VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

Big Data Analytics

Submitted by

Malingaray p jakati(1BM22CS143)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Feb-2024 to July-2024

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "LAB COURSE **Big Data Analytics**" carried out by **Malingaray p jakati (1BM22CS143),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics - (23CS6PCBDA)** work prescribed for the said degree.

Leelavathi B

Assistant Professor
Department of CSE
BMSCE, Bengaluru

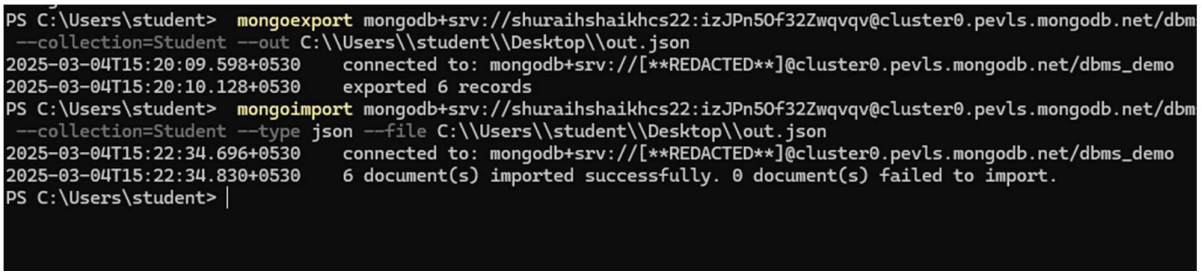
Dr. Kavitha Sooda

Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

github link: https://github.com/malingaraypj/BDA

Lab 1 MongoDB Part - 1



I. **CREATE DATABASE IN MONGODB. use myDB;**

Confirm the existence of your database

db;

To list all databases

show dbs;

II. **CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS**

1. To create a collection by the name “Student”. Let us take a look at the collection list prior to the creation of the new collection “Student”.

   db.createCollection(“Student”);

2. To drop a collection by the name “Student”.

   db.Student.drop();

3. Create a collection by the name “Students” and store the following data in it.

   db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing"});

4. Insert the document for “AryanDavid” in to the Students collection only if it does not already exist in the collection.

   db.Student.update({_id:3,StudName:"AryanDavid",Grade:"VII"},{$set:{Hobbies:"Skating"}},{upsert:true});

5. FIND METHOD

   A. To search for documents from the “Students” collection based on certain search criteria.
   db.Student.find({StudName:"Aryan David"});

   B. To display only the StudName and Grade from all the documents of the Students collection. The identifier_id should be suppressed and NOT displayed. **db.Student.find({},{StudName:1,Grade:1,_id:0});**

   C. To find those documents where the Grade is set to ‘VII’ **db.Student.find({Grade:{$eq:'VII'}}).pretty();**

   D. To find those documents from the Students collection where the Hobbies is set to either ‘Chess’ or is set to ‘Skating’. **db.Student.find({Hobbies :{ $in: ['Chess','Skating']}}).pretty ();**

E. To find documents from the Students collection where the StudName begins with "M".
db.Student.find({StudName:/^M/}).pretty();

F. To find documents from the Students collection where the StudNamehas an "e" in any position.
db.Student.find({StudName:/e/}).pretty();

G. To find the number of documents in the Students collection.
db.Student.count();

H. To sort the documents from the Students collection in the descending order of StudName.
db.Student.find().sort({StudName:-1}).pretty();

III. Import data from a CSV file

Given a CSV file "sample.txt" in the D:drive, import the file into the MongoDB collection, "SampleJSON". The collection is in the database "test".

mongoimport --db Student --collection airlines --type csv –headerline --file /home/hduser/Desktop/airline.csv

IV. Export data to a CSV file

This command used at the command prompt exports MongoDB JSON documents from "Customers" collection in the "test" database into a CSV file "Output.txt" in the D:drive.

mongoexport --host localhost --db Student --collection airlines --csv --out /home/hduser/Desktop/output.txt –fields "Year","Quarter"

V. Save Method :

Save() method will insert a new document, if the document with the _id does not exist. If it exists it will replace the exisiting document:

db.Students.save({StudName:"Vamsi", Grade:"VI"})

VI. Add a new field to existing Document:

db.Students.update({_id:4},{$set:{Location:"Network"}})

VII. Remove the field in an existing Document

db.Students.update({_id:4},{$unset:{Location:"Network"}})

VIII. Finding Document based on search criteria suppressing few fields

db.Student.find({_id:1},{StudName:1,Grade:1,_id:0});

To find those documents where the Grade is not set to 'VII'

db.Student.find({Grade:{$ne:'VII'}}).pretty();

To find documents from the Students collection where the StudName ends with s.

db.Student.find({StudName:/s$/}).pretty();

IX. to set a particular field value to NULL db.Students.update({_id:3},{$set:{Location:null}})

X. **Count the number of documents in Student Collections** db.Students.count()

XI. **Count the number of documents in Student Collections with grade :VII**
db.Students.count({Grade:"VII"})

retrieve first 3 documents
db.Students.find({Grade:"VII"}).limit(3).pretty();

Sort the document in Ascending order
db.Students.find().sort({StudName:1}).pretty();

to Skip the 1st two documents from the Students Collections

db.Students.find().skip(2).pretty()

XII. Create a collection by name "food" and add to each document add a "fruits" array db.food.insert({ _id:1, fruits:['grapes','mango','apple'] }) db.food.insert({ _id:2, fruits:['grapes','mango','cherry'] }) db.food.insert({ _id:3, fruits:['banana','mango'] })

To find those documents from the "food" collection which has the "fruits array" constitute of "grapes", "mango" and "apple". db.food.find ({fruits: ['grapes','mango','apple'] }). pretty().
To find in "fruits" array having "mango" in the first index position.
db.food.find ({'fruits.1':'grapes'})

To find those documents from the "food" collection where the size of the array is two.
db.food.find ({"fruits": {$size:2}})

To find the document with a particular id and display the first two elements from the array "fruits"
db.food.find({_id:1},{"fruits":{$slice:2}})

To find all the documets from the food collection which have elements mango and grapes in the array "fruits"
db.food.find({fruits:{$all:["mango","grapes"]}})

update on Array:

using particular id replace the element present in the 1st index position of the fruits array with apple
db.food.update({_id:3},{$set:{'fruits.1':'apple'}})

insert new key value pairs in the fruits array
db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}})

XII. Aggregate Function :

Create a collection Customers with fields custID, AcctBal, AcctType.
Now group on "custID" and compute the sum of "AccBal". db.Customers.aggregate (
{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } });

match on AcctType:"S" then group on "CustID" and compute the sum of "AccBal". db.Customers.aggregate (
{$match:{AcctType:"S"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } });

match on AcctType:"S" then group on "CustID" and compute the sum of "AccBal" and total balance greater than 1200.
db.Customers.aggregate ({$match:{AcctType:"S"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } },
{$match:{TotAccBal:{$gt:1200}}});

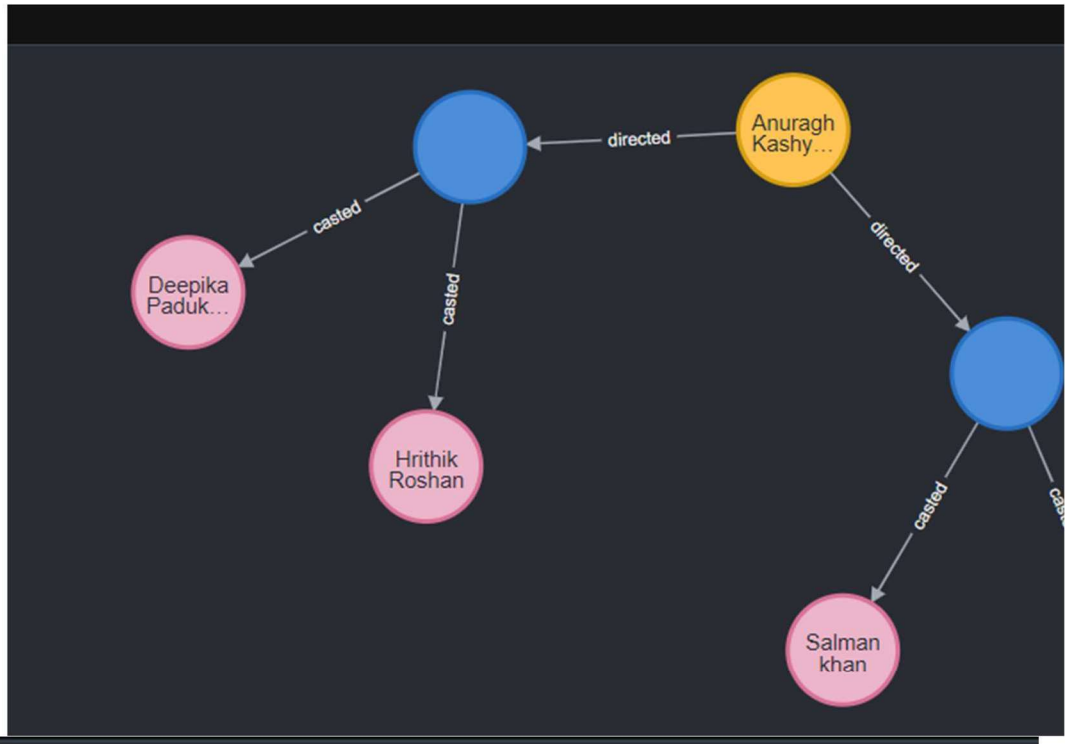
Lab 2 MongoDB Part - 2

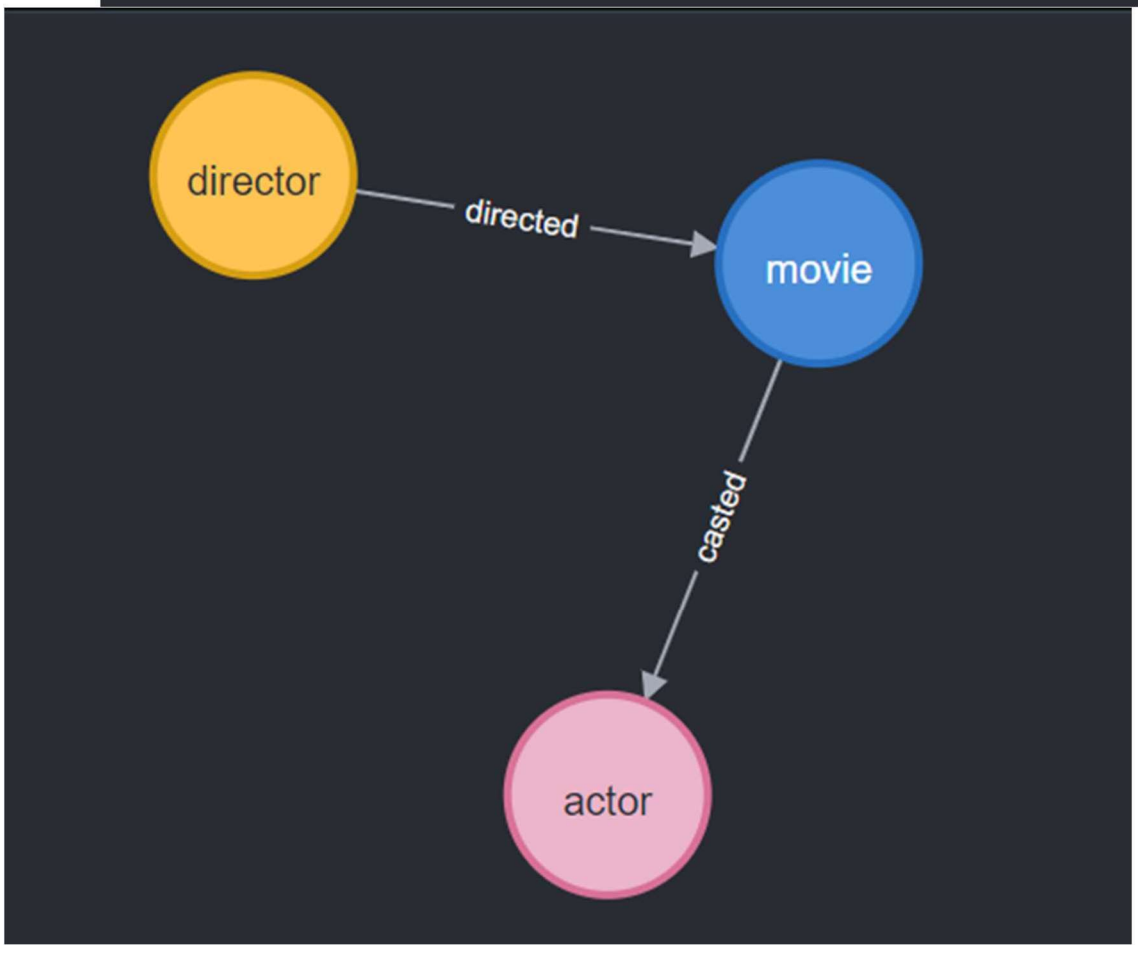
```
test> db.food.find({"fruits": {$size:2}});
[ { _id: 3, fruits: [ 'banana', 'mango' ] } ]
test> db.food.find({_id:1},{"fruits":{$slice:2}});
[ { _id: 1, fruits: [ 'grapes', 'mango' ] } ]
test> db.food.update({_id:3}, {$set: {'fruits.1':'apple'}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.food.update({_id:2}, {$push: {price:{grapes:80,mango:200,cherry:100}}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.food.update({_id:3}, {$set: {'fruits.1':'apple'}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Customers.aggregate([{ $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }
... ]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } },
...    { $match:{TotAccBal:{$gt:1200}}}
... ]);

test> db.Alphabets.insertMany([{_id:1, alphabet:"A"}, {_id:2, alphabet:"B"}, {_id:3, alphabet:"C"}]);
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3 } }
test> var myCursor = db.Alphabets.find();
```

```
TypeError: db.Students.save is not a function
test> db.Students.update({_id:4}, {$set:{Location:"Network"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Students.update({_id:4}, {$unset:{Location:"Network"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Students.update({_id:4}, {$unset:{Location:"Network"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
```

```
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.food.update({_id:3}, {$set: {'fruits.1':'apple'}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Customers.aggregate([{ $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }
... ]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } },
...    { $match:{TotAccBal:{$gt:1200}}}
... ]);

test> db.Alphabets.insertMany([{_id:1, alphabet:"A"}, {_id:2, alphabet:"B"}, {_id:3, alphabet:"C"}]);
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3 } }
test> var myCursor = db.Alphabets.find();

test> while (myCursor.hasNext()) {
...     printjson(myCursor.next());
... }
{
  _id: 1,
  alphabet: 'A'
}
{
  _id: 2,
  alphabet: 'B'
}
{
  _id: 3,
  alphabet: 'C'
}

test> show dbs;
admin    40.00 KiB
config  108.00 KiB
local   128.00 KiB
mydb     40.00 KiB
shdb    112.00 KiB
test     96.00 KiB
test>
```

Lab 3 Neo4J


directed
Anuragh Kashy…
casted
Deepika Paduk…
casted
directed
Hrithik Roshan
casted
Salman khan


director
directed
movie
casted
actor

Lab 4 Cassandra Part - I

1. What is the command used to create a keyspace named Employee with SimpleStrategy and replication factor 1?

```
CREATE KEYSPACE Employee

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

How do you create a table named Employee_Info with fields for ID, name, designation, joining date, salary, and department?

```
CREATE TABLE Employee_Info (

   Emp_Id int PRIMARY KEY,

   Emp_Name text,

   Designation text,

   Date_of_Joining date,

   Salary float,

   Dept_Name text

);
```

2. How do you insert multiple records in a batch in Cassandra?

```
BEGIN BATCH

INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)

VALUES (121, 'Anit', 'Manager', '2018-02-01', 70000.0, 'Sales');

INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)

VALUES (122, 'Priya', 'Developer', '2020-06-15', 50000.0, 'IT');
```

INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)

VALUES (123, 'Rahul', 'Analyst', '2019-11-20', 60000.0, 'Finance');
APPLY BATCH;

3. What query updates the name and department of the employee with Emp_Id = 121?

UPDATE Employee_Info

SET Emp_Name = 'Anit Kumar', Dept_Name = 'Marketing'

WHERE Emp_Id = 121;

4. What is the correct query to fetch employees whose salary is greater than 0 using ALLOW FILTERING?

SELECT * FROM Employee_Info

WHERE Salary > 0

ALLOW FILTERING;

5. How do you add a new column Projects of type set<text> to the table?

ALTER TABLE Employee_Info ADD Projects set<text>;

6. How do you update the projects of employee with Emp_Id = 121?

UPDATE Employee_Info

SET Projects = {'ProjectA', 'ProjectB'}

WHERE Emp_Id = 121;

7. How do you insert a new record into the updated table including the new Projects column with TTL?

INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)

VALUES (124, 'Neha', 'HR', '2022-03-01', 45000.0, 'HR')

USING TTL 15;

```
cqlsh> CREATE KEYSPACE Employee
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE Employee_Info (
            ...     Emp_Id int PRIMARY KEY,
            ...     Emp_Name text,
            ...     Designation text,
            ...     Date_of_Joining date,
            ...     Salary float,
            ...     Dept_Name text
            ... );
cqlsh:employee> BEGIN BATCH
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (121, 'Amit', 'Manager', '2018-02-01', 70000.0, 'Sales');
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (122, 'Priya', 'Developer', '2020-06-15', 50000.0, 'IT');
            ... INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (123, 'Rahul', 'Analyst', '2019-11-20', 60000.0, 'Finance');
            ... APPLY BATCH;
cqlsh:employee> UPDATE Employee_Info
            ... SET Emp_Name = 'Amit Kumar', Dept_Name = 'Marketing'
            ... WHERE Emp_Id = 121;
cqlsh:employee>
cqlsh:employee> SELECT * FROM Employee_Info
            ... WHERE Salary IS NOT NULL
            ... ALLOW FILTERING;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Unsupported restriction: salary IS NOT NULL"
cqlsh:employee> SELECT * FROM Employee_Info
            ... WHERE Salary > 0
            ... ALLOW FILTERING;

 emp_id | date_of_joining | dept_name | designation | emp_name   | salary
--------+-----------------+-----------+-------------+------------+--------
    123 |      2019-11-20 |   Finance |     Analyst |      Rahul |  60000
    122 |      2020-06-15 |        IT |   Developer |      Priya |  50000
    121 |      2018-02-01 | Marketing |     Manager | Amit Kumar |  70000

(3 rows)
cqlsh:employee> ALTER TABLE Employee_Info ADD Projects set<text>;
cqlsh:employee> UPDATE Employee_Info
            ... SET Projects = {'ProjectA', 'ProjectB'}
            ... WHERE Emp_Id = 121;
cqlsh:employee> INSERT INTO Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name)
            ... VALUES (124, 'Neha', 'HR', '2022-03-01', 45000.0, 'HR')
            ... USING TTL 15;
cqlsh:employee> SELECT * FROM Employee_Info;

 emp_id | date_of_joining | dept_name | designation | emp_name   | projects                | salary
--------+-----------------+-----------+-------------+------------+-------------------------+--------
    123 |      2019-11-20 |   Finance |     Analyst |      Rahul |                    null |  60000
    122 |      2020-06-15 |        IT |   Developer |      Priya |                    null |  50000
    121 |      2018-02-01 | Marketing |     Manager | Amit Kumar | {'ProjectA', 'ProjectB'} |  70000

(3 rows)
```

Lab 5 Cassandra Part - II

A. Table: library_student_info

B.Table:book_counter_info

C. Insert Data in Batch

You can repeat the `UPDATE` if you want to increment the counter multiple times. To Simulate Borrowing Book “BDA” 2 Times by Student 112

Display Table & Increase Counter

Query: Student 112 took “BDA” 2 times

```
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 TABLE Employee.Employee_Info ( Emp_Id int PRIMARY KEY, Emp_Name text, Designation text, Date_of_Joining date, Salary decimal, Dept_Name text );
AlreadyExists: Table 'employee.employee_info' already exists
cqlsh> use keyspace Employee
   ... ;
Improper use command.
cqlsh> use keyspace Employee;
Improper use command.
cqlsh> use Employee;
cqlsh:employee> CREATE TABLE Employee.Employee_Info ( Emp_Id int PRIMARY KEY, Emp_Name text, Designation text, Date_of_Joining date, Salary decimal, Dept_Name text );
AlreadyExists: Table 'employee.employee_info' already exists
cqlsh:employee> drop columnfamily employee.employee_info
            ... ;
cqlsh:employee> CREATE TABLE Employee.Employee_Info ( Emp_Id int PRIMARY KEY, Emp_Name text, Designation text, Date_of_Joining date, Salary decimal, Dept_Name text );
cqlsh:employee> BEGIN BATCH INSERT INTO Employee.Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) VALUES (101, 'John Doe', 'Manager', '2020-01-01', 75000.00, 'HR'); INSERT
 INTO Employee.Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) VALUES (102, 'Jane Smith', 'Developer', '2019-03-10', 65000.00, 'IT'); INSERT INTO Employee.Employee_Info (
Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) VALUES (121, 'James Brown', 'Developer', '2018-07-15', 68000.00, 'IT'); INSERT INTO Employee.Employee_Info (Emp_Id, Emp_Name, Designation
, Date_of_Joining, Salary, Dept_Name) VALUES (103, 'Emily White', 'Analyst', '2021-09-20', 60000.00, 'FInance'); APPLY BATCH;
cqlsh:employee> UPDATE Employee.Employee_Info SET Emp_Name = 'James Johnson', Dept_Name = 'Research' WHERE Emp_Id = 121;
cqlsh:employee> SELECT * FROM Employee.Employee_Info WHERE Salary > 0 ALLOW FILTERING;

 emp_id | date_of_joining | dept_name | designation | emp_name      | salary
--------+-----------------+-----------+-------------+---------------+----------
    121 |      2018-07-15 |  Research |   Developer | James Johnson | 68000.00
    102 |      2019-03-10 |        IT |   Developer |    Jane Smith | 65000.00
    101 |      2020-01-01 |        HR |     Manager |      John Doe | 75000.00
    103 |      2021-09-20 |   Finance |     Analyst |   Emily White | 60000.00

(4 rows)
cqlsh:employee> ALTER TABLE Employee.Employee_Info ADD Projects set<text>;
cqlsh:employee> UPDATE Employee.Employee_Info SET Projects = {'Project A', 'Project B', 'Project C'} WHERE Emp_Id = 121;
cqlsh:employee> INSERT INTO Employee.Employee_Info (Emp_Id, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name) VALUES (104, 'Michael Green', 'Tester', '2022-05-12', 54000.00, 'QA') USING TTL 15;
cqlsh:employee> exit;
```

Lab 6 Hadoop HDFS

1. mkdir

Command: hdfs dfs -mkdir /abc

Description: Creates a directory /abc in HDFS.

2. ls

Command: hadoop fs -ls /Hadoop

Description: Lists contents of the /Hadoop directory with details like permissions, owner, size, and modification date.

3. put

Command: hdfs dfs -put /home/hduser/Desktop/Welcome.txt /abc/WC.txt

Description: Copies Welcome.txt from the local file system to HDFS path /abc/WC.txt.

To view the file contents in HDFS, use:

Command: hdfs dfs -cat /abc/WC.txt

4. copyFromLocal

Command: hdfs dfs -copyFromLocal /home/hduser/Desktop/Welcome.txt /abc/WC.txt
Description: Similar to put, but only accepts local file paths as source.

To view the copied file's contents:

Command: hdfs dfs -cat /abc/WC2.txt

5. get

Command: hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/WWC.txt

Description: Downloads WC.txt from HDFS to the local path /home/hduser/Downloads/WWC.txt.

To merge multiple HDFS files into one local file:

Command: hdfs dfs -getmerge /abc/WC.txt /abc/WC2.txt /home/hduser/Desktop/Merge.txt

To check ACLs of a directory:

Command: hadoop fs -getfacl /abc/

6. copyToLocal

Command: hdfs dfs -copyToLocal /abc/WC.txt /home/hduser/Desktop
Description: Similar to get, but destination must be a local file path.

7. cat

Command: hdfs dfs -cat /abc/WC.txt

Description: Displays the contents of the file WC.txt in the terminal.

8. mv

Command: hadoop fs -mv /abc /FFF

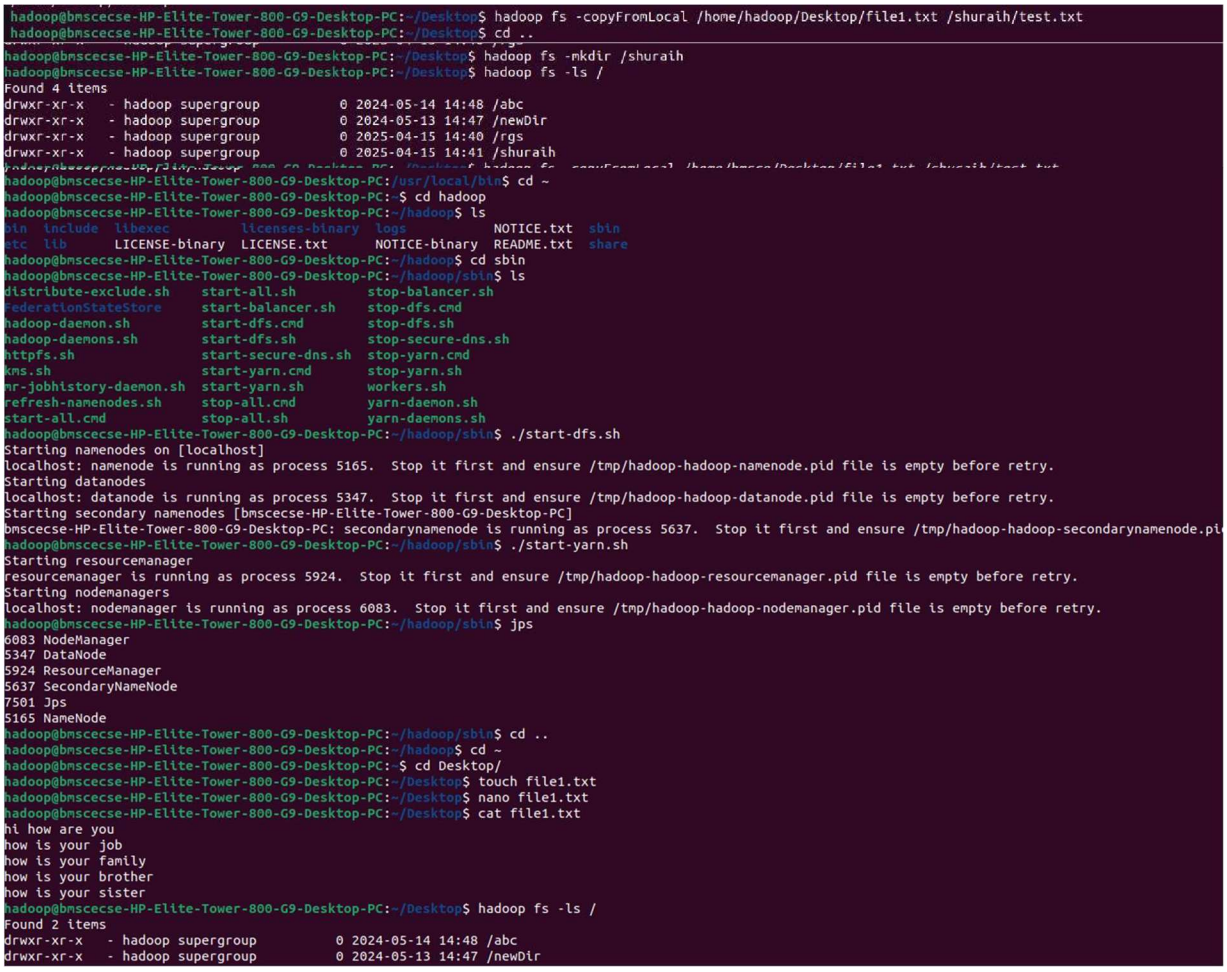
Description: Moves /abc directory in HDFS to /FFF.

9. cp

Command: hadoop fs -cp /CSE/ /LLL

Description: Copies contents from /CSE/ to /LLL within HDFS.

Screenshots



Lab 7 Word Count using Map-Reduce

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

Hadoop services are started using start-all.sh, launching daemons like NameNode, DataNode, and ResourceManager.

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ jps
7042 DataNode
7639 ResourceManager
8248 Jps
6904 NameNode
7305 SecondaryNameNode
7788 NodeManager
4975 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
```

The jps command lists all running Hadoop-related Java processes such as NameNode, DataNode, and ResourceManager.

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop jar WordCount.jar WCDriver /shuraih/test.txt /shuraih/out.txt
2025-04-29 15:11:42,145 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-04-29 15:11:42,185 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-04-29 15:11:42,185 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-04-29 15:11:42,191 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2025-04-29 15:11:42,254 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-04-29 15:11:42,303 INFO mapred.FileInputFormat: Total input files to process : 1
2025-04-29 15:11:42,328 INFO mapreduce.JobSubmitter: number of splits:1
2025-04-29 15:11:42,390 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1048437123_0001
2025-04-29 15:11:42,390 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-29 15:11:42,448 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-04-29 15:11:42,449 INFO mapreduce.Job: Running job: job_local1048437123_0001
2025-04-29 15:11:42,449 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-04-29 15:11:42,451 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2025-04-29 15:11:42,453 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-04-29 15:11:42,453 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-04-29 15:11:42,507 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-04-29 15:11:42,508 INFO mapred.LocalJobRunner: Starting task: attempt_local1048437123_0001_m_000000_0
2025-04-29 15:11:42,519 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-04-29 15:11:42,519 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-04-29 15:11:42,526 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-04-29 15:11:42,529 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/shuraih/test.txt:0+89
2025-04-29 15:11:42,538 INFO mapred.MapTask: numReduceTasks: 1
2025-04-29 15:11:42,569 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-04-29 15:11:42,569 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-04-29 15:11:42,569 INFO mapred.MapTask: soft limit at 83886080
2025-04-29 15:11:42,569 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-04-29 15:11:42,569 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2025-04-29 15:11:42,571 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2025-04-29 15:11:42,600 INFO mapred.LocalJobRunner:
2025-04-29 15:11:42,608 INFO mapred.MapTask: Starting flush of map output
2025-04-29 15:11:42,608 INFO mapred.MapTask: Spilling map output
2025-04-29 15:11:42,608 INFO mapred.MapTask: bufstart = 0; bufend = 169; bufvoid = 104857600
2025-04-29 15:11:42,608 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214320(104857280); length = 77/6553600
2025-04-29 15:11:42,611 INFO mapred.MapTask: Finished spill 0
2025-04-29 15:11:42,616 INFO mapred.Task: Task:attempt_local1048437123_0001_m_000000_0 is done. And is in the process of committing
2025-04-29 15:11:42,618 INFO mapred.LocalJobRunner: hdfs://localhost:9000/shuraih/test.txt:0+89
2025-04-29 15:11:42,618 INFO mapred.Task: Task 'attempt_local1048437123_0001_m_000000_0' done.
2025-04-29 15:11:42,621 INFO mapred.Task: Final Counters for attempt_local1048437123_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=4049
                FILE: Number of bytes written=645512
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
```

A MapReduce job is executed using hadoop jar to process test.txt and generate output in out.txt.

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /shuraih/out.txt
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-04-29 15:11 /shuraih/out.txt/_SUCCESS
-rw-r--r--   1 hadoop supergroup         69 2025-04-29 15:11 /shuraih/out.txt/part-00000
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -cat /shuraih/out.txt/part-00000
are     1
brother 1
family  1
hi      1
how     5
is      4
job     1
sister  1
you     1
your    4
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$
```

The output of the MapReduce job is displayed using hadoop fs –cat.

Lab 8 Mean-Max and Avg Temperature using Map-Reduce

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ jps
5922 NameNode
4503 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
6807 NodeManager
6312 SecondaryNameNode
6058 DataNode
7226 Jps
6653 ResourceManager
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2025-04-29 15:04 op.txt
drwxr-xr-x   - hadoop supergroup          0 2025-04-29 15:11 out.txt
```

All Hadoop daemons (NameNode, DataNode, etc.) are started using start-all.sh on the local machine.

The jps command confirms active Hadoop services such as NameNode, DataNode, and ResourceManager are running.

The hadoop fs -ls command lists the contents of the HDFS root directory, showing two output folders: op.txt and out.txt.

Average Temperature :

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/TemperatureAvg.jar AverageDriver /shuraih/temperature.txt /shuraih/output
2025-05-06 14:50:01,243 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-06 14:50:01,284 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-06 14:50:01,284 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-06 14:50:01,352 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-06 14:50:01,407 INFO input.FileInputFormat: Total input files to process : 1
2025-05-06 14:50:01,468 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-06 14:50:01,542 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local251162856_0001
2025-05-06 14:50:01,542 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-06 14:50:01,608 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-06 14:50:01,609 INFO mapreduce.Job: Running job: job_local251162856_0001
2025-05-06 14:50:01,609 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-06 14:50:01,612 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 14:50:01,613 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 14:50:01,613 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 14:50:01,614 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-06 14:50:01,780 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-06 14:50:01,781 INFO mapred.LocalJobRunner: Starting task: attempt_local251162856_0001_m_000000_0
2025-05-06 14:50:01,804 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 14:50:01,804 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 14:50:01,804 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 14:50:01,811 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-05-06 14:50:01,813 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/shuraih/temperature.txt:0+888190
2025-05-06 14:50:01,849 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-05-06 14:50:01,849 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-06 14:50:01,849 INFO mapred.MapTask: soft limit at 83886080
2025-05-06 14:50:01,849 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-06 14:50:01,849 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2025-05-06 14:50:01,851 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2025-05-06 14:50:01,991 INFO mapred.LocalJobRunner:
2025-05-06 14:50:01,991 INFO mapred.MapTask: Starting flush of map output
2025-05-06 14:50:01,992 INFO mapred.MapTask: Spilling map output
2025-05-06 14:50:01,992 INFO mapred.MapTask: bufstart = 0; bufend = 59076; bufvoid = 104857600
2025-05-06 14:50:01,992 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26188144(104752576); length = 26253/6553600
2025-05-06 14:50:02,000 INFO mapred.MapTask: Finished spill 0
2025-05-06 14:50:02,005 INFO mapred.Task: Task:attempt_local251162856_0001_m_000000_0 is done. And is in the process of committing
2025-05-06 14:50:02,007 INFO mapred.LocalJobRunner: map
2025-05-06 14:50:02,007 INFO mapred.Task: Task 'attempt_local251162856_0001_m_000000_0' done.
2025-05-06 14:50:02,009 INFO mapred.Task: Final Counters for attempt_local251162856_0001_m_000000_0: Counters: 23
        Fle System Counters
                FILE: Number of bytes read=4258
                FILE: Number of bytes written=713814
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=888190
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=1
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=6565
                Map output records=6564
                Map output bytes=59076
                Bytes written=8
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /shuraih
Found 4 items
drwxr-xr-x   - hadoop supergroup          0 2025-04-29 15:11 /shuraih/out.txt
drwxr-xr-x   - hadoop supergroup          0 2025-05-06 14:50 /shuraih/output
-rw-r--r--   1 hadoop supergroup     888190 2025-05-06 14:14 /shuraih/temperature.txt
-rw-r--r--   1 hadoop supergroup         89 2025-04-15 14:43 /shuraih/test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /shuraih/output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-06 14:50 /shuraih/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup          8 2025-05-06 14:50 /shuraih/output/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /shuraih/output/part-r-00000
1901    46
```

Mean Max Temperature:

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/MeanMaxTemp.jar MeanMaxDriver /shuraih/temperature.txt /shuraih/output1
2025-05-06 15:08:42,166 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-06 15:08:42,207 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-06 15:08:42,207 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-06 15:08:42,268 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-06 15:08:42,320 INFO input.FileInputFormat: Total input files to process : 1
2025-05-06 15:08:42,349 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-06 15:08:42,414 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2013736307_0001
2025-05-06 15:08:42,414 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-06 15:08:42,485 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-06 15:08:42,485 INFO mapreduce.Job: Running job: job_local2013736307_0001
2025-05-06 15:08:42,486 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-06 15:08:42,489 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 15:08:42,489 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 15:08:42,489 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 15:08:42,490 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-06 15:08:42,539 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-06 15:08:42,539 INFO mapred.LocalJobRunner: Starting task: attempt_local2013736307_0001_m_000000_0
2025-05-06 15:08:42,550 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2025-05-06 15:08:42,550 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 15:08:42,550 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2025-05-06 15:08:42,557 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-05-06 15:08:42,559 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/shuraih/temperature.txt:0+888190
2025-05-06 15:08:42,595 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-05-06 15:08:42,595 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-06 15:08:42,595 INFO mapred.MapTask: soft limit at 83886080
2025-05-06 15:08:42,595 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-06 15:08:42,595 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2025-05-06 15:08:42,597 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2025-05-06 15:08:42,667 INFO mapred.LocalJobRunner:
2025-05-06 15:08:42,668 INFO mapred.MapTask: Starting flush of map output
2025-05-06 15:08:42,668 INFO mapred.MapTask: Spilling map output
2025-05-06 15:08:42,668 INFO mapred.MapTask: bufstart = 0; bufend = 45948; bufvoid = 104857600
2025-05-06 15:08:42,668 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26188144(104752576); length = 26253/6553600
2025-05-06 15:08:42,678 INFO mapred.MapTask: Finished spill 0
2025-05-06 15:08:42,684 INFO mapred.Task: Task:attempt_local2013736307_0001_m_000000_0 is done. And is in the process of committing
2025-05-06 15:08:42,685 INFO mapred.LocalJobRunner: map
2025-05-06 15:08:42,686 INFO mapred.Task: Task 'attempt_local2013736307_0001_m_000000_0' done.
2025-05-06 15:08:42,688 INFO mapred.Task: Final Counters for attempt_local2013736307_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=4367
                FILE: Number of bytes written=703883
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=888190
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=1
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=6565
                Map output records=6564
                Map output bytes=45948
                Bytes written=74
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /shuraih/output1
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-06 15:08 /shuraih/output1/_SUCCESS
-rw-r--r--   1 hadoop supergroup         74 2025-05-06 15:08 /shuraih/output1/part-r-00000
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /shuraih/output1/part-r-00000
01      4
02      0
03      7
04      44
05      100
06      168
07      219
08      198
09      141
10      100
11      19
12      3
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

Lab 9 Scala and pySpark

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

2. 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.

```
  GNU nano 6.2                                wordCount.py *
from pyspark import SparkContext

sc = SparkContext("local", "SimpleWordCount")


rdd = sc.textFile("text1.txt")

counts = (rdd.flatMap(lambda line: line.split())
             .map(lambda word: (word.lower(), 1))
             .reduceByKey(lambda a, b: a + b)
             .filter(lambda x: x[1] > 4))


for word, count in counts.collect():
    print(word, count)

sc.stop()
```

Spark Shell Execution Screenshots

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ sudo apt update
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:5 https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 InRelease
Ign:1 https://downloads.apache.org/cassandra/debian 40x InRelease
Err:6 https://downloads.apache.org/cassandra/debian 40x Release
  404  Not Found [IP: 88.99.208.237 443]
Hit:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
W: https://repo.mongodb.org/apt/ubuntu/dists/jammy/mongodb-org/6.0/InRelease: Key is stored in legacy trusted
E: The repository 'http://www.apache.org/dist/cassandra/debian 40x Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ sudo apt install python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

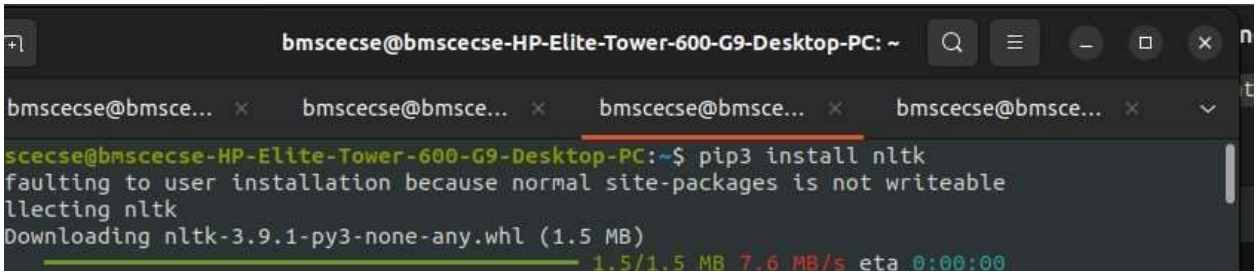
```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ pip3 install pyspark
Defaulting to user installation because normal site-packages is not writeable
Collecting pyspark
  Downloading pyspark-3.5.5.tar.gz (317.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.2/317.2 MB 1.8 MB/s eta 0:00:00
```

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ mkdir ~/pyspark-wordcount
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ cd ~/pyspark-wordcount
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ nano.txt
nano.txt: command not found
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ nano file.txt
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ nano wordcount.py
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ python3 wordcount.py
```

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ python3 wordcount.py
25/05/20 11:41:52 WARN Utils: Your hostname, bmscecse-HP-Elite-Tower-600-G9-Desktop-PC resolves to a loopb
25/05/20 11:41:52 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
25/05/20 11:41:52 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
scala 4
```

3. 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.

```
  GNU nano 6.2                          streaming_cleaner.py *
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re

# Set up Spark context and streaming context
sc = SparkContext("local[2]", "TextCleanerStreaming")
sc.setLogLevel("ERROR")
ssc = StreamingContext(sc, 5)  # 5-second batch interval

# Set of stop words and lemmatizer
stop_words = set(stopwords.words("english"))
lemmatizer = WordNetLemmatizer()

# Connect to TCP socket on localhost:9999
lines = ssc.socketTextStream("localhost", 9999)

def clean_text(line):
    # Lowercase and remove punctuation
    line = re.sub(r"[^a-zA-Z\s]", "", line.lower())
    words = line.split()
    # Remove stopwords and lemmatize
    cleaned = [lemmatizer.lemmatize(word) for word in words if word not in stop_words]
    return " ".join(cleaned)

# Clean each line and print
lines.map(clean_text).pprint()

# Start streaming
ssc.start()
ssc.awaitTermination()
```



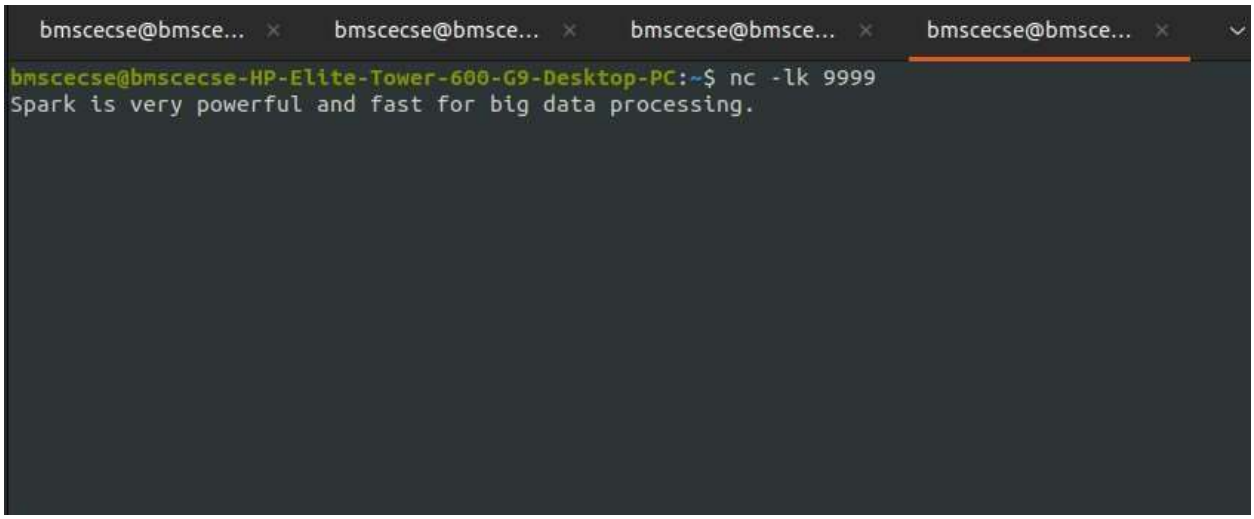
Installation of Natural Language Toolkit (nltk)

nlt

```
04.5
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ python3
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download('stopwords')
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/bmscecse/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
>>> nltk.download('wordnet')
[nltk_data] Downloading package wordnet to /home/bmscecse/nltk_data...
True
>>> exit()
```

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ nano streaming_cleaner.py
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ python3 streaming_cleaner.py
25/05/20 12:05:10 WARN Utils: Your hostname, bmscecse-HP-Elite-Tower-600-G9-Desktop-PC resolv
es to a loopback address: 127.0.1.1; using 10.124.3.71 instead (on interface eno1)
25/05/20 12:05:10 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/
spark-unsafe_2.12-3.0.3.jar) to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platfor
```

Executing the streaming_cleaner.py



Starting a TCP server that listens for incoming connections on port 9999

Output- cleaned data