

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

### **Big Data Analytics (23CS6PCBDA)**

*Submitted by*

**Nithya Lakshmi V(1BM22CS186)**

*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-2025 to June-2025**

**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 (23CS6PCBDA)**" carried out by **Nithya Lakshmi V(1BM22CS186)**, 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 **Big Data Analytics - (23CS6PCBDA)** work prescribed for the said degree.

**Prof. Vikranth BM**  
Assistant Professor  
BMSCE, Bengaluru

**Dr. Kavitha Sooda**  
Professor and HoD of CSE  
BMSCE, Bengaluru

<b>Lab Program</b>	<b>INDEX Program Details</b>	<b>Page No</b>
1	MongoDB- CRUD Operations Demonstration (Practice and Self Study)	1-5
2	Perform the following DB operations using Cassandra. a) Create a keyspace by name Employee b) Create a column family by name Employee-info with attributes... (and more)	6-7
3	Perform the following DB operations using Cassandra. a) Create a keyspace by name Library b) Create a column family by name Library-info with attributes... (and more)	8-10
4	Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)	11-12
5	Implement Wordcount program on Hadoop framework	13-15
6	From the following link extract the weather data <a href="https://github.com/tomwhite/hadoop-book/blob/master/input/ncdc/all...">https://github.com/tomwhite/hadoop-book/blob/master/input/ncdc/all...</a> (and more)	16-23
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.	24-28
8	Write a Scala program to print numbers from 1 to 100 using for loop.	29
9	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.	30
10	Write a simple streaming program in Spark to receive text data streams on a particular port, perform basic text cleaning (like white space removal, stop words removal, lemmatization, etc.), and print the cleaned text on the screen. (Open Ended Question).	31-32

### **Course Outcome**

<b>CO 1</b>	Apply the concept of NoSQL, Hadoop or Spark for a given task
<b>CO 2</b>	Analyze big data analytics mechanisms that can be applied to obtain solution for a given problem.
<b>CO 3</b>	Design and implement solutions using data analytics mechanisms for a given problem.

# LABORATORY PROGRAM – 1

## MongoDB- CRUD Operations Demonstration (Practice and Self Study)

### OBSERVATION

### COMMAND WITH OUTPUT - USING ATLAS

```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>mongosh "mongodb+srv://cluster0.qh8blz4.mongodb.net/" --apiVersion 1 --username likhithcs22
Enter password: *****
Current Mongosh Log ID: 67c6c754899c67e814fa4213
Connecting to:      mongodb+srv://<credentials>@cluster0.qh8blz4.mongodb.net/?appName=mongosh+2.4.0
Using MongoDB:     8.0.5 (API Version 1)
Using Mongosh:    2.4.0

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

Atlas atlas-2vljb9-shard-0 [primary] test> show dbs
e-commerce 108.00 KiB
myDB        40.00 KiB
admin       232.00 KiB
local       15.70 GiB
Atlas atlas-2vljb9-shard-0 [primary] test> use myDB
switched to db myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> db
myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
...
... db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
...
... db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
...
... db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
...
... db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67c6c898899c67e814fa4218') }
}
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67c6c8a3899c67e814fa4219') }
}
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67c6c8f7899c67e814fa421a') }
}
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67c6c8fb899c67e814fa421b') }
}

C:\Users\likhi>mongosh "mongodb+srv://cluster0.qh8blz4.mongodb.net/" --apiVersion 1 --username likhithcs22
Enter password: *****
Current Mongosh Log ID: 6833148466c722794490defd
Connecting to:      mongodb+srv://<credentials>@cluster0.qh8blz4.mongodb.net/?appName=mongosh+2.2.9
Using MongoDB:     8.0.9 (API Version 1)
Using Mongosh:    2.2.9
mongosh 2.5.1 is available for download: https://www.mongodb.com/try/download/shell

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

Atlas atlas-2vljb9-shard-0 [primary] test> show dbs
e-commerce 108.00 KiB
myDB        72.00 KiB
admin       312.00 KiB
local       64.34 GiB
Atlas atlas-2vljb9-shard-0 [primary] test> use myDB
switched to db myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> db
myDB
Atlas atlas-2vljb9-shard-0 [primary] myDB> show collections
Student
```

```
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.find()
[
  {
    _id: ObjectId('67c6c898899c67e814fa4214'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c898899c67e814fa4215'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c898899c67e814fa4216'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c898899c67e814fa4217'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c898899c67e814fa4218'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  },
  {
    _id: ObjectId('67c6c8a3899c67e814fa4219'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c8f7899c67e814fa421a'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c8fb899c67e814fa421b'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c8fd899c67e814fa421c'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c904899c67e814fa421d'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6ca34899c67e814fa421e'),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
```

```
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.updateOne({"RollNo": 10}, {"$set": {"email": "john.deo@gmail.com"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-2vljb9-shard-0 [primary] myDB> db.Student.find(
... {"Name": /F/}
...
[
  {
    _id: ObjectId('67c6ca34899c67e814fa421e'),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.deo@gmail.com'
  }
]
Atlas atlas-2vljb9-shard-0 [primary] myDB> |
```

## MongoDB- CRUD Operations Demonstration (Practice and Self Study)

### COMMAND WITH OUTPUT - USING UBUNTU TERMINAL

```
MyDataBase> use MyDataBase
already on db MyDataBase
MyDataBase> show collections
Customers
NewStudent
Student
MyDataBase> db.Student.find();
[
  {
    _id: 1,
    studName: 'Michellejacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, Grade: 'VII', StudName: 'AryanDavid', Hobbies: 'Skating' },
  { _id: 2, Grade: 'VIII', StudName: 'Ram', Hobbies: 'Learning' }
]
```

```

test> use MyDataBase
switched to db MyDataBase
MyDataBase> show collections
NewStudent
NewStudent2
Student
MyDataBase> db.NewStudent2.drop();
true
MyDataBase> db.createCollection("Customers");
{ ok: 1 }
MyDataBase> db.Customers.insertMany([{"cust_id":1,Balance:200, Type:"S"},]);
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67d00571207666297fa3b81a') }
}
MyDataBase> db.Customers.insert({cust_id:1,Balance:1000, Type:"Z"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67d0058f207666297fa3b81b') }
}
My DataBase> db.Customers.insert({cust_id:2,Balance:100, Type:"Z"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67d0059c207666297fa3b81c') }
}
My DataBase> db.Customers.insert({cust_id:2,Balance:1000, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67d005a5207666297fa3b81d') }
}
My DataBase> db.Customers.insert({cust_id:2,Balance:500, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67d005ad207666297fa3b81e') }
}
My DataBase> db.Customers.insert({cust_id:2,Balance:50, Type:"S"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67d005b2207666297fa3b81f') }
}
My DataBase> db.Customers.insert({cust_id:3,Balance:500, Type:"Z"});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId('67d005ba207666297fa3b820') }
}

My DataBase> db.Customers.aggregate([
...   {
...     $group: {
...       _id: "$cust_id", // Group by cust_id
...       minAccBal: { $min: "$Balance" }, // Find the minimum Balance
...       maxAccBal: { $max: "$Balance" } // Find the maximum Balance
...     }
...   }
... ]);
[ { _id: 3, minAccBal: 500, maxAccBal: 500 },
  { _id: 2, minAccBal: 50, maxAccBal: 1000 },
  { _id: 1, minAccBal: 200, maxAccBal: 1000 }

]

My DataBase> db.Customers.aggregate([
...   { $match: { Type: "Z" } },
...   { $group: { _id: "$cust_id", TotAccBal: { $sum: "$Balance" } } },
...   { $match: { TotAccBal: { $gt: 1200 } } }
... ]);

```

```
MyDataBase> db.Customers.aggregate([
...   { $match: { Type: "Z" } },
...   {
...     $group: {
...       _id: "$cust_id",
...       TotAccBal: { $sum: "$Balance" }
...     }
...   },
...   {
...     $match: {
...       TotAccBal: { $gt: 200 }
...     }
...   }
... ]);
[ { _id: 3, TotAccBal: 500 }, { _id: 1, TotAccBal: 1000 } ]
```

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --host localhost --db MyDataBase --collection NewStudent2 --type=csv --file /home/bmscecse/Desktop/135.txt --headerline
2025-03-11T14:55:05.192+0530    connected to: mongodb://localhost/
2025-03-11T14:55:05.360+0530    3 document(s) imported successfully. 0 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db MyDataBase --collection NewStudent2 --type=json --file /home/bmscecse/Desktop/135.txt
2025-03-11T14:55:24.438+0530    error parsing command line options: unknown option "file"
2025-03-11T14:55:24.438+0530    try 'mongoexport --help' for more information
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db MyDataBase --collection NewStudent2 --type=json --out /home/bmscecse/Desktop/135.txt
2025-03-11T14:55:32.771+0530    connected to: mongodb://localhost/
2025-03-11T14:55:32.780+0530    exported 3 records
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

## LABORATORY PROGRAM – 2

Perform the following DB operations using Cassandra

### Questions:

- a) Create a keyspace by name Employee
- b) Create a column family by name
  - Employee-Info with attributes
  - Emp\_Id Primary Key, Emp\_Name,
  - Designation, Date\_of\_Joining,
  - Salary, Dept\_Name
- c) Insert the values into the table in batch
- d) Update Employee name and Department of Emp-Id 121
- e) Sort the details of Employee records based on salary
- f) Alter the schema of the table Employee\_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
- g) Update the altered table to add project names.
- h) Create a TTL of 15 seconds to display the values of Employees.

### OBSERVATION

### COMMAND WITH OUTPUT

```
cqlsh> CREATE KEYSPACE IF NOT EXISTS Employee
...   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_Info (
...   ...   Emp_Id INT PRIMARY KEY,
...   ...   Emp_Name TEXT,
...   ...   Designation TEXT,
...   ...   Date_of_Joining DATE,
...   ...   Salary DOUBLE,
...   ...   Dept_Name TEXT
...   ... );
cqlsh:employee> BEGIN BATCH
...   ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
...   ... VALUES (121, 'John Doe', 'Manager', '2018-01-01', 90000, 'HR');
...
...   ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
...   ... VALUES (122, 'Alice Smith', 'Developer', '2019-05-21', 75000, 'IT');
...
...   ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
...   ... VALUES (123, 'Rahul Roy', 'Analyst', '2020-07-15', 65000, 'IT');
...   ... APPLY BATCH;
cqlsh:employee> UPDATE Employee_Info
...   ... SET Emp_Name = 'John Smith', Dept_Name = 'Finance'
...   ... WHERE Emp_Id = 121;
cqlsh:employee> select * from Employee_Info;

emp_id | date_of_joining | dept_name | designation | emp_name      | salary
-----+-----------------+-----------+-------------+--------------+-----
  123 |    2020-07-15 |       IT |     Analyst | Rahul Roy |  65000
  122 |    2019-05-21 |       IT | Developer | Alice Smith |  75000
  121 |    2018-01-01 | Finance | Manager   | John Smith |  90000

(3 rows)
```

```
(3 rows)
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_By_Dept (
...   Dept_Name TEXT,
...   Salary DOUBLE,
...   Emp_Id INT,
...   Emp_Name TEXT,
...   Designation TEXT,
...   Date_of_Joining DATE,
...   PRIMARY KEY (Dept_Name, Salary, Emp_Id)
... ) WITH CLUSTERING ORDER BY (Salary DESC, Emp_Id ASC);
cqlsh:employee> BEGIN BATCH
...   INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
...   VALUES ('HR', 90000, 121, 'John Smith', 'Manager', '2018-01-01');
...
...   INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
...   VALUES ('IT', 75000, 122, 'Alice Smith', 'Developer', '2019-05-21');
...
...   INSERT INTO Employee_By_Dept (Dept_Name, Salary, Emp_Id, Emp_Name, Designation, Date_of_Joining)
...   VALUES ('IT', 65000, 123, 'Rahul Roy', 'Analyst', '2020-07-15');
...   APPLY BATCH;
cqlsh:employee> SELECT * FROM Employee_By_Dept WHERE Dept_Name = 'IT';

dept_name | salary | emp_id | date_of_joining | designation | emp_name
-----+-----+-----+-----+-----+-----+
IT | 75000 | 122 | 2019-05-21 | Developer | Alice Smith
IT | 65000 | 123 | 2020-07-15 | Analyst | Rahul Roy

(2 rows)
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects SET<TEXT>;
cqlsh:employee> UPDATE Employee_Info SET Projects = ['ERP System', 'HR Portal'] WHERE Emp_Id = 121;
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
...   VALUES (124, 'Sneha Kapoor', 'Tester', '2023-03-10', 55000, 'QA') USING TTL 15;
cqlsh:employee> select * from Employee_Info;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary
-----+-----+-----+-----+-----+-----+-----+
123 | 2020-07-15 | IT | Analyst | Rahul Roy | null | 65000
122 | 2019-05-21 | IT | Developer | Alice Smith | null | 75000
121 | 2018-01-01 | Finance | Manager | John Smith | ['ERP System', 'HR Portal'] | 90000

(3 rows)
```

## LABORATORY PROGRAM – 3

### Perform the following DB operations using Cassandra

#### Questions:

- a) Create a keyspace by name Library
- b) 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
- c) Insert the values into the table in batch
- d) Display the details of the table created and increase the value of the counter
- e) Write a query to show that a student with id 112 has taken a book “BDA” 2 times.
- f) Export the created column to a csv file
- g) Import a given csv dataset from local file system into Cassandra column family

#### OBSERVATION

#### COMMAND WITH OUTPUT

```
cqlsh:employee> CREATE KEYSPACE IF NOT EXISTS Library
...   WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh:employee> USE Library;
cqlsh:library> CREATE TABLE IF NOT EXISTS Library_Info (
...   Stud_Id INT PRIMARY KEY,
...   Stud_Name TEXT,
...   Book_Name TEXT,
...   Book_Id TEXT,
...   Date_of_Issue DATE
... );
cqlsh:library> CREATE TABLE IF NOT EXISTS Book_Counter (
...   Stud_Id INT,
...   Book_Name TEXT,
...   Counter_Value COUNTER,
...   PRIMARY KEY ((Stud_Id), Book_Name)
... );
cqlsh:library> BEGIN BATCH
...   INSERT INTO Library_Info (Stud_Id, Stud_Name, Book_Name, Book_Id, Date_of_Issue)
...   VALUES (112, 'Anjali Rao', 'BDA', 'B101', '2024-10-01');
...
...   INSERT INTO Library_Info (Stud_Id, Stud_Name, Book_Name, Book_Id, Date_of_Issue)
...   VALUES (113, 'Karthik N', 'AI', 'B102', '2024-11-11');
...   APPLY BATCH;
cqlsh:library> UPDATE Book_Counter SET Counter_Value = Counter_Value + 1 WHERE Stud_Id = 112 AND Book_Name = 'BDA';
cqlsh:library> UPDATE Book_Counter SET Counter_Value = Counter_Value + 1 WHERE Stud_Id = 112 AND Book_Name = 'BDA';
cqlsh:library> SELECT * FROM Book_Counter WHERE Stud_Id = 112 AND Book_Name = 'BDA';

stud_id | book_name | counter_value
-----+-----+-----
  112  |    BDA   |      4
-----+-----+-----
```

(1 rows)

```
cqlsh:students> DESCRIBE TABLE Students_Info;
CREATE TABLE students.students_info (
    roll_no int PRIMARY KEY,
    dateofjoining timestamp,
    last_exam_percent double,
    studname text
) WITH 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 memtable = 'default'
    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';
cqlsh:students> BEGIN BATCH
...   INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
...     VALUES (1, 'Asha', '2012-03-12', 79.9);
...   INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
...     VALUES (2, 'Kiran', '2012-03-12', 89.9);
...   INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
...     VALUES (3, 'Shanthi', '2012-03-12', 90.9);
...   INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
...     VALUES (4, 'Smith', '2012-03-12', 67.9);
...   INSERT INTO Students_Info (Roll_No, StudName, DateOfJoining, Last_Exam_Percent)
...     VALUES (5, 'Rohan', '2012-03-12', 56.9);
... APPLY BATCH;
cqlsh:students> SELECT * FROM Students_Info;

```

roll_no	dateofjoining	last_exam_percent	studname
5	2012-03-11 18:30:00.000000+0000	56.9	Rohan
1	2012-03-11 18:30:00.000000+0000	79.9	Asha
2	2012-03-11 18:30:00.000000+0000	89.9	Kiran
4	2012-03-11 18:30:00.000000+0000	67.9	Smith
3	2012-03-11 18:30:00.000000+0000	90.9	Shanthi

(5 rows)

```
cqlsh> CREATE KEYSPACE Students WITH REPLICATION =
...   {'class': 'SimpleStrategy', 'replication_factor': '1'};
cqlsh>
cqlsh> USE Students;
cqlsh:students> DESCRIBE KEYSPACES;

```

companies	library	products	system	system_traces
company	pro	productss	system_auth	system_views
employe	prod	productsss	system_distributed	system_virtual_schema
employee	productname	students	system_schema	

```
cqlsh:students> CREATE TABLE Students_Info (
...   Roll_No int PRIMARY KEY,
...   StudName text,
...   DateOfJoining timestamp,
...   last_exam_Percent double
... );
cqlsh:students> SELECT * FROM system.schema_keyspaces;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table schema_keyspaces does not exist"
cqlsh:students>
cqlsh:students> SELECT * FROM system.schema.keyspaces;

```

keyspace_name	durable_writes	replication
companies	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_auth	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_schema	True	{'class': 'org.apache.cassandra.locator.LocalStrategy'}
library	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
products	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_distributed	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
system	True	{'class': 'org.apache.cassandra.locator.LocalStrategy'}
productsss	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
prod	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
pro	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_traces	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}
students	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
company	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
employe	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
productname	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
employe	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
productss	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}

(17 rows)

```
cqlsh:students> DESCRIBE TABLES;

```

students_info
---------------

```

cqlsh:students> SELECT * FROM Students_Info WHERE Roll_No IN (1,2,3);
   roll_no | dateofjoining           | last_exam_percent | studname
-----+-----+-----+-----+
   1 | 2012-03-11 18:30:00.000000+0000 |      79.9 |     Asha
   2 | 2012-03-11 18:30:00.000000+0000 |      89.9 |    Kiran
   3 | 2012-03-11 18:30:00.000000+0000 |      90.9 | Shanthi

(3 rows)
cqlsh:students> CREATE INDEX ON Students_Info (StudName);
cqlsh:students> SELECT * FROM Students_Info WHERE StudName = 'Asha';

   roll_no | dateofjoining           | last_exam_percent | studname
-----+-----+-----+-----+
   1 | 2012-03-11 18:30:00.000000+0000 |      79.9 |     Asha

(1 rows)
cqlsh:students> SELECT Roll_No, StudName FROM Students_Info LIMIT 2;

   roll_no | studname
-----+-----
   5 | Rohan
   1 | Asha

(2 rows)
cqlsh:students> SELECT Roll_No AS USN FROM Students_Info;

   usn
-----
   5
   1
   2
   4
   3

(5 rows)
cqlsh:students> UPDATE Students_Info
... SET StudName = 'David Sheen'
... WHERE Roll_No = 2;
cqlsh:students> UPDATE Students_Info SET Roll_No = 6 WHERE Roll_No = 3; -- ✗ ERROR!
InvalidRequest: Error from server: code=2200 [Invalid query] message="PRIMARY KEY part roll_no found in SET part"

```

```

cqlsh:students> DELETE Last_Exam_Percent FROM Students_Info WHERE Roll_No = 2;
cqlsh:students> DELETE FROM Students_Info WHERE Roll_No = 2;
cqlsh:students> ALTER TABLE Students_Info ADD hobbies SET<text>;
cqlsh:students> ALTER TABLE Students_Info ADD languages LIST<text>;
cqlsh:students> UPDATE Students_Info
... SET hobbies = hobbies + {'Chess', 'Table Tennis'}
... WHERE Roll_No = 1;
cqlsh:students> CREATE TABLE library_book (
...   counter_value counter,
...   book_name text,
...   stud_name text,
...   PRIMARY KEY(book_name, stud_name)
... );
cqlsh:students> UPDATE library_book
...   SET counter_value = counter_value + 1
...   WHERE book_name = 'Big Data Analytics' AND stud_name = 'Jeet';
cqlsh:students> CREATE TABLE userlogin (
...   userid int PRIMARY KEY,
...   password text
... );
cqlsh:students> INSERT INTO userlogin (userid, password)
...   VALUES (1, 'infy') USING TTL 30;
cqlsh:students> SELECT TTL(password) FROM userlogin WHERE userid = 1;

   ttl(password)
-----
   20

(1 rows)
cqlsh:students> COPY Students_Info TO '/home/bmscsecse/Desktop/Student_Info.csv';
Using 16 child processes

Starting copy of students.students_info with columns [roll_no, dateofjoining, hobbies, languages, last_exam_percent, studname].
Processed: 4 rows; Rate:      38 rows/s; Avg. rate:      38 rows/s
4 rows exported to 1 files in 0.124 seconds.
cqlsh:students> COPY Students_Info FROM '/home/bmscsecse/Desktop/Student_Info.csv';
Using 16 child processes

Starting copy of students.students_info with columns [roll_no, dateofjoining, hobbies, languages, last_exam_percent, studname].
Processed: 4 rows; Rate:      7 rows/s; Avg. rate:     11 rows/s
4 rows imported from 1 files in 0.377 seconds (0 skipped).
cqlsh:students> COPY person (id, fname, lname) FROM STDIN;
Column family person not found
cqlsh:students> COPY Students_Info TO STDOUT;
5,2012-03-11 18:30:00.000+0000,,56.9,Rohan
1,2012-03-11 18:30:00.000+0000,"{'Chess', 'Table Tennis'}",,79.9,Asha
4,2012-03-11 18:30:00.000+0000,,67.9,Smith
3,2012-03-11 18:30:00.000+0000,,,90.9,Shanthi
cqlsh:students>

```

## LABORATORY PROGRAM – 4

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

#### OBSERVATION

#### COMMAND WITH OUTPUT

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mkdir /Lab05
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Hadoop
ls: `/Hadoop': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab05
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ touch test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -put ./text.txt /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab05
Found 1 items
-rw-r--r-- 1 hadoop supergroup 19 2024-05-13 14:33 /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /Lab05/text.txt
Hello
How are you?
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab05
Found 2 items
-rw-r--r-- 1 hadoop supergroup 15 2024-05-13 14:40 /Lab05/test.txt
-rw-r--r-- 1 hadoop supergroup 19 2024-05-13 14:33 /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -getmerge /Lab05 /text.txt /Lab05 /test.txt ../Downloads/Merged.txt
getmerge: '/text.txt': No such file or directory
getmerge: '/test.txt': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -getmerge /Lab05/text.txt /Lab05/test.txt ../Downloads/Merged.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -getfacl /Lab05
# file: /Lab05
# owner: hadoop
# group: supergroup
user::rwx
group::r-x
other::r-x
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -put /home/hadoop/Desktop/Welcome.txt /abc/WC.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyFromLocal /home/hadoop/Desktop/Welcome.txt /abc/WC.txt
copyFromLocal: '/abc/WC.txt': File exists
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -get /abc/WC.txt /home/hadoop/Downloads/WWC.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -getmerge /abc/ /home/hadoop/Desktop/Merge.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -getfacl /abc/
# file: /abc
# owner: hadoop
# group: supergroup
user::rwx
group::r-x
other::r-x

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /abc/WC.txt
hello world
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -mv /abc /FFF
hadoop fs -ls /FFF
Found 3 items
-rw-r--r-- 1 hadoop supergroup 12 2025-04-15 14:53 /FFF/WC.txt
-rw-r--r-- 1 hadoop supergroup 12 2024-05-14 14:35 /FFF/file.txt
-rw-r--r-- 1 hadoop supergroup 12 2024-05-14 14:38 /FFF/file_cp_local.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -cp /CSE/ /LLL
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab05/text.txt ..//Documents
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab05/test.txt ..//Documents
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /Lab05/text.txt
Hello
How are you?
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mv /Lab05 /test_Lab05
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /test_Lab05
Found 2 items
-rw-r--r-- 1 hadoop supergroup 15 2024-05-13 14:40 /test_Lab05/test.txt
-rw-r--r-- 1 hadoop supergroup 19 2024-05-13 14:33 /test_Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cp /test_Lab05/ /Lab05
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /Lab05
Found 2 items
-rw-r--r-- 1 hadoop supergroup 15 2024-05-13 14:51 /Lab05/test.txt
-rw-r--r-- 1 hadoop supergroup 19 2024-05-13 14:51 /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /test_Lab05
Found 2 items
-rw-r--r-- 1 hadoop supergroup 15 2024-05-13 14:40 /test_Lab05/test.txt
-rw-r--r-- 1 hadoop supergroup 19 2024-05-13 14:33 /test_Lab05/text.txt
```

# LABORATORY PROGRAM – 5

## Implement Wordcount program on Hadoop framework

### OBSERVATION

### CODE, COMMAND WITH OUTPUT

#### **Driver Code**

```
// Importing libraries
import java.io.IOException;
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.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 Configuration 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);
        conf.setJobName("WordCount");

        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("Job Exit Code: " + exitCode);
    }
}
```

#### **Mapper Code**

```
// Importing libraries
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 reporter)
        throws IOException {
        String line = value.toString();

        // Splitting the line on whitespace
        for (String word : line.split("\\s+")) {
            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;

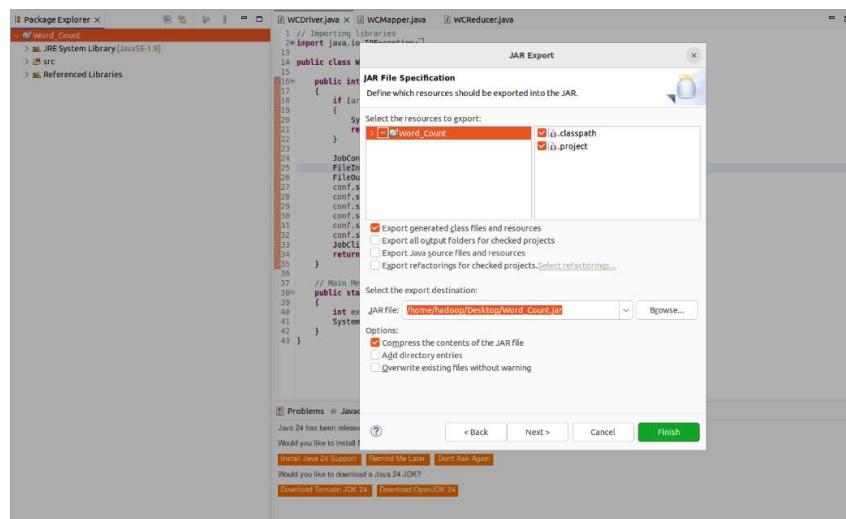
public class WCReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> values,
                       OutputCollector<Text, IntWritable> output,
                       Reporter reporter) throws IOException {
        int count = 0;

        // Counting the frequency of each word
        while (values.hasNext()) {
            count += values.next().get();
        }

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

```



```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mkdir /Lab06
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab06
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ jps
7360 DataNode
7928 ResourceManager
8681 Jps
7178 NameNode
8091 NodeManager
7644 SecondaryNameNode
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ cd ..
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano file1.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -copyFromLocal -f /home/hadoop/Desktop/file1.txt /rgs/test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar /home/hadoop/Desktop/WordCount.jar wordcount.WordCount /rgs/test.txt /output
JAR does not exist or is not a normal file: /home/hadoop/Desktop/WordCount.jar
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar /home/hadoop/Desktop/Word_Count.jar wordcount.WordCount /rgs/test.txt /output
Exception in thread "main" java.lang.ClassNotFoundException: wordcount.WordCount
        at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:476)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:594)
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:527)
        at java.base/java.lang.Class.forName0(Native Method)
        at java.base/java.lang.Class.forName(Class.java:398)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:321)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -cat /output/part-00000
are      1
brother  1
family   1
hi       1
how     5
is      4
job      1
sister  1
you      1
your     4
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /output
Found 2 items
-rw-r--r--  1 hadoop supergroup          0 2024-05-21 15:21 /output/_SUCCESS
-rw-r--r--  1 hadoop supergroup         69 2024-05-21 15:21 /output/part-00000
```

# LABORATORY PROGRAM – 6

## Implement Weather program on Hadoop framework

### Questions:

From the following link extract the weather data

<https://github.com/tomwhite/hadoopbook/tree/master/input/ncdc/all>

- Create a MapReduce program to find average temperature for each year from NCDC data set.
- find the mean max temperature for every month.

### OBSERVATION

### CODE, COMMAND WITH OUTPUT – A

#### **Driver Code**

```
package temp;

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.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 both input and output parameters.");
            System.exit(-1);
        }

        // Creating a configuration and job instance
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Average Calculation");

        job.setJarByClass(AverageDriver.class);

        // Input and output paths
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        // Setting mapper and reducer classes
        job.setMapperClass(AverageMapper.class);
        job.setReducerClass(AverageReducer.class);

        // Output key and value types
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        // Submitting the job and waiting for it to complete
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

## Mapper Code

```
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;

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

        String line = value.toString();

        // Extract year from fixed position
        String year = line.substring(15, 19);
        int temperature;

        // Determine if there's a '+' sign
        if (line.charAt(87) == '+') {
            temperature = Integer.parseInt(line.substring(88, 92));
        } else {
            temperature = Integer.parseInt(line.substring(87, 92));
        }

        // Quality check character
        String quality = line.substring(92, 93);

        // Only emit if data is valid
        if (temperature != MISSING && quality.matches("[01459]")) {
            context.write(new Text(year), new IntWritable(temperature));
        }
    }
}
```

## Reducer Code

```
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> {

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

        int sumTemp = 0;
        int count = 0;

        for (IntWritable value : values) {
            sumTemp += value.get();
            count++;
        }

        if (count > 0) {
            int average = sumTemp / count;
            context.write(key, new IntWritable(average));
        }
    }
}
```

}

Name	Size	Type	Modified
META-INF	25 bytes	Folder	
.classpath	2.2 kB	unknown	06 May 2025, 14:40
.project	377 bytes	unknown	06 May 2025, 14:34
AverageDriver.class	1.6 kB	Java class	06 May 2025, 14:42
AverageMapper.class	2.4 kB	Java class	06 May 2025, 14:42
AverageReducer.class	2.3 kB	Java class	06 May 2025, 14:42

```
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscscse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ jps
7056 DataNode
7332 SecondaryNameNode
7638 ResourceManager
8231 Jps
5883 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
7804 NodeManager
6877 NameNode
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -ls /\
> ^C
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -ls /
Found 4 items
drwxr-xr-x - hadoop supergroup 0 2025-04-15 15:00 /FFF
drwxr-xr-x - hadoop supergroup 0 2025-04-15 15:34 /LLL
drwxr-xr-x - hadoop supergroup 0 2024-05-13 14:46 /file
drwxr-xr-x - hadoop supergroup 0 2024-05-13 15:18 /newDataFlair
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -ls /weather
ls: `/weather': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -mkdir /weather
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -copyFromLocal /home/hadoop/Desktop/1901.txt /weather/test.txt

hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: $ Hadoop jar /home/hadoop/Desktop/AverageTemperature.jar AverageDriver /weather/test.txt /weather/output
2025-05-06 14:59:23.239 INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat: Input split for job: /weather/test.txt
2025-05-06 14:59:23.279 INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat: Scheduled Metrics snapshot period at 10 seconds(s).
2025-05-06 14:59:23.279 INFO org.apache.hadoop.metrics.sink.impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-06 14:59:23.348 WARN org.apache.hadoop.mapreduce.lib.input.FileInputFormat: Total Input splits to process : 1
2025-05-06 14:59:23.348 INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat: Input splits to process : 1
2025-05-06 14:59:23.487 INFO org.apache.hadoop.mapreduce.JobSubmissionHandler: Submitting tokens for job: job_local191822813_0001
2025-05-06 14:59:23.488 INFO org.apache.hadoop.mapreduce.JobSubmissionHandler: Executing with tokens: []
2025-05-06 14:59:23.566 INFO org.apache.hadoop.mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-06 14:59:23.566 INFO org.apache.hadoop.mapreduce.Job: Running job: job_local191822813_0001
2025-05-06 14:59:23.566 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: Output committer null
2025-05-06 14:59:23.564 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 14:59:23.565 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 14:59:23.565 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: Output cleanup _temporary Folders under output directory:false, ignore cleanup failures: false
2025-05-06 14:59:23.565 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: Output committer class org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-06 14:59:23.693 INFO org.apache.hadoop.mapreduce.lib.output.LocalJobRunner: Starting task: attempt_local191822813_0001_m_000000_0
2025-05-06 14:59:23.615 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 14:59:23.615 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 14:59:23.615 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary Folders under output directory:false, ignore cleanup failures: false
2025-05-06 14:59:23.615 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: Output committer null
2025-05-06 14:59:23.624 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: Processing split: hdfs://localhost:9000/weather/test.txt:0+888190
2025-05-06 14:59:23.658 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: (EQUATOR) 0 kv1 26214396(104857584)
2025-05-06 14:59:23.658 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: mapreduce.task.io.sort.mb: 10
2025-05-06 14:59:23.658 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: mapreduce.task.io.sort.size: 104857600
2025-05-06 14:59:23.658 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: mapreduce.task.io.sort.start = 0; bufSize = 104857600
2025-05-06 14:59:23.658 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: mapreduce.task.io.sort.length = 6553600
2025-05-06 14:59:23.666 INFO org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
```

```
2025-05-06 14:59:24,581 INFO mapreduce.Job: Counters: 36
  File System Counters
    FILE: Number of bytes read=153118
    FILE: Number of bytes written=1493804
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=1776380
    HDFS: Number of bytes written=8
    HDFS: Number of read operations=15
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=4
    HDFS: Number of bytes read erasure-coded=0
  Map-Reduce Framework
    Map input records=6565
    Map output records=6564
    Map output bytes=59076
    Map output materialized bytes=72210
    Input split bytes=103
    Combine input records=0
    Combine output records=0
    Reduce input groups=1
    Reduce shuffle bytes=72210
    Reduce input records=6564
    Reduce output records=1
    Spilled Records=13128
    Shuffled Maps =1
    Failed Shuffles=0
    Merged Map outputs=1
    GC time elapsed (ms)=0
    Total committed heap usage (bytes)=1266679808
  Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=888190
  File Output Format Counters
    Bytes Written=8
```

```
Bytes Written=8
hadoop@bmscsecsse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /weather
Found 2 items
drwxr-xr-x  - hadoop supergroup          0 2025-05-06 14:59 /weather/output
-rw-r--r--  1 hadoop supergroup     888190 2025-05-06 14:50 /weather/test.txt
hadoop@bmscsecsse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /weather/output
Found 2 items
-rw-r--r--  1 hadoop supergroup          0 2025-05-06 14:59 /weather/output/_SUCCESS
-rw-r--r--  1 hadoop supergroup          8 2025-05-06 14:59 /weather/output/part-r-00000
hadoop@bmscsecsse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /weather/output/part-r-00000
1901 46
hadoop@bmscsecsse-HP-Elite-Tower-800-G9-Desktop-PC:~$ 
```

## OBSERVATION

### CODE, COMMAND WITH OUTPUT – B

#### Driver Code

```
package meanmax;

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.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 both input and output parameters.");
            System.exit(-1);
        }

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Mean and Max Temperature");

        job.setJarByClass(MeanMaxDriver.class);

        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);
    }
}
```

#### Mapper Code

```
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;

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

        String line = value.toString();
```

```

// Extract month from positions 19-20
String month = line.substring(19, 21);
int temperature;

// Extract temperature considering optional '+'
if (line.charAt(87) == '+') {
    temperature = Integer.parseInt(line.substring(88, 92));
} else {
    temperature = Integer.parseInt(line.substring(87, 92));
}

// Quality check
String quality = line.substring(92, 93);

if (temperature != MISSING && quality.matches("[01459]")) {
    context.write(new Text(month), new IntWritable(temperature));
}
}
}

```

### Reducer Code

```

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, Text> {

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

        int sumTemp = 0;
        int count = 0;
        int maxTemp = Integer.MIN_VALUE;

        for (IntWritable value : values) {
            int temp = value.get();
            sumTemp += temp;
            count++;
        }

        if (count > 0) {
            int avgTemp = sumTemp / count;
            String result = "mean=" + avgTemp + " max=" + maxTemp;
            context.write(key, new Text(result));
        }
    }
}

```

**Package Explorer**

Min\_Max\_Temp

JRE System Library [JavaSE-1.8]

src

(default package)

MMDriver.java

MMMapper.java

MMReducer.java

```

hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
localhost: namenode is running as process 5478. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 5644. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [hnsccese-HP-Elite-Tower-800-G9-Desktop-PC]
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: secondarynamenode is running as process 5931. Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
Starting resourcemanager is running as process 6214. Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 6375. Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hdfs dfs -copyFromLocal /home/hadoop/Desktop/1991 /rgs/temp
copyFromLocal: /home/hadoop/Desktop/1991 does not exist.
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hdfs dfs -copyFromLocal /home/hadoop/Desktop/1991 /rgs/1993
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop jar /home/hadoop/Downloads/Mean_Max_Temp.jar MMDriver /rgs/avtemp.txt /out8
JAR does not exist or is not a normal file: /home/hadoop/Downloads/Mean_Max_Temp.jar
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop jar /home/hadoop/Downloads/Mean_Max_Temp.jar MMDriver /rgs/avtemp.txt /out8
JAR does not exist or is not a normal file: /home/hadoop/Downloads/Mean_Max_Temp.jar
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop jar /home/hadoop/Desktop/Mean_Max_Temp.jar MMDriver /rgs/avtemp.txt /out8
2025-05-06 15:23:05.430 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-06 15:23:05.471 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-06 15:23:05.471 INFO impl.MetricsSystemImpl: Metrics system has been started
2025-05-06 15:23:05.531 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-06 15:23:05.575 INFO mapreduce.JobSubmitter: Cleaning up the staging area file:/tmp/hadoop/mapred/staging/hadoop1762985270/.staging/job_local1762985270_0001
Exception in thread "main" org.apache.hadoop.mapreduce.lib.input.InvalidInputException: Input path does not exist: hdfs://localhost:9000/rgs/avtemp.txt
    at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:340)
    at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.listStatus(FileInputFormat.java:279)
    at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.getPaths(FileInputFormat.java:404)
    at org.apache.hadoop.mapreduce.JobSubmitter.writeNewSplits(JobSubmitter.java:319)
    at org.apache.hadoop.mapreduce.JobSubmitter.writeSplits(JobSubmitter.java:327)
    at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:200)
    at org.apache.hadoop.mapreduce.Job$1.run(Job.java:1078)
    at org.apache.hadoop.mapreduce.Job$1.run(Job.java:1075)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1146)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
    at java.lang.Thread.run(Thread.java:748)
    at java.base/java.security.AccessController.doPrivileged(Native Method)
    at java.base/javax.security.auth.Subject.doAs(Subject.java:423)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1899)
    at org.apache.hadoop.mapreduce.Job.submit(Job.java:1075)
    at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1096)
    at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1046)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(Native Method)
    at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.base/java.lang.reflect.Method.invoke(Method.java:566)
    at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
Caused by: java.io.IOException: Input path does not exist: hdfs://localhost:9000/rgs/avtemp.txt
    at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:313)
...
19 more
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hdfs dfs -cat /out8/*
cat: /out8/*: No such file or directory
hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hdfs dfs -ls /

```

Caused by: java.io.IOException: Input path does not exist: hdfs://localhost:9000/rgs/avtemp.txt
 at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:313)

```

hadoop@hnsccese-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop jar /home/hadoop/Desktop/Mean_Max_Temp.jar MMDriver /rgs/1993 /out8
2025-05-06 15:26:34.876 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-06 15:26:34.916 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-06 15:26:34.916 INFO impl.MetricsSystemImpl: Metrics system has been started
2025-05-06 15:26:34.976 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-06 15:26:35.029 INFO input.FileInputFormat: Total input files to process : 1
2025-05-06 15:26:35.081 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-06 15:26:35.148 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local063792118_0001
2025-05-06 15:26:35.150 INFO mapreduce.JobSubmitter: Starting task attempt_local063792118_0001_m_000000_0
2025-05-06 15:26:35.715 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-06 15:26:35.216 INFO mapreduce.Job: Running job: job_local063792118_0001
2025-05-06 15:26:35.220 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 15:26:35.221 INFO output.FileOutputCommitter: OutputCommitter set in config null
2025-05-06 15:26:35.221 INFO output.FileOutputCommitter: OutputCommitter algorithm version is 2
2025-05-06 15:26:35.221 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 15:26:35.221 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-06 15:26:35.267 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-06 15:26:35.268 INFO mapred.LocalJobRunner: Starting task attempt_local063792118_0001_m_000000_0
2025-05-06 15:26:35.270 INFO mapred.MapTask: Using Committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 15:26:35.278 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 15:26:35.279 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 15:26:35.286 INFO mapred.Task: Using ResourceCalculatorProcessTree : []
2025-05-06 15:26:35.287 INFO mapred.Task: Processing Map Task attempt_local063792118_0001_m_000000_0
2025-05-06 15:26:35.321 INFO mapred.Task: Task attempt_local063792118_0001_m_000000_0 failed with exitValue 1
2025-05-06 15:26:35.322 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-06 15:26:35.322 INFO mapred.MapTask: soft lmtt at 83886080
2025-05-06 15:26:35.322 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-06 15:26:35.325 INFO mapred.MapTask: map output collector class = org.apache.hadoop.mapred.MapOutputBuffer
2025-05-06 15:26:35.440 INFO mapred.LocalJobRunner:
2025-05-06 15:26:35.441 INFO mapred.MapTask: Starting flush of map output
2025-05-06 15:26:35.441 INFO mapred.MapTask: Spilling map output
2025-05-06 15:26:35.441 INFO mapred.MapTask: bufstart = 20214396; bufend = 45948; bufvoid = 104857600
2025-05-06 15:26:35.441 INFO mapred.MapTask: Kvend = 26188144(104752576); length = 26253/6553600
2025-05-06 15:26:35.451 INFO mapred.MapTask: Flushed spill 0
2025-05-06 15:26:35.456 INFO mapred.Task: Task attempt_local063792118_0001_m_000000_0 is done. And is in the process of committing
2025-05-06 15:26:35.458 INFO mapred.LocalJobRunner: map
2025-05-06 15:26:35.458 INFO mapred.Task: Task attempt_local063792118_0001_m_000000_0 done.
2025-05-06 15:26:35.461 INFO mapred.Task: Final Counters for attempt_local063792118_0001_m_000000_0: Counters: 23
    File system counters
        FILE: Number of bytes read=4359
        FILE: Number of bytes written=1993
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=888190
        HDFS: Number of read operations=5
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=1
        HDFS: Number of bytes read erasure-coded=0
    Map-Reduce Framework
        Map-Reduce counters

```

```
2025-05-06 15:26:36,233 INFO mapreduce.Job: Counters: 36
  File System Counters
    FILE: Number of bytes read=126914
    FILE: Number of bytes written=1466688
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=1776380
    HDFS: Number of bytes written=74
    HDFS: Number of read operations=15
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=4
    HDFS: Number of bytes read erasure-coded=0
  Map-Reduce Framework
    Map input records=6565
    Map output records=6564
    Map output bytes=45948
    Map output materialized bytes=59082
    Input split bytes=95
    Combine input records=0
    Combine output records=0
    Reduce input groups=12
    Reduce shuffle bytes=59082
    Reduce input records=6564
    Reduce output records=12
    Spilled Records=13128
    Shuffled Maps =1
    Failed Shuffles=0
    Merged Map outputs=1
    GC time elapsed (ms)=0
    Total committed heap usage (bytes)=1052770304
  Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=888190
  File Output Format Counters
    Bytes Written=74
```

```
hadoop@bmscsece-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /out8/*
```

```
01      4
02      0
03      7
04     44
05     100
06     168
07     219
08     198
09     141
10     100
11      19
12      3
```

## LABORATORY PROGRAM – 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.**

### OBSERVATION

### CODE, COMMAND WITH OUTPUT

#### Driver Code (TopNDriver.java)

```
package samples.topn;

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

public class TopNDriver {

    public static void main(String[] args) throws Exception {
        if (args.length != 3) {
            System.err.println("Usage: TopNDriver <in> <temp-out> <final-out>");
            System.exit(2);
        }

        Configuration conf = new Configuration();

        //==== Job 1: Word Count ====
        Job wcJob = Job.getInstance(conf, "word count");
        wcJob.setJarByClass(TopNDriver.class);
        wcJob.setMapperClass(WordCountMapper.class);
        wcJob.setCombinerClass(WordCountReducer.class);
        wcJob.setReducerClass(WordCountReducer.class);
        wcJob.setOutputKeyClass(Text.class);
        wcJob.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(wcJob, new Path(args[0]));
        Path tempDir = new Path(args[1]);
        FileOutputFormat.setOutputPath(wcJob, tempDir);

        if (!wcJob.waitForCompletion(true)) {
            System.exit(1);
        }

        //==== Job 2: Top N ====
        Job topJob = Job.getInstance(conf, "top 10 words");
        topJob.setJarByClass(TopNDriver.class);
        topJob.setMapperClass(TopNMapper.class);
        topJob.setReducerClass(TopNReducer.class);
        topJob.setMapOutputKeyClass(IntWritable.class);
        topJob.setMapOutputValueClass(Text.class);
        topJob.setOutputKeyClass(Text.class);
        topJob.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(topJob, tempDir);
        FileOutputFormat.setOutputPath(topJob, new Path(args[2]));
    }
}
```

```
        System.exit(topJob.waitForCompletion(true) ? 0 : 1);  
    }  
}
```

## Mapper Code (WordCountMapper.java)

```
package samples.topn;

import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class WordCountMapper
    extends Mapper<Object, Text, Text, IntWritable> {
```

```
@Override
protected void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {

    // clean & tokenize
    String clean = value.toString()
        .toLowerCase()
        .replaceAll(tokens, " ");
    StringTokenizer itr = new StringTokenizer(clean);
    while (itr.hasMoreTokens()) {
        word.set(itr.nextToken().trim());
        context.write(word, ONE);
    }
}
```

## Mapper Code (TopNMapper.java)

```
package samples.topn;
```

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

public class TopNMapper
    extends Mapper<Object, Text, IntWritable, Text> {

    private IntWritable count = new IntWritable();
    private Text word = new Text();

    @Override
    protected void map(Object key, Text value, Context context)
        throws IOException, InterruptedException {
        // input line: word \t count
        String[] parts = value.toString().split("\t");
        if (parts.length == 2) {
            word.set(parts[0]);
            count.set(Integer.parseInt(parts[1]));
            // emit count → word, so Hadoop sorts by count
            context.write(count, word);
        }
    }
}
```

### **Reducer Code (WordCountReducer.java)**

```
package samples.topn;

import java.io.IOException;

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

public class WordCountReducer
    extends Reducer<Text, IntWritable, Text, IntWritable> {

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

        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
}
```

### **Reducer Code (TopNReducer.java)**

```
package samples.topn;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Map;
import java.util.TreeMap;

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

public class TopNReducer
    extends Reducer<IntWritable, Text, Text, IntWritable> {

    // TreeMap with descending order of keys (counts)
    private TreeMap<Integer, List<String>> countMap =
        new TreeMap<>(Collections.reverseOrder());

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

        int cnt = key.get();
        List<String> words = countMap.getOrDefault(cnt, new ArrayList<>());
        for (Text w : values) {
            words.add(w.toString());
        }
        countMap.put(cnt, words);
    }

    @Override
    protected void cleanup(Context context)
        throws IOException, InterruptedException {

        // collect top 10 word→count pairs
        List<WordCount> topList = new ArrayList<>();
        int seen = 0;
        for (Map.Entry<Integer, List<String>> entry : countMap.entrySet()) {
            int cnt = entry.getKey();
            for (String w : entry.getValue()) {
```

```
topList.add(new WordCount(w, cnt));
seen++;
if(seen == 10) break;
}
if(seen == 10) break;
}

// sort these 10 entries alphabetically by word
Collections.sort(topList, (a, b) -> a.word.compareTo(b.word));

// emit final top 10 in alphabetical order
for (WordCount wc : topList) {
    context.write(new Text(wc.word), new IntWritable(wc.count));
}
}

// helper class
private static class WordCount {
    String word;
    int count;
    WordCount(String w, int c) { word = w; count = c; }
}
}
```

```

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 /input_dir/input.txt /output_dir
2021-05-08 19:54:54,582 INFO client.DefaultNoHANMFalloverProxyProvider: 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 19:54:57,587 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1620483374279_0001/
2021-05-08 19:54:57,588 INFO mapreduce.Job: Running job: job_1620483374279_0001
2021-05-08 19:55:13,792 INFO mapreduce.Job: Job job_1620483374279_0001 running in uber mode : false
2021-05-08 19:55:13,794 INFO mapreduce.Job: map 0% reduce 0%
2021-05-08 19:55:20,020 INFO mapreduce.Job: map 100% reduce 0%
2021-05-08 19:55:27,116 INFO mapreduce.Job: map 100% reduce 100%
2021-05-08 19:55:33,199 INFO mapreduce.Job: Job job_1620483374279_0001 completed successfully
2021-05-08 19:55:33,334 INFO mapreduce.Job: Counters: 54
    File System Counters
        FILE: Number of bytes read=65
        FILE: Number of bytes written=530397
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=142
        HDFS: Number of bytes written=31
        HDFS: Number of read operations=8
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
        HDFS: Number of bytes read erasure-coded=0

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

C:\hadoop-3.3.0\sbin>

```

## **LABORATORY PROGRAM – 8**

**Write a Scala program to print numbers from 1 to 100 using for loop.**

## OBSERVATION

## CODE, COMMAND WITH OUTPUT

## LABORATORY PROGRAM – 9

**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.**

### OBSERVATION

### CODE, COMMAND WITH OUTPUT

```
scala> val rdd = spark.sparkContext.textFile("file:/home/bmscecse/Desktop/scala")
rdd: org.apache.spark.rdd.RDD[String] = file:/home/bmscecse/Desktop/scala MapPartitionsRDD[1] at textFile at <console>:23

scala> val counts = rdd.flatMap(_.split("\\s+")).map(word => (word.toLowerCase, 1)).reduceByKey(_ + _).filter(_._2 > 4)
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:25

scala> counts.collect().foreach{ case (word, count) => println(s"$word $count") }
spark 6

scala>
```

## LABORATORY PROGRAM – 10

**Write a simple streaming program in Spark to receive text data streams on a particular port, perform basic text cleaning (like white space removal, stop words removal, lemmatization, etc.), and print the cleaned text on the screen. (Open Ended Question).**

### OBSERVATION

### CODE, COMMAND WITH OUTPUT

```
# Install NLTK and download required data (run once)
!pip install nltk

import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

from pyspark.sql import SparkSession
from pyspark.sql.functions import col, lower, regexp_replace, split, explode, udf
from pyspark.sql.types import ArrayType, StringType
from pyspark.ml.feature import StopWordsRemover
from nltk.stem import WordNetLemmatizer

# Initialize SparkSession
spark = SparkSession.builder.appName("TextProcessing").getOrCreate()

# Define your input lines
lines = [
    "Hello, I hate you.",
    "I hate that I love you.",
    "Don't want to, but I can't put",
    "nobody else above you."
]

# Create DataFrame from lines
df = spark.createDataFrame(lines, "string").toDF("value")

# Step 1: Lowercase and remove punctuation
df_clean = df.select(regexp_replace(lower(col("value")), "[^a-zA-Z\\s]", "")).alias("cleaned"))

# Step 2: Tokenize the cleaned text
df_tokens = df_clean.select(split(col("cleaned"), "\\s+").alias("tokens"))

# Step 3: Remove stop words
remover = StopWordsRemover(inputCol="tokens", outputCol="filtered")
df_filtered = remover.transform(df_tokens)

# Step 4: Lemmatization using NLTK WordNetLemmatizer with UDF
lemmatizer = WordNetLemmatizer()

def lemmatize_words(words):
    return [lemmatizer.lemmatize(word) for word in words]

lemmatize_udf = udf(lemmatize_words, ArrayType(StringType()))
```

```
df_lemmatized = df_filtered.withColumn("lemmatized", lemmatize_udf(col("filtered")))

# Step 5: Explode the lemmatized words and show results
df_lemmatized.select(explode(col("lemmatized")).alias("word")).show(truncate=False)
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.2.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.5.0)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]  Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
+-----+
|word |
+-----+
|hello |
|hate |
|hate |
|love |
|dont |
|want |
|cant |
|put |
|nobody|
|else |
+-----+
```