

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

### Big Data Analytics (23CS6PCBDA)

*Submitted by*

**Manvendra Singh Chouhan(1BM22CS148)**

*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 Manvendra Singh Chouhan (1BM22CS148), who is bonafide student of B. M. S. College of Engineering. It is in partial fulfilment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2025. 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.

Amruta  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

Dr. Kavitha Sooda  
Professor and Head  
Department of CSE  
BMSCE, Bengaluru

## Index Sheet

Sl. No.	Experiment Title	Page No.
1	MongoDB- CRUD Demonstration.	1-4
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 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.	5-6
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 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	7-8
4	Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)	9-10
5	Implement Wordcount program on Hadoop framework	11-14
6	From the following link extract the weather data <a href="https://github.com/tomwhite/hadoop">https://github.com/tomwhite/hadoop</a> book/tree/master/input/ncdc/all Create a Map Reduce program to a) find average temperature for each year from NCDC data set. b) find the mean max temperature for every month.	15-23
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.	24-30
8	Write a Scala program to print numbers from 1 to 100 using for loop.	31
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.	32-33

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).	34-35
----	--	-------

## Course Outcome

<b>CO1</b>	Apply the concept of NoSQL, Hadoop or Spark for a given task
<b>CO2</b>	Analyze big data analytics mechanisms that can be applied to obtain solution for a given problem.
<b>CO3</b>	Design and implement solutions using data analytics mechanisms for a given problem.

## Experiment-1

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

### Code & Output:

1. Create a database “Student” with the following attributes Rollno, Name , Age, ContactNo, Email-Id, grade, hobby:

use Students;

2. Insert 5 appropriate values according to the below queries.

```
db.students.insertMany([
```

```
{ "Rollno": 10, "Name": "John", "Age": 20, "ContactNo": "1234567890", "Email-Id":  
"john@example.com", "grade": "A", "hobby": "Reading" },
```

```
{ "Rollno": 11, "Name": "Alice", "Age": 21, "ContactNo": "9876543210", "Email-Id":  
"alice@example.com", "grade": "B", "hobby": "Painting" },
```

```
{ "Rollno": 12, "Name": "Bob", "Age": 22, "ContactNo": "2345678901", "Email-Id":  
"bob@example.com", "grade": "C", "hobby": "Cooking" },
```

```
{ "Rollno": 13, "Name": "Eve", "Age": 23, "ContactNo": "3456789012", "Email-Id":  
"eve@example.com", "grade": "A" },
```

```
{ "Rollno": 14, "Name": "Charlie", "Age": 24, "ContactNo": "4567890123", "Email-Id":  
"charlie@example.com", "hobby": "Gardening" }])
```

```

Atlas atlas-wanmtx-shard-0 [primary] Student> use Students
switched to db Students
Atlas atlas-wanmtx-shard-0 [primary] Students> show collections

Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.insertMany([
...   { "Rollno": 10, "Name": "John", "Age": 20, "ContactNo": "1234567890", "Email-Id":
"john@example.com", "grade": "A", "hobby": "Reading" },
...   { "Rollno": 11, "Name": "Alice", "Age": 21, "ContactNo": "9876543210", "Email-Id":
"alice@example.com", "grade":
"B", "hobby": "Painting" },
...   { "Rollno": 12, "Name": "Bob", "Age": 22, "ContactNo": "2345678901", "Email-Id": "
bob@example.com", "grade": "C", "hobby": "Cooking" },
...   { "Rollno": 13, "Name": "Eve", "Age": 23, "ContactNo": "3456789012", "Email-Id": "
eve@example.com", "grade": "A"
},
...   { "Rollno": 14, "Name": "Charlie", "Age": 24, "ContactNo": "4567890123", "Email-Id
": "charlie@example.com", "hobby": "Gardening" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("661ce9dc76a00ff8cc51dae1"),
    '1': ObjectId("661ce9dc76a00ff8cc51dae2"),
    '2': ObjectId("661ce9dc76a00ff8cc51dae3"),
    '3': ObjectId("661ce9dc76a00ff8cc51dae4"),
    '4': ObjectId("661ce9dc76a00ff8cc51dae5")
  }
}

```

3. Write query to update Email-Id of a student with rollno 10.

```

db.students.updateOne(
{ "Rollno": 10 },
{ $set: { "Email-Id": "john.doe@example.com" } }
)

```

```

Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.updateOne(
...   { "Rollno": 10 },
...   { $set: { "Email-Id": "john.doe@example.com" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

4. Replace the student name from “Alice” to “Alicee” of rollno 11

```
db.students.updateOne(
  { "Rollno": 11 },
  { $set: { "Name": "Alice" } }
)
```

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.updateOne(
...   { "Rollno": 11 },
...   { $set: { "Name": "Alice" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

5. Display Student Name and grade(Add if grade is not present)where the \_id column is 1.

```
db.students.find({}, { "Name": 1, "grade": { $ifNull: ["$grade", "Not available"] }, "_id": 0
})
```

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.find({}, { "Name": 1, "grade":
{ $ifNull: ["$grade", "Not available"] }, "_id": 0 })
[
  { Name: 'John', grade: 'A' },
  { Name: 'Alice', grade: 'B' },
  { Name: 'Bob', grade: 'C' },
  { Name: 'Eve', grade: 'A' },
  { Name: 'Charlie', grade: 'Not available' }
]
```

6. Update to add hobbies

```
db.students.updateMany(
  { "Name": "Eve" },
  { $set: { "hobby": "Dancing" } })
```

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.updateMany(
...   { "Name": "Eve" },
...   { $set: { "hobby": "Dancing" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

7. Find documents where hobbies is set neither to Chess nor to Skating

```
db.students.find({ "hobby": { $nin: ["Chess", "Skating"] } })
```

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.find({ "hobby": { $nin: ["Chess", "Skating"] } })
[
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae1"),
    Rollno: 10,
    Name: 'John',
    Age: 20,
    ContactNo: '1234567890',
    'Email-Id': 'john.doe@example.com',
    grade: 'A',
    hobby: 'Reading'
  },
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae2"),
    Rollno: 11,
    Name: 'Alice',
    Age: 21,
    ContactNo: '9876543210',
    'Email-Id': 'alice@example.com',
    grade: 'B',
    hobby: 'Painting'
  },
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae3"),
    Rollno: 12,
    Name: 'Bob',
    Age: 22,
    ContactNo: '2345678901',
    'Email-Id': 'bob@example.com',
    grade: 'C',
    hobby: 'Cooking'
  },
]
```

8. Find documents whose name begins with A

```
db.students.find({ "Name": /^A/ })
```

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.find({ "Name": /^A/ })
[
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae2"),
    Rollno: 11,
    Name: 'Alice',
    Age: 21,
    ContactNo: '9876543210',
    'Email-Id': 'alice@example.com',
    grade: 'B',
    hobby: 'Painting'
  }
]
```



## Experiment – 2

Q) 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  
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

### Code & Output:

```
cqlsh> DESCRIBE KEYSPACES

employee  system_auth      system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

cqlsh> CREATE TABLE IF NOT EXISTS Employee_Info(
... Emp_Id INT PRIMARY KEY,
... Emp_name TEXT,
... designation TEXT,
... date_of_joining DATE,
... Salary FLOAT,
... Dep_name TEXT,
... Projects SET<TEXT>;
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> USE Employee
...
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 FLOAT, Dep_name TEXT, Projects SET<TEXT>;
cqlsh:employee> describe keyspace Employee

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

CREATE TABLE employee.employee_info (
  emp_id int PRIMARY KEY,
  date_of_joining date,
  dep_name text,
  designation text,
  emp_name text,
  salary float,
  projects set<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 mentable = '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 mentable_flush_period_in_ms = 0
AND min_index_interval = 128
```

```
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqlsh:employee> select * from employee_info;
```

emp_id	bonus	date_of_joining	dep_name	designation	emp_name	projects	salary
120	12000	2024-05-06	Engineering	Developer	Priyanka GH	{'Project B', 'ProjectA'}	1e+06
123	null	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	null	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	11000	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	0

(4 rows)

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

emp_id	bonus	date_of_joining	dep_name	designation	emp_name	projects	salary
120	12000	2024-05-06	Engineering	Developer	Priyanka GH	{'Project B', 'ProjectA'}	1e+06
123	null	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	null	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	11000	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	null

(4 rows)

```
cqlsh:employee>
```

```
AND speculative_retry = '99p';
```

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

emp_id	date_of_joining	dep_name	designation	emp_name	projects	salary
120	2024-05-06	Engineering	Developer	Priyanka	{'Project B', 'ProjectA'}	1e+06
123	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	9e+05

(4 rows)

```
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id = '120';
```

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid STRING constant (120) for "emp_id" of type int"
```

```
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id=120;
```

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

emp_id	date_of_joining	dep_name	designation	emp_name	projects	salary
120	2024-05-06	Engineering	Developer	Priyanka GH	{'Project B', 'ProjectA'}	1e+06
123	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	9e+05

(4 rows)

```
cqlsh:employee> select * from employee_info order by salary;
```

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="ORDER BY is only supported when the partition key is restricted by an EQ or an IN."
```

```
cqlsh:employee> alter table employee_info add bonus INT;
```

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

emp_id	bonus	date_of_joining	dep_name	designation	emp_name	projects	salary
120	null	2024-05-06	Engineering	Developer	Priyanka GH	{'Project B', 'ProjectA'}	1e+06
123	null	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	null	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	null	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	9e+05

(4 rows)

```
cqlsh:employee> update employee_info set bonus = 12000 where emp_id = 120;
```

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

emp_id	bonus	date_of_joining	dep_name	designation	emp_name	projects	salary
120	12000	2024-05-06	Engineering	Developer	Priyanka GH	{'Project B', 'ProjectA'}	1e+06
123	null	2024-05-07	Engineering	Engineer	Sadhana	{'Project M', 'Project P'}	1.2e+06
122	null	2024-05-06	Management	HR	Rachana	{'Project C', 'Project M'}	9e+05
121	null	2024-05-06	Management	Developer	Shreya	{'Project C', 'ProjectA'}	9e+05

(4 rows)

```
cqlsh:employee> update employee_info set bonus = 11000 where emp_id = 121;
```

```
cqlsh:employee> select * from employee_info using ttl 15 where emp_id = 123;
```

```
SyntaxException: line 1:28 mismatched input 'using' expecting EOF (select * from employee_info [using] ttl...)
```

```
cqlsh:employee> select * from employee_info where emp_id = 121 using ttl 15;
```

```
SyntaxException: line 1:47 no viable alternative at input 'using' (...employee_info where emp_id = 121 [using]...)
```

```
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
```

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

## Experiment – 3

Q) Perform the following DB operations using Cassandra

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

### Code & Output:

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Students WITH REPLICATION={
... 'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES

students  system_auth      system_schema  system_views
system    system_distributed    system_traces  system_virtual_schema

cqlsh> SELECT * FROM system.schema_keyspaces;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table schema_keyspaces does not exist"
cqlsh> use Students;
cqlsh:students> create table Students_Info(Roll_No int Primary key,StudName text,DateOfJoining timestamp,last_exam_Percent double);
cqlsh:students> describe tables;

students_info

cqlsh:students> describe table students;
Table 'students' not found in keyspace 'students'
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 mentable = '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 mentable_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,'Sadhana','2023-10-09', 98) insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent) values(2,'Rutu','2023-10-10', 97) insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent) values(3,'Rachana','2023-10-10', 97.5) insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent) values(4,'Charu','2023-10-06', 96.5) apply batch;
cqlsh:students> select * from students_info;

roll_no | dateofjoining | last_exam_percent | studname
-----+-----+-----+-----
1 | 2023-10-06 18:30:00.000000+0000 | 98 | Sadhana
2 | 2023-10-09 18:30:00.000000+0000 | 97 | Rutu
4 | 2023-10-05 18:30:00.000000+0000 | 96.5 | Charu
3 | 2023-10-09 18:30:00.000000+0000 | 97.5 | Rachana

(4 rows)
cqlsh:students> select * from students_info where roll_no in (1,2,3);

roll_no | dateofjoining | last_exam_percent | studname
-----+-----+-----+-----
1 | 2023-10-06 18:30:00.000000+0000 | 98 | Sadhana
2 | 2023-10-09 18:30:00.000000+0000 | 97 | Rutu
3 | 2023-10-09 18:30:00.000000+0000 | 97.5 | Rachana

(3 rows)
cqlsh:students> select * from students_info where Studname='Charu';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
cqlsh:students> create index on Students_info(StudName);
cqlsh:students> select * from students_info where Studname='Charu';

roll_no | dateofjoining | last_exam_percent | studname
-----+-----+-----+-----
4 | 2023-10-05 18:30:00.000000+0000 | 96.5 | Charu

(1 rows)
cqlsh:students> select Roll_no,StudName from students_info LIMIT 2;

```

```

(4 rows)
cqlsh:students> select * from students_info where roll_no in (1,2,3);

roll_no | dateofjoining | last_exam_percent | studname
-----+-----+-----+-----
1 | 2023-10-06 18:30:00.000000+0000 | 98 | Sadhana
2 | 2023-10-09 18:30:00.000000+0000 | 97 | Rutu
3 | 2023-10-09 18:30:00.000000+0000 | 97.5 | Rachana

(3 rows)
cqlsh:students> select * from students_info where Studname='Charu';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
cqlsh:students> create index on Students_info(StudName);
cqlsh:students> select * from students_info where Studname='Charu';

roll_no | dateofjoining | last_exam_percent | studname
-----+-----+-----+-----
4 | 2023-10-05 18:30:00.000000+0000 | 96.5 | Charu

(1 rows)
cqlsh:students> select Roll_no,StudName from students_info LIMIT 2;

roll_no | studname
-----+-----
1 | Sadhana
2 | Rutu

(2 rows)
cqlsh:students> SELECT Roll_no as "USN" from Students_info;

USN
---
1
2
4
3

```

## Experiment - 4

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

### Code & Output:

```
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 -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$ 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$ 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
```

## Experiment - 5

Q) Implement Wordcount program on Hadoop framework

Code:

### Mapper Code: WCMapper.java

```
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: WCReducer.java

```
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> {

    public void reduce(Text key, Iterator<IntWritable> value, OutputCollector<Text,
IntWritable> output, Reporter rep) throws IOException {

        int count = 0;

        while (value.hasNext()) {

            IntWritable i = value.next();

            count += i.get();

        }

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

    }
}

```

### **Driver Code: WCDriver.java**

```

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;
    }
}

```

```

public static void main(String args[]) throws Exception {
    int exitCode = ToolRunner.run(new WCDriver(), args);
    System.out.println(exitCode);
}
}

```

## Output:

```

hadoop@bmscece-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -copyFromLocal -f /home/hadoop/Desktop/file1.txt /rgs/test.txt
hadoop@bmscece-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@bmscece-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@bmscece-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@bmscece-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

```

## Experiment – 6

Q) From the following link extract the weather data

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

Create a Map Reduce program to

- a) find average temperature for each year from NCDC data set.
- b) find the mean max temperature for every month.

### Code:

- a) Find average temperature for each year from NCDC data set

### AverageDriver.java

```
package temp;

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

public class AverageDriver {

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

        if (args.length != 2) {

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

            System.exit(-1);

        }

        Job job = new Job();

        job.setJarByClass(AverageDriver.class);

        job.setJobName("Max temperature");
```

```

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

job.setMapperClass(AverageMapper.class);
job.setReducerClass(AverageReducer.class);

job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

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

```

### **AverageMapper.java**

```

package temp;

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

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

    public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context)
        throws IOException, InterruptedException {

```

```

int temperature;

String line = value.toString();
String year = line.substring(15, 19);

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

String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches("[01459]"))
    context.write(new Text(year), new IntWritable(temperature));
}
}

```

### **AverageReducer.java**

```

package temp;

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

public class AverageReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable> values,
        Reducer<Text, IntWritable, Text, IntWritable>.Context context)
        throws IOException, InterruptedException {

```

```

int max_temp = 0;

int count = 0;

for (IntWritable value : values) {

    max_temp += value.get();

    count++;

}

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

}

}

```

## Output:

```

C:\hadoop-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.DefaultNoHARMFaloverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-15 14:52:51,805 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-000000

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /avgtemp_outputdir/part-r-000000
1901    46

C:\hadoop-3.3.0\sbin>

```

**b) Find the mean max temperature for every month**

**MeanMaxDriver.java**

```
package meanmax;

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

public class MeanMaxDriver {

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Please Enter the input and output parameters");
            System.exit(-1);
        }

        Job job = new Job();
        job.setJarByClass(MeanMaxDriver.class);
        job.setJobName("Max temperature");

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

        job.setMapperClass(MeanMaxMapper.class);
        job.setReducerClass(MeanMaxReducer.class);
```

```

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

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

```

## **MeanMaxMapper.java**

java

CopyEdit

```
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));
}
}

```

### **MeanMaxReducer.java**

java

CopyEdit

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));
}
}

```

**Output:**

```

C:\hadoop-3.3.0\sbin>hadoop jar C:\meanmax.jar meanmax.MeanMaxDriver /input_dir/temp.txt /meanmax_output
2021-05-21 20:28:05,250 INFO client.DefaultHadoopFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-21 20:28:06,662 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-21 20:28:06,916 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621608943095_0001
2021-05-21 20:28:08,425 INFO input.FileInputFormat: Total input files to process : 1
2021-05-21 20:28:09,107 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621608943095_0001
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-21 20:28:10,029 INFO conf.Configuration: resource-types.xml not found
2021-05-21 20:28:10,030 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-21 20:28:10,676 INFO impl.VarClientImpl: Submitted application application_1621608943095_0001
2021-05-21 20:28:11,005 INFO mapreduce.Job: The url to track the job: http://LAPTOP-36329ESD:8088/proxy/application_1621608943095_0001/
2021-05-21 20:28:11,006 INFO mapreduce.Job: Running job: job_1621608943095_0001
2021-05-21 20:28:29,385 INFO mapreduce.Job: Job job_1621608943095_0001 running in uber mode : false
2021-05-21 20:28:29,389 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-21 20:28:40,664 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-21 20:28:50,832 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-21 20:28:58,965 INFO mapreduce.Job: Job job_1621608943095_0001 completed successfully
2021-05-21 20:28:59,178 INFO mapreduce.Job: Counters: 54

File System Counters:
  FILE: Number of bytes read=50882
  FILE: Number of bytes written=648091
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=894860
  HDFS: Number of bytes written=74
  HDFS: Number of read operations=0
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0

Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=8077
  Total time spent by all reduces in occupied slots (ms)=7511
  Total time spent by all map tasks (ms)=8077
  Total time spent by all reduce tasks (ms)=7511
  Total vcore-milliseconds taken by all map tasks=8077
  Total vcore-milliseconds taken by all reduce tasks=7511
  Total megabyte-milliseconds taken by all map tasks=8270848
  Total megabyte-milliseconds taken by all reduce tasks=7691264

```

```

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

C:\hadoop-3.3.0\sbin>

```

## Experiment – 7

Q) 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.

Code :

### **TopN.java (Driver)**

java

CopyEdit

```
package samples.topn;
```

```
import java.io.IOException;
```

```
import java.util.StringTokenizer;
```

```
import org.apache.hadoop.conf.Configuration;
```

```
import org.apache.hadoop.fs.Path;
```

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

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

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

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

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

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

```
import org.apache.hadoop.util.GenericOptionsParser;
```

```
public class TopN {
```

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

```
        Configuration conf = new Configuration();
```

```
        String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
```

```

if (otherArgs.length != 2) {
    System.err.println("Usage: TopN <in> <out>");
    System.exit(2);
}

```

```

Job job = Job.getInstance(conf);
job.setJobName("Top N");
job.setJarByClass(TopN.class);
job.setMapperClass(TopNMapper.class);
job.setReducerClass(TopNReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

```

```

FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));

```

```

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

```

```

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

```

```

    public void map(Object key, Text value, Mapper<Object, Text, Text,
IntWritable>.Context context)
        throws IOException, InterruptedException {
        String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");

```

```
StringTokenizer itr = new StringTokenizer(cleanLine);
```

```
while (itr.hasMoreTokens()) {  
    this.word.set(itr.nextToken().trim());  
    context.write(this.word, one);  
}  
}  
}  
}
```

### **TopNCombiner.java**

```
java
```

```
CopyEdit
```

```
package samples.topn;
```

```
import java.io.IOException;
```

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

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

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

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

```
    public void reduce(Text key, Iterable<IntWritable> values,
```

```
        Reducer<Text, IntWritable, Text, IntWritable>.Context context)
```

```
        throws IOException, InterruptedException {
```

```
        int sum = 0;
```

```
        for (IntWritable val : values)
```

```
            sum += val.get();
```

```
        context.write(key, new IntWritable(sum));
```

```
    }
```

```
}
```

## **TopNMapper.java**

```
java
```

```
CopyEdit
```

```
package samples.topn;
```

```
import java.io.IOException;
```

```
import java.util.StringTokenizer;
```

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

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

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

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

```
    private static final IntWritable one = new IntWritable(1);
```

```
    private Text word = new Text();
```

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

```
    public void map(Object key, Text value, Mapper<Object, Text, Text,  
IntWritable>.Context context)
```

```
        throws IOException, InterruptedException {
```

```
        String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
```

```
        StringTokenizer itr = new StringTokenizer(cleanLine);
```

```
        while (itr.hasMoreTokens()) {
```

```
            this.word.set(itr.nextToken().trim());
```

```
            context.write(this.word, one);
```

```
        }
```

```
    }
```

```
}
```

## **TopNReducer.java**

```
java
```

```
CopyEdit
```

```
package samples.topn;
```

```
import java.io.IOException;
```

```
import java.util.HashMap;
```

```
import java.util.Map;
```

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

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

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

```
import utils.MiscUtils;
```

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

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

```
    public void reduce(Text key, Iterable<IntWritable> values,
```

```
        Reducer<Text, IntWritable, Text, IntWritable>.Context context)
```

```
        throws IOException, InterruptedException {
```

```
        int sum = 0;
```

```
        for (IntWritable val : values)
```

```
            sum += val.get();
```

```
        this.countMap.put(new Text(key), new IntWritable(sum));
```

```
    }
```

```
    protected void cleanup(Reducer<Text, IntWritable, Text, IntWritable>.Context context)
```

```
        throws IOException, InterruptedException {
```



```

Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(this.countMap);
int counter = 0;
for (Text key : sortedMap.keySet()) {
    if (counter++ == 20)
        break;
    context.write(key, sortedMap.get(key));
}
}
}

```

## Output:

```

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.DefaultHadoopFailoverProxyProvider: 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-JG329ES0:8088/proxy/application_1620483374279_0001/
2021-05-08 19:54:57,508 INFO mapreduce.Job: Running job: job_1620483374279_0001
2021-05-08 19:55:13,792 INFO mapreduce.Job: Job job_1620483374279_0001 running in uber mode : false
2021-05-08 19:55:13,794 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-08 19:55:20,020 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-08 19:55:27,116 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-08 19:55:33,199 INFO mapreduce.Job: Job job_1620483374279_0001 completed successfully
2021-05-08 19:55:33,334 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=65
    FILE: Number of bytes written=530397
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=142
    HDFS: Number of bytes written=31
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0

```

```

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

C:\hadoop-3.3.0\sbin>

```

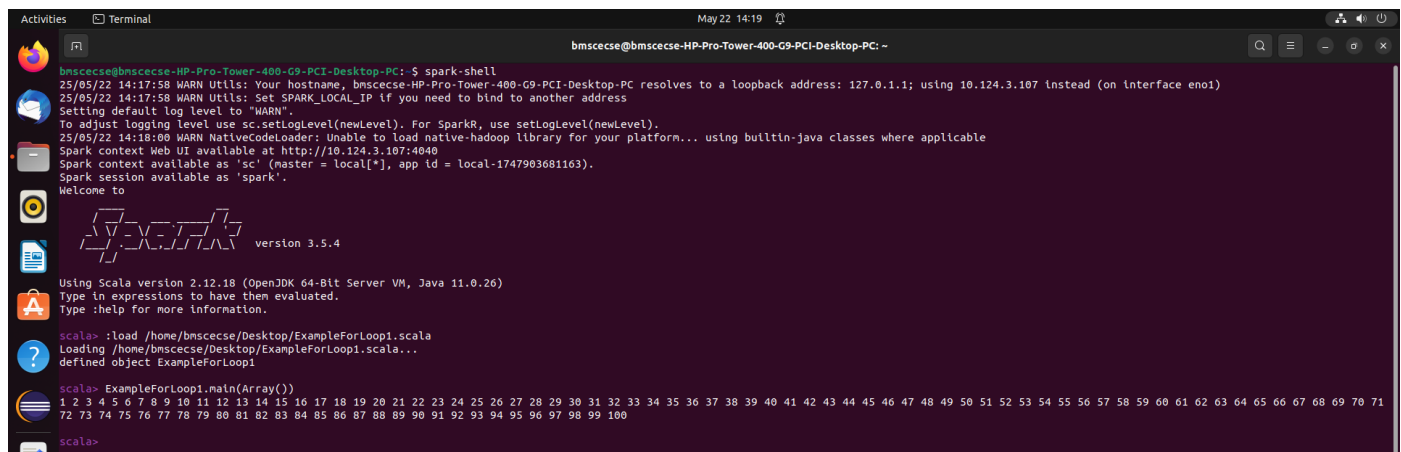
## Experiment – 8

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

**Code :**

```
object ExampleForLoop1 {  
  
    def main(args: Array[String]): Unit = {  
  
        for (counter <- 1 to 100)  
  
            print(counter + " ")  
  
        // to print new line  
  
        println()  
  
    }  
  
}
```

**Output:**



```
Activities Terminal May 22, 14:19 bmscscse@bmscscse-HP-Pro-Tower-400-G9-PCI-Desktop-PC: ~  
bmscscse@bmscscse-HP-Pro-Tower-400-G9-PCI-Desktop-PC: $ spark-shell  
25/05/22 14:17:58 WARN Utils: Your hostname, bmscscse-HP-Pro-Tower-400-G9-PCI-Desktop-PC resolves to a loopback address: 127.0.1.1; using 10.124.3.107 instead (on interface eno1)  
25/05/22 14:17:58 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
25/05/22 14:18:00 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
Spark context Web UI available at http://10.124.3.107:4040  
Spark context available as 'sc' (master = local[*], app id = local-1747903601163).  
Spark session available as 'spark'.  
Welcome to  
version 3.5.4  
Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.26)  
Type in expressions to have them evaluated.  
Type :help for more information.  
scala> :load /home/bmscscse/Desktop/ExampleForLoop1.scala  
Loading /home/bmscscse/Desktop/ExampleForLoop1.scala...  
defined object ExampleForLoop1  
scala> ExampleForLoop1.main(Array())  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71  
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100  
scala>
```

## Experiment-9

Q) 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

**Code :**

```
import org.apache.spark.sql.SparkSession

object FilterWordCount {

  def main(args: Array[String]): Unit = {

    if (args.length < 1) {

      System.err.println("Usage: FilterWordCount <file>")

      System.exit(1)

    }

    val spark = SparkSession.builder()

      .appName("FilterWordCount")

      .master("local[*]") // Use local[*] for running on local machine with multiple cores

      .getOrCreate()

    val rdd = spark.sparkContext.textFile(args(0))

    val counts = rdd

      .flatMap(_.split("\\s+"))

      .map(_.replaceAll("[p{Punct}]", ""))

      .filter(_.nonEmpty)

      .map(w => (w.toLowerCase, 1))

      .reduceByKey(_ + _)

      .filter(_._2 > 4)

    counts.collect().foreach{ case (w, c) => println(s"$w -> $c") }

    spark.stop()

  }

}
```

## Output

```
bmscsec@bmscsec-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ spark-shell
25/05/22 14:19:33 WARN Utils: Your hostname, bmscsec-HP-Pro-Tower-400-G9-PCI-Desktop-PC resolves to a loopback address: 127.0.1.1; using 10.124.3.187 instead (on interface eno1)
25/05/22 14:19:33 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/05/22 14:19:35 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
25/05/22 14:19:36 WARN Utils: Service 'SparkUI' could not bind to port 4040. Attempting port 4041.
Spark context Web UI available at http://10.124.3.187:4041/
Spark context available as 'sc' (master = local[*], app id = local-1747903776341).
Spark session available as 'spark'.
Welcome to

      ____              __
     / ___/____  _____  ____
    / _  /_  / ____/ ___/ _  /
   /___/  /_/____/_/    /___/ version 3.5.4

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.26)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :load /home/bmscsec/Desktop/FilterWordCount.scala
Loading /home/bmscsec/Desktop/FilterWordCount.scala...
import org.apache.spark.sql.SparkSession
defined object FilterWordCount

scala> FilterWordCount.main(Array("/home/bmscsec/Desktop/sample.txt"))
25/05/22 14:21:05 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
mainNav -> 5

scala>
```

## Experiment-10

Q) 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).

### Code:

```
import org.apache.spark.streaming.{Seconds, StreamingContext}

import org.apache.spark.ml.feature.{RegexTokenizer, StopWordsRemover}

import org.apache.spark.sql.functions._

object TextStreamCleaner {

  def main(args: Array[String]): Unit = {

    val ssc = new StreamingContext(sc, Seconds(5))

    val sparkSession = spark

    import sparkSession.implicits._

    val lines = ssc.socketTextStream("localhost", 9999)\

    lines.foreachRDD { rdd =>

      if (!rdd.isEmpty()) {

        val df = rdd.toDF("text")

        val tokenizer = new RegexTokenizer()

          .setInputCol("text")

          .setOutputCol("words")

          .setPattern("\\W")

        val tokenizedDF = tokenizer.transform(df)

        val remover = new StopWordsRemover()

          .setInputCol("words")

          .setOutputCol("filtered")

        val cleanedDF = remover.transform(tokenizedDF)

        cleanedDF.select("filtered").show(false)
```

```

    }
}

ssc.start()

ssc.awaitTermination()

}

}

```

## Output

```

bmsccece@bmsccece-HP-Pro-Tower-400-G9-PCI-Desktop-PC: ~$ nc -lk 9999
hi my name is vagisha

bmsccece@bmsccece-HP-Pro-Tower-400-G9-PCI-Desktop-PC: ~$ spark-shell
25/05/22 15:13:53 WARN Utils: Your hostname, bmsccece-HP-Pro-Tower-400-G9-PCI-Desktop-PC resolves to a loopback address: 127.0.1.1; using 10.124.3.107 instead (on interface eno1)
25/05/22 15:13:53 WARN Utils: set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/05/22 15:13:55 WARN NativeCodeLoader: Unable to load native hadoop library for your platform... using builtin-java classes where applicable
25/05/22 15:13:55 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://10.124.3.107:4041
Spark context available as 'sc' (master = local[*], app id = local-1747907035836).
Spark session available as 'spark'.
Welcome to

  ____  __
 / ___/ /  __
/ /   / /  /
/ /___/ /  /
/_____/___/

version 3.5.4

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.20)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :load /home/bmsccece/Desktop/TextStreamCleaner.scala
Loading /home/bmsccece/Desktop/TextStreamCleaner.scala...
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark.ml.feature.{RegexTokenizer, StopWordsRemover}
import org.apache.spark.sql.functions._
warning: one deprecation (since Spark 3.4.0); for details, enable '-setting -deprecation' or '-replay -deprecation'
defined object TextStreamCleaner

scala> TextStreamCleaner.main(Array())
25/05/22 15:16:39 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
25/05/22 15:16:39 WARN BlockManager: Block input-0-1747907199400 replicated to only 0 peer(s) instead of 1 peers
-----
|filtered|
-----+-----
|[M, name, vagisha]|
-----

[Stage 0:>] (0 + 1) / 1]

```