 7

AIM-Create a Map Reduce program to

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

```
package temp;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class AverageDriver {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("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);23
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}
```

AverageMapper

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

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class AverageMapper extends Mapper<LongWritable, Text, Text, IntWritable>
{ 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);24
if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(year), new IntWritable(temperature));
}
}
```

AverageReducer

```
package temp;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class AverageReducer extends Reducer<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();

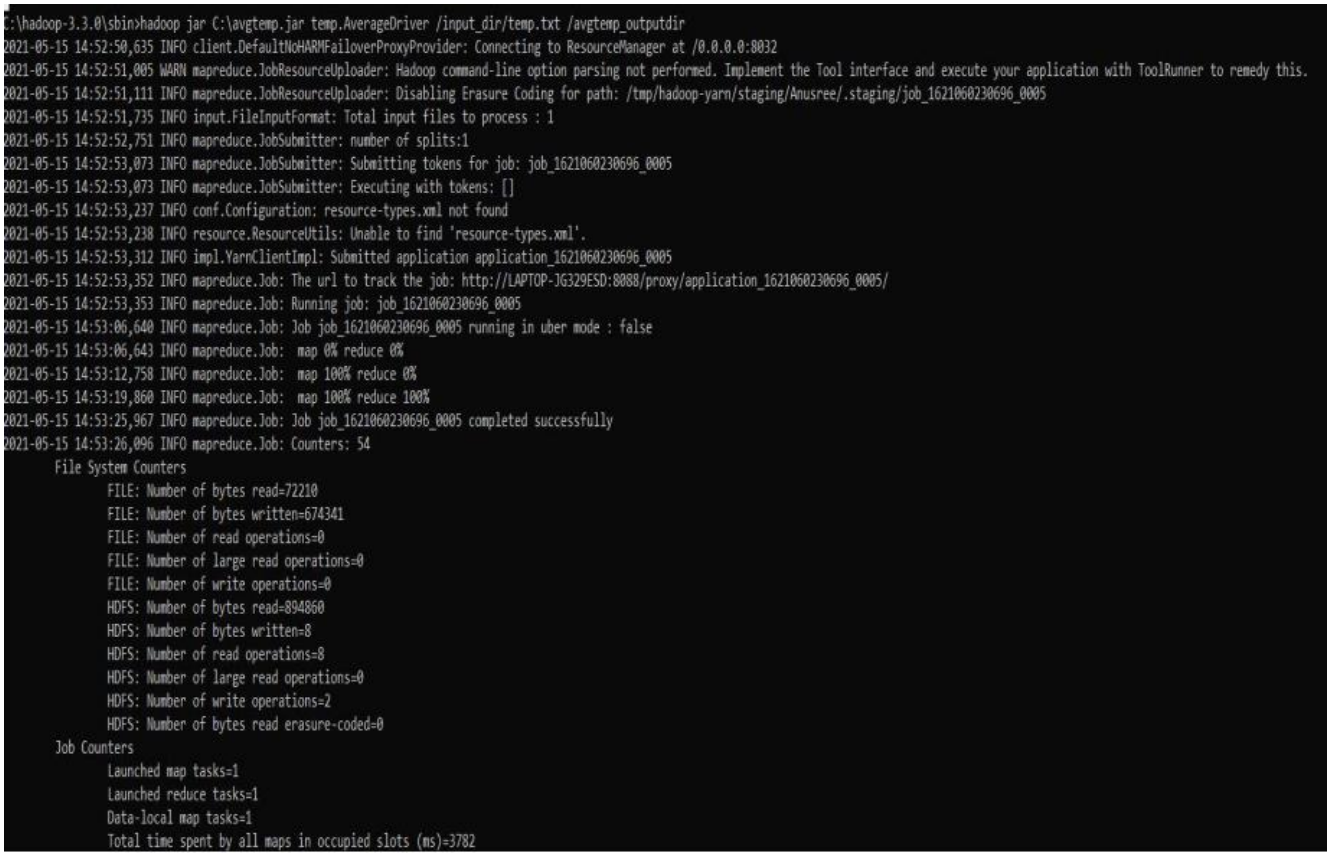
count++;

}

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

}

}



b) Find the mean max temperature for every month

MeanMaxDriver

package meanmax;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

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

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

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

if (args.length != 2) {

System.err.println("Please Enter the input and output parameters");

System.exit(-1);

}

Job job = new Job();

job.setJarByClass(MeanMaxDriver.class);

job.setJobName("Max temperature");

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

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

job.setMapperClass(MeanMaxMapper.class);

job.setReducerClass(MeanMaxReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}
```

```
C:\hadoop-3.3.0\sbin>hadoop jar C:\avgtemp.jar temp.AverageDriver /input_dir/temp.txt /avgtemp_outputdir
2021-05-15 14:52:50,635 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-15 14:52:51,005 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-15 14:52:51,111 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621060230696_0005
2021-05-15 14:52:51,735 INFO input.FileInputFormat: Total input files to process : 1
2021-05-15 14:52:52,751 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621060230696_0005
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-15 14:52:53,237 INFO conf.Configuration: resource-types.xml not found
2021-05-15 14:52:53,238 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-15 14:52:53,312 INFO impl.YarnClientImpl: Submitted application application_1621060230696_0005
2021-05-15 14:52:53,352 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621060230696_0005/
2021-05-15 14:52:53,353 INFO mapreduce.Job: Running job: job_1621060230696_0005
2021-05-15 14:53:06,640 INFO mapreduce.Job: Job job_1621060230696_0005 running in uber mode : false
2021-05-15 14:53:06,643 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-15 14:53:12,758 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-15 14:53:19,860 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-15 14:53:25,967 INFO mapreduce.Job: Job job_1621060230696_0005 completed successfully
2021-05-15 14:53:26,096 INFO mapreduce.Job: Counters: 54
      File System Counters
            FILE: Number of bytes read=72210
            FILE: Number of bytes written=674341
            FILE: Number of read operations=0
            FILE: Number of large read operations=0
            FILE: Number of write operations=0
            HDFS: Number of bytes read=894860
            HDFS: Number of bytes written=8
            HDFS: Number of read operations=8
            HDFS: Number of large read operations=0
            HDFS: Number of write operations=2
            HDFS: Number of bytes read erasure-coded=0
      Job Counters
            Launched map tasks=1
            Launched reduce tasks=1
            Data-local map tasks=1
            Total time spent by all maps in occupied slots (ms)=3782
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -ls /avgtemp_outputdir
Found 2 items
-rw-r--r--   1 Anusree supergroup          0 2021-05-15 14:53 /avgtemp_outputdir/_SUCCESS
-rw-r--r--   1 Anusree supergroup          8 2021-05-15 14:53 /avgtemp_outputdir/part-r-00000

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /avgtemp_outputdir/part-r-00000
1901    46

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

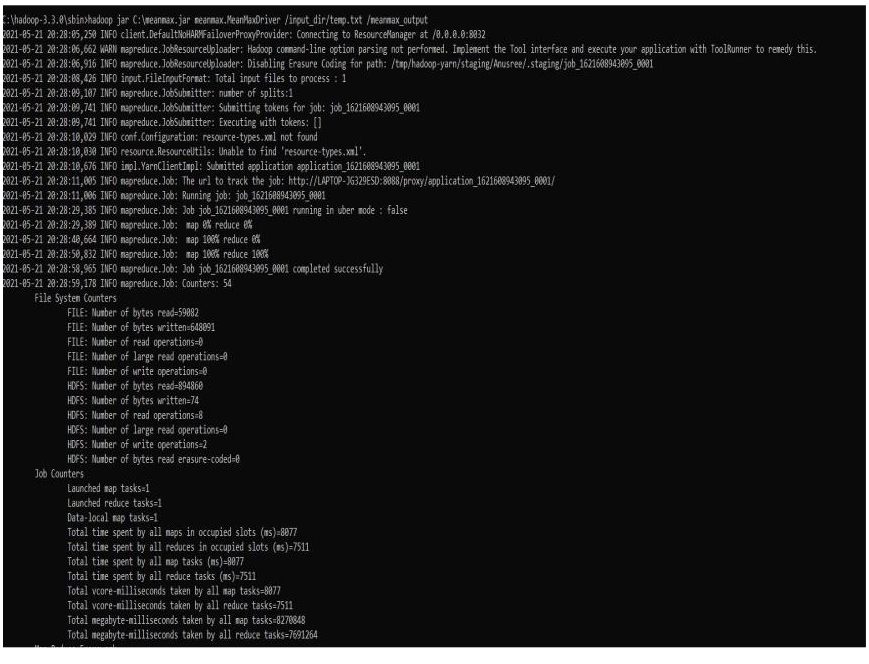
MeanMax Mapper

```
package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class MeanMaxMapper extends Mapper<LongWritable, Text, Text, IntWritable>
{ public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
int temperature;
String line = value.toString();
String month = line.substring(19, 21);
if (line.charAt(87) == '+') {
temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(month), new IntWritable(temperature));
}
}
```

MeanMax Reducer

```
package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class MeanMaxReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
```

```
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int total_temp = 0;
int count = 0;
int days = 0;

for (IntWritable value : values) {
int temp = value.get();
if (temp > max_temp)
max_temp = temp;
count++;
if (count == 3) {
total_temp += max_temp;
max_temp = 0;
count = 0;
days++;
}
}
context.write(key, new IntWritable(total_temp / days));
}
}
```



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

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

LAB 8

AIM- 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.

alphabetic order listing only top 10 maximum occurrences of words.

//Driver Code

```
package wordCount;

import java.io.IOException;

import org.apache.hadoop.conf.Configured;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.FileInputFormat;

import org.apache.hadoop.mapred.FileOutputFormat;

import org.apache.hadoop.mapred.JobClient;

import org.apache.hadoop.mapred.JobConf;

import org.apache.hadoop.util.Tool;

import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

public int run(String args[]) throws IOException29

{

if (args.length < 2)

{

System.out.println("Please give valid inputs");

return -1;

}

JobConf conf = new JobConf(WCDriver.class);

FileInputFormat.setInputPaths(conf, new Path(args[0]));

FileOutputFormat.setOutputPath(conf, new Path(args[1]));
conf.setMapperClass(WCMapper.class);

conf.setReducerClass(WCReducer.class);

conf.setMapOutputKeyClass(Text.class);
```

```
conf.setMapOutputValueClass(IntWritable.class);
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
JobClient.runJob(conf);
return 0;
}
// Main Method
public static void main(String args[]) throws Exception
{
int exitCode = ToolRunner.run(new WCDriver(), args);
System.out.println(exitCode);
}
}
```

//Mapper Code

```
package wordCount;30
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class WCMapper extends MapReduceBase implements Mapper<LongWritable,Text,
Text, IntWritable> {
// Map function
public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable>
output, Reporter
rep) throws IOException
{
String line = value.toString();
// Splitting the line on spaces
for (String word : line.split(" "))
```

```
{
if (word.length() > 0)
{
output.collect(new Text(word), new IntWritable(1));
}
}
}
}
```

//Reducer Code

```
package wordCount;
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class WCReducer extends MapReduceBase implements Reducer<Text,IntWritable,
Text, IntWritable> {
// Reduce function
public void reduce(Text key, Iterator<IntWritable> value, OutputCollector<Text, IntWritable>
output,Reporter rep) throws IOException
{
int count = 0;
// Counting the frequency of each words
```



```
while (value.hasNext())
{
IntWritable i = value.next();
count += i.get();
```

}

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

}

}

//Hadoop Commands

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 10473. Stop it first. hduser@localhost's password:

localhost: datanode running as process 10644. 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 10857. Stop it first. starting yarn daemons

resourcemanager running as process 9796. Stop it first. hduser@localhost's password:

localhost: nodemanager running as process 10160. Stop it first.
hduser@bmsce-Precision-T1700:~$ jps

10160 NodeManager
7441 org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar

9796 ResourceManager

12692 org.eclipse.equinox.launcher_1.5.600.v20191014-

2022.jar 10644 DataNode

10857 SecondaryNameNode

10473 NameNode

15100 Jps

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

Found 10 items33

drwxr-xr-x - hduser supergroup 0 2023-01-23 09:52 /gou drwxr-xr-

x - hduser supergroup 0 2023-01-23 10:33 /har drwxr-xr-x - hduser

supergroup 0 2023-04-14 10:50 /input drwxr-xr-x - hduser

supergroup 0 2023-05-23 09:58 /output1 drwxr-xr-x - hduser

supergroup 0 2023-01-23 15:57 /output2 drwxr-xr-x - hduser

supergroup 0 2023-01-15 10:27 /rgs drwxr-xr-x - hduser supergroup

0 2023-01-23 11:09 /stud drwxr-xr-x - hduser supergroup 0 2023-

05-23 15:50 /testing drwxrwxr-x - hduser supergroup 0 2023-05-23

11:24 /tmp drwxr-xr-x - hduser supergroup 0 2023-05-01 16:03

/user hduser@bmsce-Precision-T1700:~$ hadoop fs -mkdir

/1bm20cs216 hduser@bmsce-Precision-T1700:~$ hadoop fs -

copyFromLocal /home/hduser/Desktop/sample.txt

/1bm20cs216/test.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /1bm20cs216/test.txt

hi how are you

how is your job

how is your family

how is your brother
how is your sister

hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/Documents/wordCount.jar

wordCount.WCDriver /1bm20cs216/test.txt /1bm20cs216/output

File System Counters

FILE: Number of bytes read=8614

FILE: Number of bytes written=510599

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=178

HDFS: Number of bytes written=69

HDFS: Number of read operations=13

HDFS: Number of large read operations=0

HDFS: Number of write operations=4

Map-Reduce Framework

Map input records=5

Map output records=20

Map output bytes=169

Map output materialized bytes=215

Input split bytes=87

Combine input records=0

Combine output records=0

Reduce input groups=10

Reduce shuffle bytes=215

Reduce input records=20

Reduce output records=10

Spilled Records=40

Shuffled Maps =1
Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=1

CPU time spent (ms)=0

Physical memory (bytes) snapshot=0

Virtual memory (bytes) snapshot=0

Total committed heap usage (bytes)=47185920038

Shuffle Errors

BAD_ID=0

CONNECTION=0

IO_ERROR=0

WRONG_LENGTH=0

WRONG_MAP=0

WRONG_REDUCE=0

File Input Format Counters

Bytes Read=89

File Output Format Counters

Bytes Written=69

0

hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /1bm20cs216/output/part-00000

are 1

brother 1

family 1

hi 1

how 5

is 4

job 1

sister 1
you 1

your

4

(_+_);

reducedata.collect;

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

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

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

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

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

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

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