

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Big Data Analytics (23CS6PCBDA)

Submitted by:

Nidhi A (1BM22CS177)

**Under the Guidance of
Vikranth B.M.
Assistant Professor, BMSCE**

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

March 2024 - June 2024

**B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "**Big Data Analytics**" carried out by **Nidhi A (1BM22CS177)**, 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 2025. The Lab report has been approved as it satisfies the academic requirements in respect of **Big Data Analytics - (23CS6PCBDA)** work prescribed for the said degree.

Vikranth B.M.
Associate Professor
Department of CSE
BMSCE, Bengaluru

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

Table Of Contents

		<ul style="list-style-type: none"> • Create a TTL of 15 seconds to display the values of Employees. 	
--	--	--	--

		2.3.2	Code with Output	
	2.4	Experiment - 4		10
		2.4.1	Question: Hadoop Installation Screenshot	
		2.4.2	Screenshot	
	2.5	Experiment - 5		12
		2.5.1	Question: Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)	
		2.5.2	Code with Output	
	2.6	Experiment - 6		17
		2.6.1	Question: Implement WordCount Program on Hadoop framework.	
		2.6.2	Code with Output	
	2.7	Experiment - 7		21
		2.7.1	Question: From the following link extract the weather data: https://github.com/tomwhite/hadoop-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.	
		2.7.2	Code with Output	
	2.8	Experiment - 8		24
		2.8.1	Question: 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.	
		2.8.2	Code with Output	
	2.9	Experiment - 9		
		2.9.1	Question: Write a Scala program to print numbers from 1 to 100 using for loop	

	2.10	Experiment - 10	
		Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.	
	2.10.2	Code with Output	

1. Course Outcomes

CO1: Apply the concepts of NoSQL, Hadoop, Spark for a given task

CO2: Analyse data analytic techniques for a given problem .

CO3: Conduct experiments using data analytics mechanisms for a given problem.

2. Experiments

2.1 Experiment - 1

2.1.1 Question:

MongoDB - CRUD Demonstration.

2.1.2 Code with 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": "Alicee" } }  
)
```

```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.updateOne(  
...   { "Rollno": 11 },  
...   { $set: { "Name": "Alicee" } }  
... )  
{  
  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: 'Alicee', 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
}

```

Perform the following DB operations using Cassandra.

- Create a keyspace by name Employee
- Create a column family by name, Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name
- Insert the values into the table in batch
- Update Employee name and Department of Emp-Id 121
- Sort the details of Employee records based on salary
- Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
- Update the altered table to add project names.
- Create a TTL of 15 seconds to display the values of Employees.

2.1.3 Code with Output:

```

bmsccse@bmsccse-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 Employee with replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
syntaxException: Line 1:89 mismatched input ';' expecting ')' (...with replication = {'class': 'SimpleStrategy', 'replication_factor': 1}...)
cqlsh> create keyspace Employee WITH replication={'class': 'SimpleStrategy', 'replication_factor':1};
ConfigurationException: Unrecognized strategy option (replication_factor) passed to SimpleStrategy for keyspace employee
cqlsh> create keyspace Employee WITH replication={'class': 'SimpleStrategy', 'replication_factor':1};
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 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

```

```
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 = '999';
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;
```

2.2 Experiment - 2

2.2.1 Question:

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.

2.2.2 Code with Output:

```
bmseccse@bmseccse-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 memtable = 'default'  
    AND crc_check_chance = 1.0  
    AND default_time_to_live = 0  
    AND extensions = {}  
    AND gc_grace_seconds = 8640000  
    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,'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 | 2023-10-08 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 | 2023-10-08 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 | 2023-10-08 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 |
+-----+
```

2.3 Experiment - 3

2.3.1 Question:

Perform the following DB operations using Cassandra.

- Create a keyspace by name Employee
- Create a column family by name, Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name
- Insert the values into the table in batch
- Update Employee name and Department of Emp-Id 121
- Sort the details of Employee records based on salary
- Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
- Update the altered table to add project names.
- Create a TTL of 15 seconds to display the values of Employees.

2.3.2 Code with Output:

```
bmscse@bmscse-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 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';
```

2.4 Experiment - 4

2.4.1 Question:

Hadoop Installation Screenshot

2.4.2 Screenshot:

The screenshot shows the Hadoop Web UI interface. At the top left is the Hadoop logo. To its right, the title "All Applications" is displayed. On the left side, there is a sidebar with a tree view under the "Cluster" section, showing "About", "Nodes", "Node Labels", and "Applications". Under "Applications", several states are listed: NEW, NEW_SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, and KILLED. Below the sidebar is a "Scheduler Metrics" section with a table header for "Scheduler Type" and "Scheduling Resource Type". The table shows one entry for "Capacity Scheduler" with values "[memory:mb (unit=Mi), vcores]". The main area displays a table titled "Cluster Metrics" with columns for "Apps Submitted", "Apps Pending", "Apps Running", "Apps Completed", "Containers Running", "Used Resources", and "Total Resources". The table shows 0 entries in all columns except "Containers Running" which is 0. Below this is another table titled "Cluster Nodes Metrics" with columns for "Active Nodes", "Decommissioning Nodes", "Decommissioned Nodes", "Lost Nodes", and "Unhealt". The table shows 1 active node, 0 decommissioning nodes, 0 decommissioned nodes, 0 lost nodes, and 0 unhealthy nodes. A message "No data available in table" is visible at the bottom of the table. At the very bottom, a status bar says "Showing 0 to 0 of 0 entries".

2.5 Experiment - 5

2.5.1 Question:

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

2.5.2 Code 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 -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
```

2.5.3 Question:

Implement WordCount Program on Hadoop framework.

2.5.4 Code with Output:

Mapper Code:

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

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

Reducer Code:

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

} }

Driver Code: WCDriver Java Class file.

```
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);
    }
}
```

2.6 Experiment - 6

2.6.1 Question:

From the following link extract the weather data:

<https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all>

Create a Map Reduce program to:

- c) Find average temperature for each year from NCDC data set.
- d) Find the mean max temperature for every month.

2.6.2 Code with Output:

a) Find average temperature for each year from NCDC data

set. AverageDriver:

```
package temp;
import org.apache.hadoop.fs.Path; import
org.apache.hadoop.io.IntWritable; import
org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class AverageDriver {
    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Please Enter the input and output parameters");
            System.exit(-1);
        }
        Job job = new Job();
        job.setJarByClass(AverageDriver.class);
        job.setJobName("Max temperature");
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        job.setMapperClass(AverageMapper.class);
        job.setReducerClass(AverageReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

AverageMapper:

```
package temp;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class AverageMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    public static final int MISSING = 9999;
    public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
    IntWritable>.Context context) throws IOException, InterruptedException {
        int temperature;
        String line = value.toString(); String
        year = line.substring(15, 19); if
        (line.charAt(87) == '+') {
```

```

temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(year), new IntWritable(temperature));
}
}

```

AverageReducer:

```

package temp;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class AverageReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int count = 0;
for (IntWritable value : values) {
max_temp += value.get();
count++;
}
context.write(key, new IntWritable(max_temp / count));
}}

```

```

:~\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.DefaultNWARNFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-15 14:52:51,005 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-15 14:52:51,111 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621060230696_0005
2021-05-15 14:52:51,735 INFO input.FileInputFormat: Total input files to process : 1
2021-05-15 14:52:52,751 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621060230696_0005
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-15 14:52:53,237 INFO conf.Configuration: resource-types.xml not found
2021-05-15 14:52:53,238 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-15 14:52:53,312 INFO impl.YarnClientImpl: Submitted application application_1621060230696_0005
2021-05-15 14:52:53,352 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329E5D:8088/proxy/application_1621060230696_0005/
2021-05-15 14:52:53,352 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 10% reduce 0%
2021-05-15 14:53:19,868 INFO mapreduce.Job: map 10% reduce 100%
2021-05-15 14:53:25,967 INFO mapreduce.Job: Job job_1621060230696_0005 completed successfully
2021-05-15 14:53:26,996 INFO mapreduce.Job: Counters: 54
    File System Counters
        FILE: Number of bytes read=72210
        FILE: Number of bytes written=674341
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=894860
        HDFS: Number of bytes written=8
        HDFS: Number of read operations=8
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
        HDFS: Number of bytes read erasure-coded=0
    Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=3782

```

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

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

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

2.7 Experiment - 7

- a) find the mean max temperature for every month **MeanMaxDriver.class**

```
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.class

```
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.class

```

package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class MeanMaxReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int max_temp = 0;
int total_temp = 0;
int count = 0;
int days = 0;
for (IntWritable value : values) {
int temp = value.get();
if (temp > max_temp)
max_temp = temp;
count++;
if (count == 3) {
total_temp += max_temp;
max_temp = 0;
count = 0;
days++;
}
}
context.write(key, new IntWritable(total_temp / days));
}
}

```

```

C:\hadoop-3.3.0\sbin>hadoop jar C:\meanmax.jar MeanMaxDriver /input_dir/temp.txt /meanmax_output
2021-05-21 20:28:05,258 INFO client.DefaultNoHARNFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8088
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_1621688943095_0001
2021-05-21 20:28:08,425 INFO input.FileInputFormat: Total input files to process :1
2021-05-21 20:28:09,167 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621688943095_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.ResourcesUtils: Unable to find 'resource-types.xml'
2021-05-21 20:28:10,676 INFO impl.YarnClientImpl: Submitted application application_1621688943095_0001
2021-05-21 20:28:11,005 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621688943095_0001/
2021-05-21 20:28:11,006 INFO mapreduce.Job: Running job: job_1621688943095_0001
2021-05-21 20:28:29,385 INFO mapreduce.Job: Job job_1621688943095_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_1621688943095_0001 completed successfully
2021-05-21 20:28:59,178 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=59882
FILE: Number of bytes written=648891
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=834860
HDFS: Number of bytes written=74
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)=8877
Total time spent by all reduces in occupied slots (ms)=7511
Total time spent by all map tasks (ms)=8877
Total time spent by all reduce tasks (ms)=7511
Total wcore-milliseconds taken by all map tasks=8877
Total wcore-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>

```

2.8 Experiment - 8

2.8.1 Question:

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.

2.8.2 Code with Output:

Driver-TopN.class package

TopNCombiner.class

```
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.class

```
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.class

```
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));
}
}
}
}

```

```

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>hdfs dfs -cat /output_dir/*
hello    2
hadoop   1
world    1
bye      1

C:\hadoop-3.3.0\sbin>

```

```
C:\hadoop-3.3.0\bin>hadoop jar C:\sort.jar samples.topN /input_dir/input.txt /output_dir
2021-05-08 19:54:54,582 INFO client.DefaultHttpAWMFailoverProxyProvider: 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 19:54:56,582 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-JG329E5D: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 19:55:13,793 INFO mapreduce.Job: map 0% reduce 0%
2021-05-08 19:55:20,828 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
```

2.9 Experiment - 9

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

Code with output:

```
object ExampleForLoop2 {  
    def main(args: Array[String]) {  
        var counter: Int=0;  
  
        for(counter <- 1 until 101)  
            print(counter + " ");  
  
        // to print new line  
        println();  
    }  
}
```

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

2.10 Experiment - 10

Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.

```
// Step 1: Load the text file into an RDD  
val lines = sc.textFile("input.txt")  
  
// Step 2: Split each line into words using flatMap  
val words = lines.flatMap(line => line.split("\\s+"))  
  
// Step 3: Map each word to a (word, 1) pair  
val wordPairs = words.map(word => (word, 1))  
  
// Step 4: Reduce by key to count each word  
val wordCounts = wordPairs.reduceByKey((a, b) => a + b)  
  
// Step 5: Filter words with count > 4  
val filteredWords = wordCounts.filter{ case (word, count) => count > 4 }  
  
// Step 6: Collect and print the results  
filteredWords.collect().foreach { case (word, count) => println(s"$word: $count") }
```

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC: ~
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC: ~
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC: ~
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC: ~

res26: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[28] at map at
<console>:26

scala> .reduceByKey(_ + _)
res27: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[29] at reduceByKey
at <console>:26

scala> .filter(_.value > 4)
res28: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[30] at filter
at <console>:26

scala> result.collect().foreach(println)
Spark is fast. Spark is powerful. Spark is amazing. Spark is fast. Spark rocks.
Fast spark processing.

scala> res22: org.apache.spark.rdd.RDD[(String, Int)]
res30: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[23] at filter
at <console>:26

scala> res22.collect().foreach(println)
(spark,6)

scala> []
```