VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT on

Big Data Analytics

Submitted by

Mallikarjun M Kuri(1BM22CS144)

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

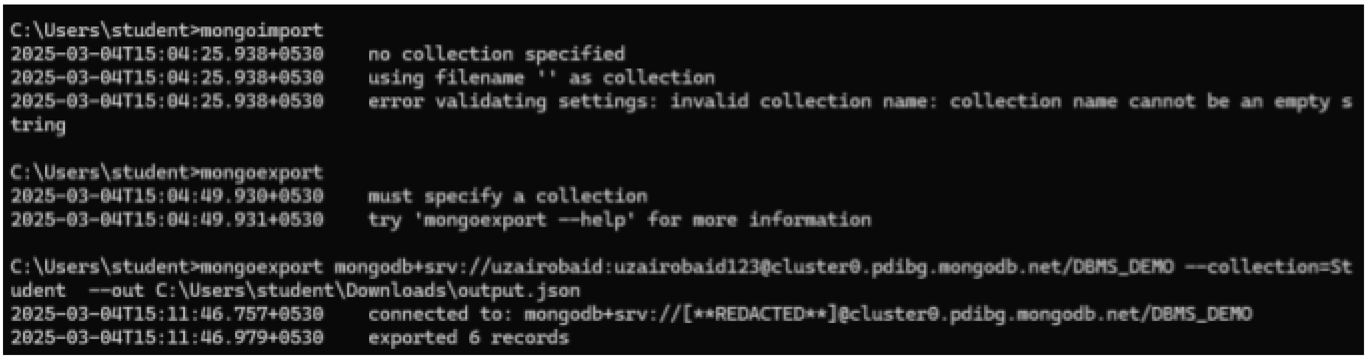
This is to certify that the Lab work entitled “LAB COURSE **Big Data Analytics**” carried out by **Mallikarjun M Kuri(1BM22CS144),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** 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.

Amruta B	**Dr. Kavitha Sooda**
Assistant Professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

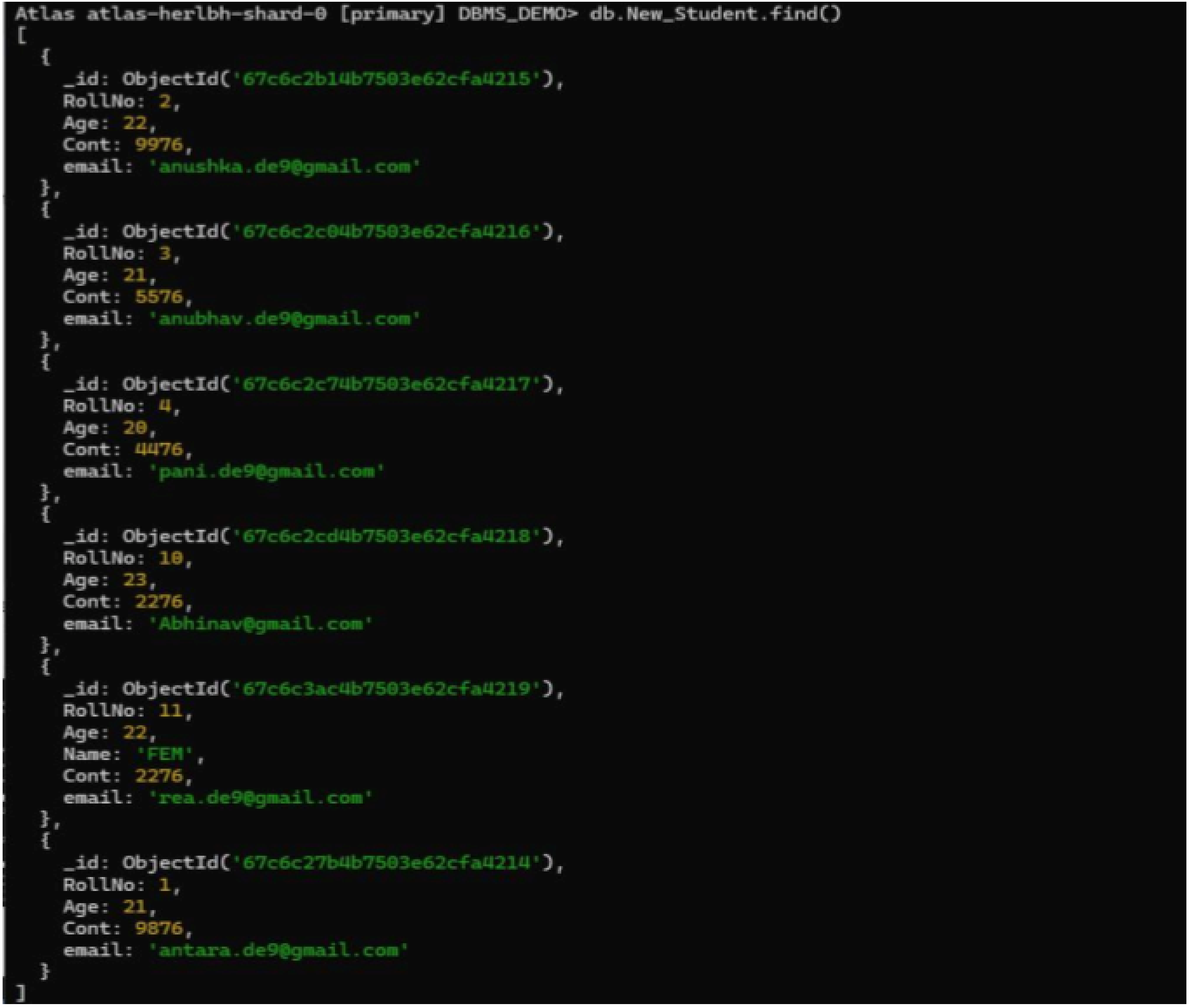
Index Sheet

github link: https://github.com/MalliKarjun008/BDA_Lab

Lab 1 MongoDB Part - 1



```
C:\Users\student>mongoimport
2025-03-04T15:04:25.938+0530    no collection specified
2025-03-04T15:04:25.938+0530    using filename '' as collection
2025-03-04T15:04:25.938+0530    error validating settings: invalid collection name: collection name cannot be an empty s
tring

C:\Users\student>mongoexport
2025-03-04T15:04:49.930+0530    must specify a collection
2025-03-04T15:04:49.931+0530    try 'mongoexport --help' for more information

C:\Users\student>mongoexport mongodb+srv://uzairobaid:uzairobaid123@cluster0.pdibg.mongodb.net/DBMS_DEMO --collection=St
udent  --out C:\Users\student\Downloads\output.json
2025-03-04T15:11:46.757+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.pdibg.mongodb.net/DBMS_DEMO
2025-03-04T15:11:46.979+0530    exported 6 records
```



```
Atlas atlas-herlbh-shard-0 [primary] DBMS_DEMO> db.New_Student.find()
[
  {
    _id: ObjectId('67c6c2b14b7503e62cfa4215'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c2c04b7503e62cfa4216'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c2c74b7503e62cfa4217'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c2cd4b7503e62cfa4218'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  },
  {
    _id: ObjectId('67c6c3ac4b7503e62cfa4219'),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c27b4b7503e62cfa4214'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  }
]
```

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}}})

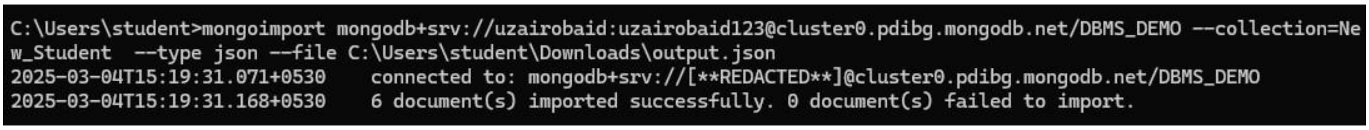
XII. Aggregate Function :

Create a collection Customers with fields custID, AcctBal, AcctType.

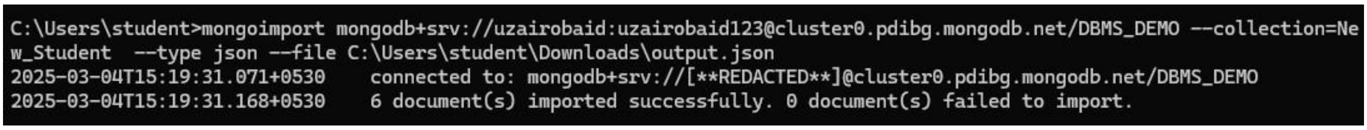
Now group on “custID” and compute the sum of “AccBal”. db.Customers.aggregate ({$group : { _id : “$custID”,TotAccBal : {$sum:”$AccBal”} } });

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}}});

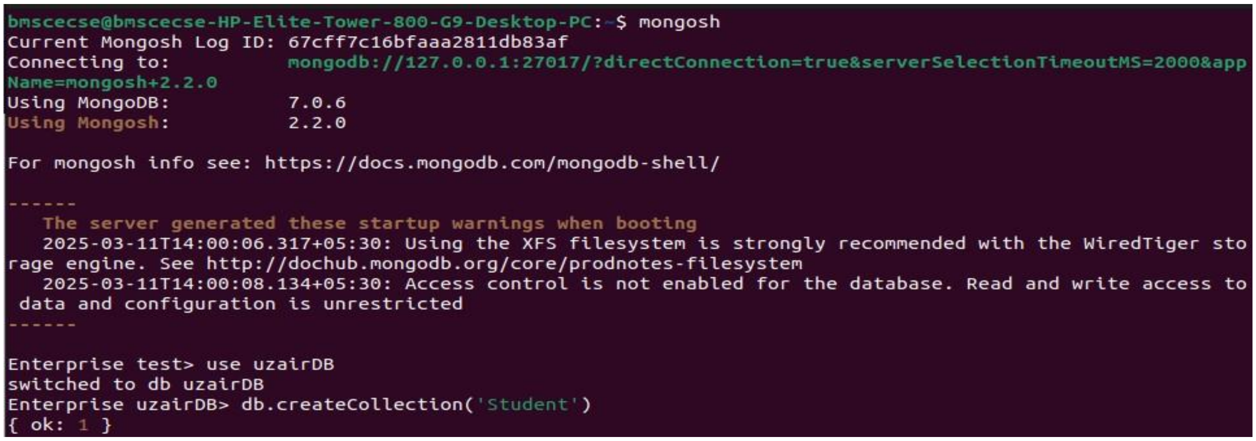


```
C:\Users\student>mongoimport mongodb+srv://uzairobaid:uzairobaid123@cluster0.pdibg.mongodb.net/DBMS_DEMO --collection=Ne
w_Student  --type json --file C:\Users\student\Downloads\output.json
2025-03-04T15:19:31.071+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.pdibg.mongodb.net/DBMS_DEMO
2025-03-04T15:19:31.168+0530    6 document(s) imported successfully. 0 document(s) failed to import.
```

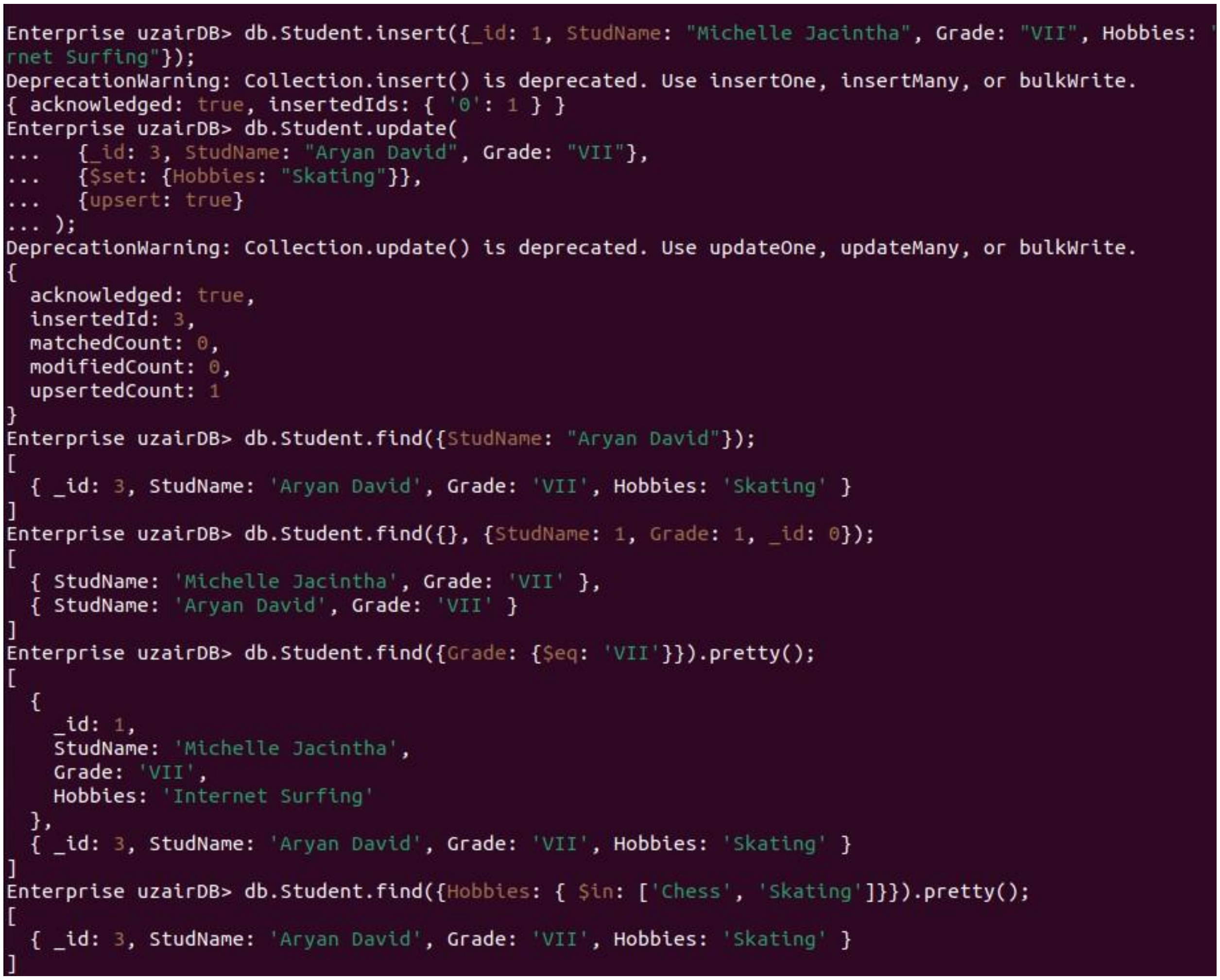


```
C:\Users\student>mongoimport mongodb+srv://uzairobaid:uzairobaid123@cluster0.pdibg.mongodb.net/DBMS_DEMO --collection=Ne
w_Student  --type json --file C:\Users\student\Downloads\output.json
2025-03-04T15:19:31.071+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.pdibg.mongodb.net/DBMS_DEMO
2025-03-04T15:19:31.168+0530    6 document(s) imported successfully. 0 document(s) failed to import.
```

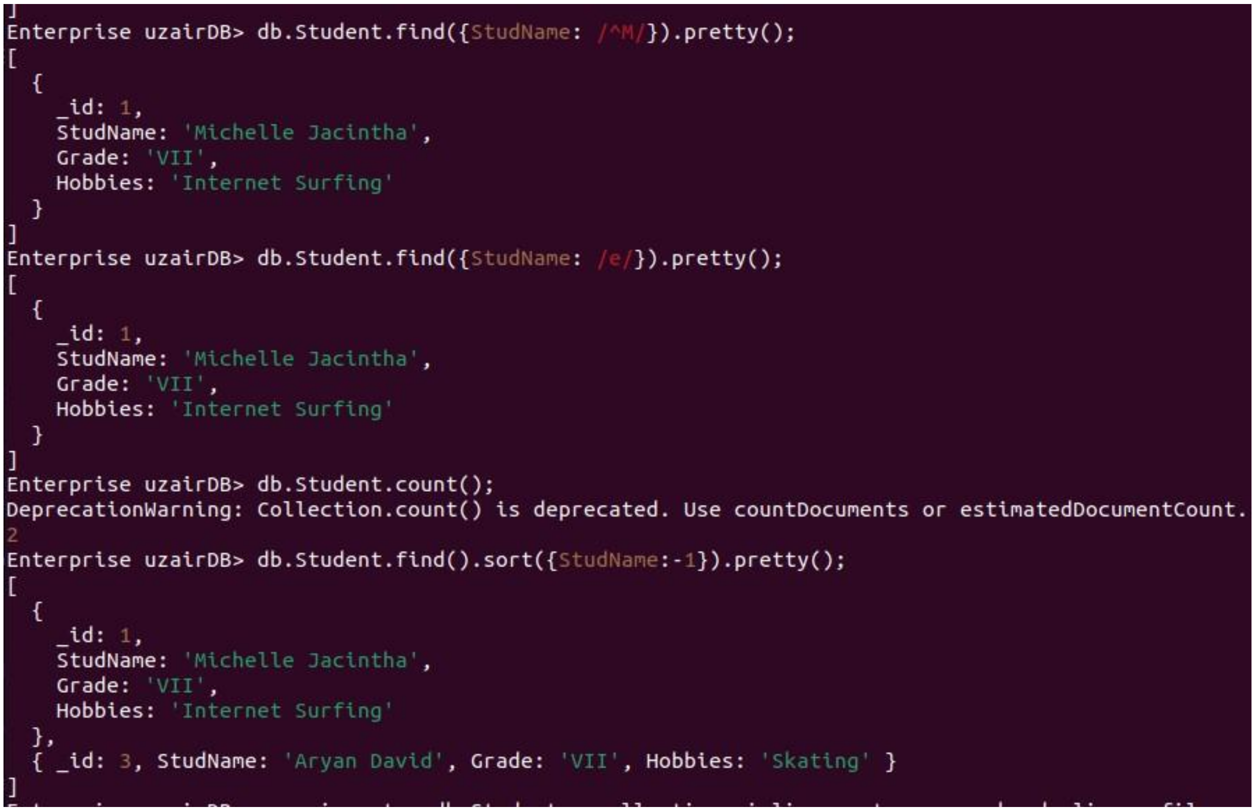
Lab 2 MongoDB Part - 2



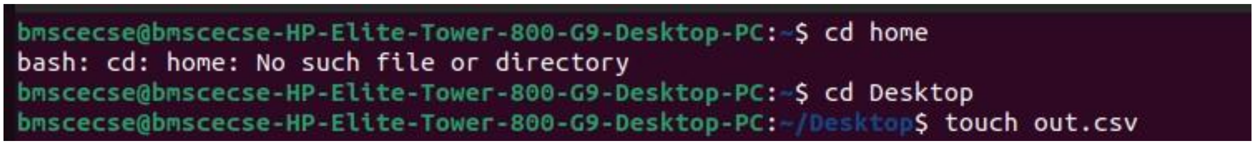
```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongosh
Current Mongosh Log ID: 67cff7c16bfaaa2811db83af
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&app
Name=mongosh+2.2.0
Using MongoDB:          7.0.6
Using Mongosh:          2.2.0

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

------
   The server generated these startup warnings when booting
   2025-03-11T14:00:06.317+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger sto
rage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2025-03-11T14:00:08.134+05:30: Access control is not enabled for the database. Read and write access to
 data and configuration is unrestricted
------

Enterprise test> use uzairDB
switched to db uzairDB
Enterprise uzairDB> db.createCollection('Student')
{ ok: 1 }
```



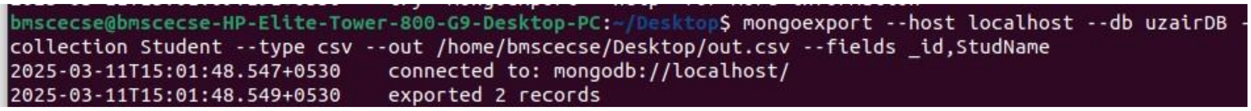
```
Enterprise uzairDB> db.Student.insert({_id: 1, StudName: "Michelle Jacintha", Grade: "VII", Hobbies: "
rnet Surfing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 1 } }
Enterprise uzairDB> db.Student.update(
...   {_id: 3, StudName: "Aryan David", Grade: "VII"},
...   {$set: {Hobbies: "Skating"}},
...   {upsert: true}
... );
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: 3,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
Enterprise uzairDB> db.Student.find({StudName: "Aryan David"});
[
  { _id: 3, StudName: 'Aryan David', Grade: 'VII', Hobbies: 'Skating' }
]
Enterprise uzairDB> db.Student.find({}, {StudName: 1, Grade: 1, _id: 0});
[
  { StudName: 'Michelle Jacintha', Grade: 'VII' },
  { StudName: 'Aryan David', Grade: 'VII' }
]
Enterprise uzairDB> db.Student.find({Grade: {$eq: 'VII'}}).pretty();
[
  {
    _id: 1,
    StudName: 'Michelle Jacintha',
    Grade: 'VII',
    Hobbies: 'Internet Surfing'
  },
  { _id: 3, StudName: 'Aryan David', Grade: 'VII', Hobbies: 'Skating' }
]
Enterprise uzairDB> db.Student.find({Hobbies: { $in: ['Chess', 'Skating']}}).pretty();
[
  { _id: 3, StudName: 'Aryan David', Grade: 'VII', Hobbies: 'Skating' }
]
```



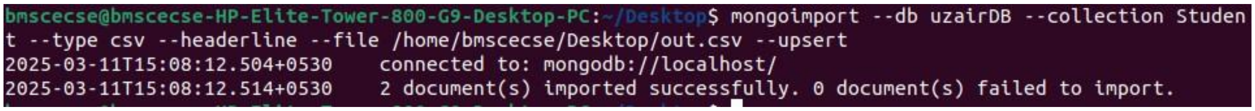
```
]
Enterprise uzairDB> db.Student.find({StudName: /^M/}).pretty();
[
  {
    _id: 1,
    StudName: 'Michelle Jacintha',
    Grade: 'VII',
    Hobbies: 'Internet Surfing'
  }
]
Enterprise uzairDB> db.Student.find({StudName: /e/}).pretty();
[
  {
    _id: 1,
    StudName: 'Michelle Jacintha',
    Grade: 'VII',
    Hobbies: 'Internet Surfing'
  }
]
Enterprise uzairDB> db.Student.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
2
Enterprise uzairDB> db.Student.find().sort({StudName:-1}).pretty();
[
  {
    _id: 1,
    StudName: 'Michelle Jacintha',
    Grade: 'VII',
    Hobbies: 'Internet Surfing'
  },
  { _id: 3, StudName: 'Aryan David', Grade: 'VII', Hobbies: 'Skating' }
]
```



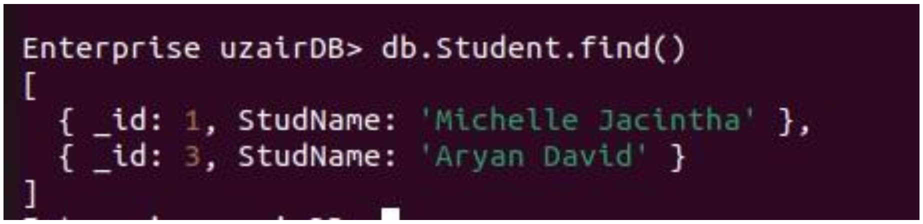
```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd home
bash: cd: home: No such file or directory
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd Desktop
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ touch out.csv
```



```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ mongoexport --host localhost --db uzairDB -
collection Student --type csv --out /home/bmscecse/Desktop/out.csv --fields _id,StudName
2025-03-11T15:01:48.547+0530    connected to: mongodb://localhost/
2025-03-11T15:01:48.549+0530    exported 2 records
```

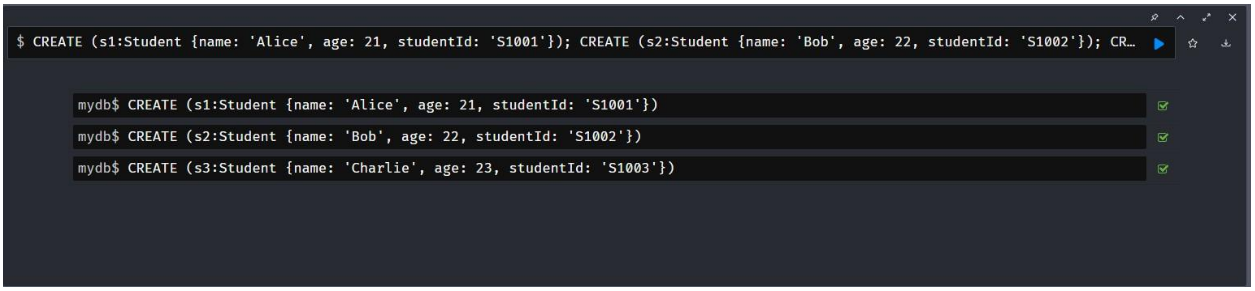


```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ mongoimport --db uzairDB --collection Studen
t --type csv --headerline --file /home/bmscecse/Desktop/out.csv --upsert
2025-03-11T15:08:12.504+0530    connected to: mongodb://localhost/
2025-03-11T15:08:12.514+0530    2 document(s) imported successfully. 0 document(s) failed to import.
```

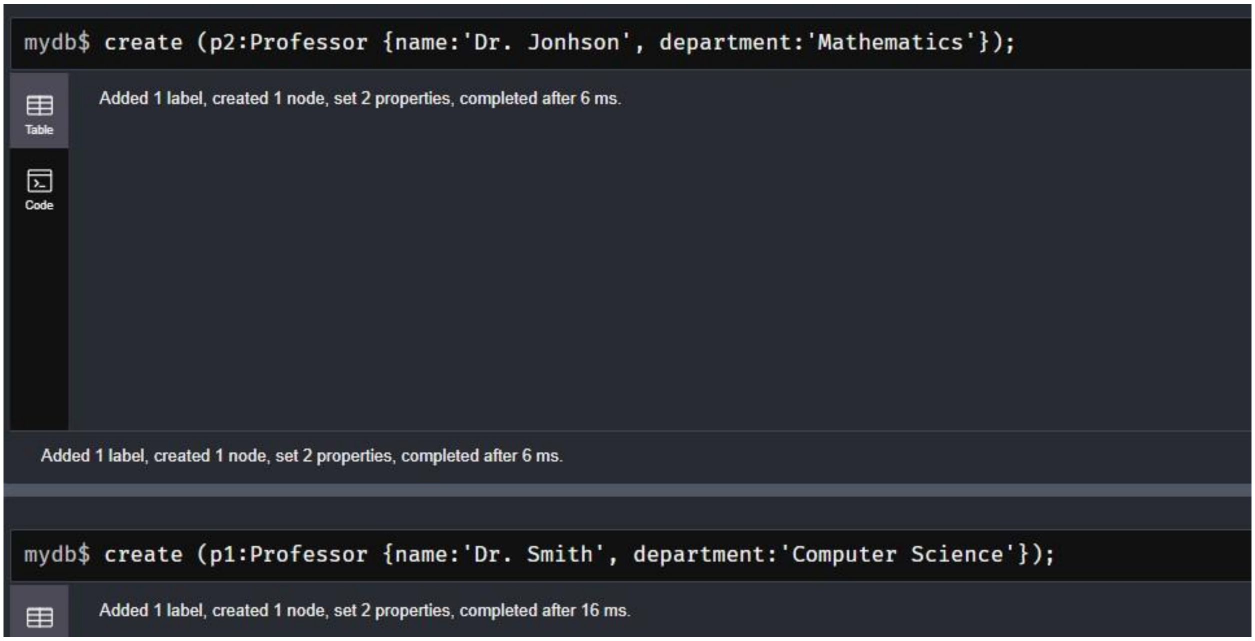


```
Enterprise uzairDB> db.Student.find()
[
  { _id: 1, StudName: 'Michelle Jacintha' },
  { _id: 3, StudName: 'Aryan David' }
]
```

Lab 3 Neo4J



```
$ CREATE (s1:Student {name: 'Alice', age: 21, studentId: 'S1001'}); CREATE (s2:Student {name: 'Bob', age: 22, studentId: 'S1002'}); CR…

mydb$ CREATE (s1:Student {name: 'Alice', age: 21, studentId: 'S1001'})
mydb$ CREATE (s2:Student {name: 'Bob', age: 22, studentId: 'S1002'})
mydb$ CREATE (s3:Student {name: 'Charlie', age: 23, studentId: 'S1003'})
```



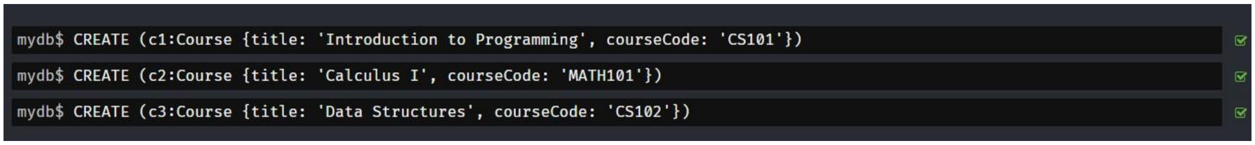
```
mydb$ create (p2:Professor {name:'Dr. Jonhson', department:'Mathematics'});
```

Added 1 label, created 1 node, set 2 properties, completed after 6 ms.

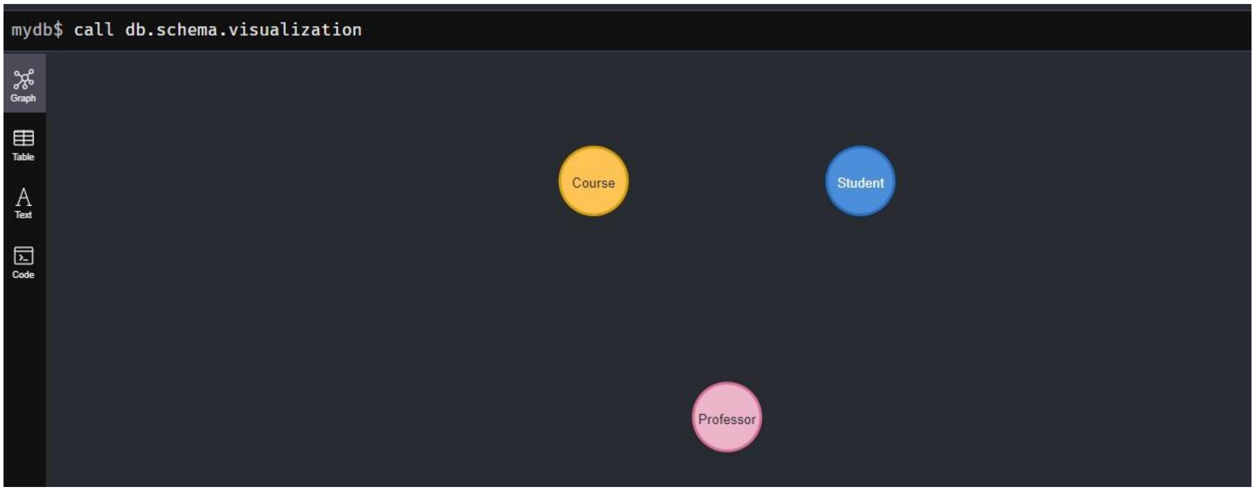
Added 1 label, created 1 node, set 2 properties, completed after 6 ms.

```
mydb$ create (p1:Professor {name:'Dr. Smith', department:'Computer Science'});
```

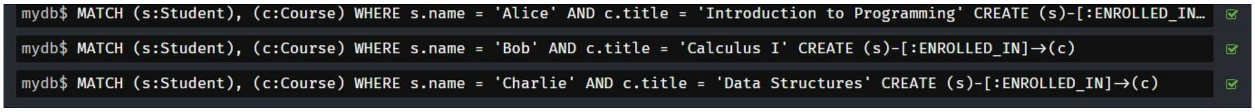
Added 1 label, created 1 node, set 2 properties, completed after 16 ms.



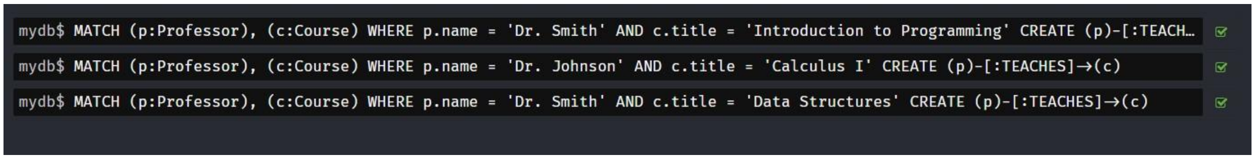
```
mydb$ CREATE (c1:Course {title: 'Introduction to Programming', courseCode: 'CS101'})
mydb$ CREATE (c2:Course {title: 'Calculus I', courseCode: 'MATH101'})
mydb$ CREATE (c3:Course {title: 'Data Structures', courseCode: 'CS102'})
```



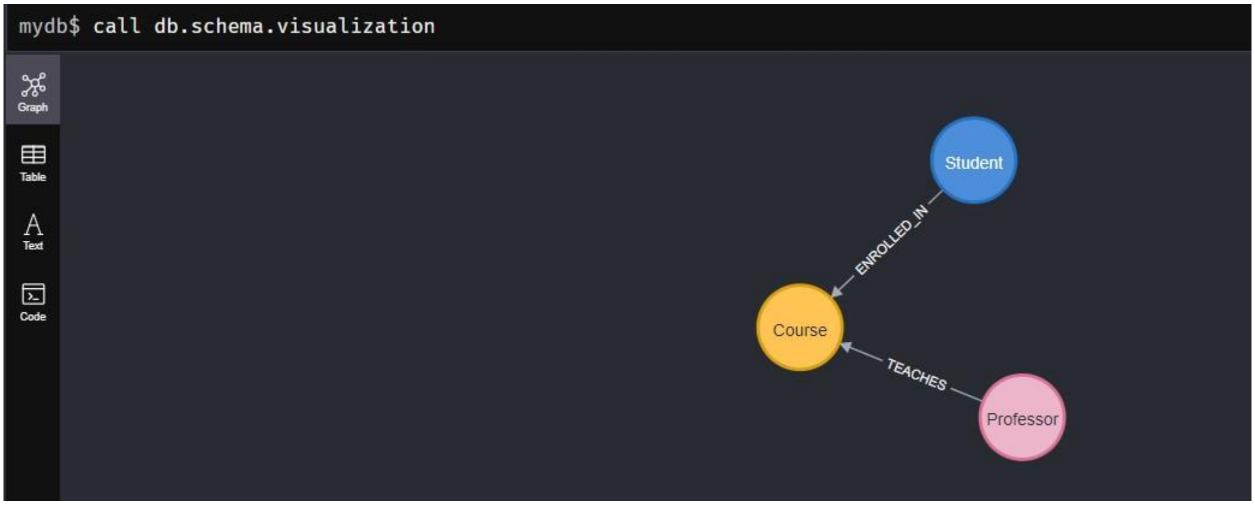
```
mydb$ call db.schema.visualization
```



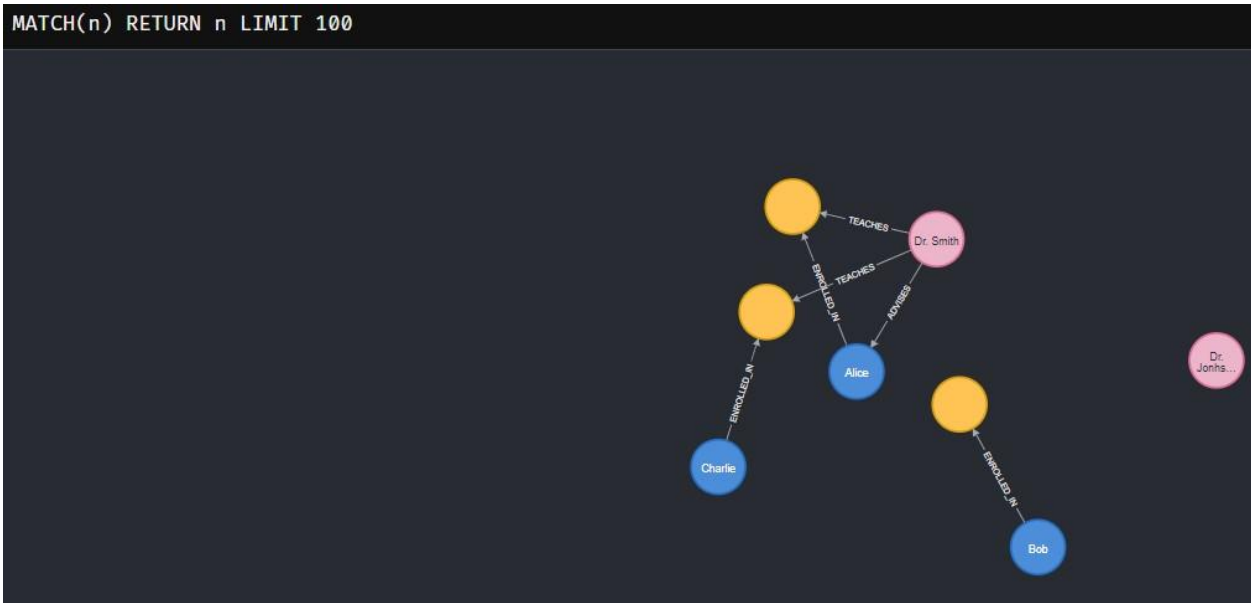
```
mydb$ MATCH (s:Student), (c:Course) WHERE s.name = 'Alice' AND c.title = 'Introduction to Programming' CREATE (s)-[:ENROLLED_IN…
mydb$ MATCH (s:Student), (c:Course) WHERE s.name = 'Bob' AND c.title = 'Calculus I' CREATE (s)-[:ENROLLED_IN]→(c)
mydb$ MATCH (s:Student), (c:Course) WHERE s.name = 'Charlie' AND c.title = 'Data Structures' CREATE (s)-[:ENROLLED_IN]→(c)
```



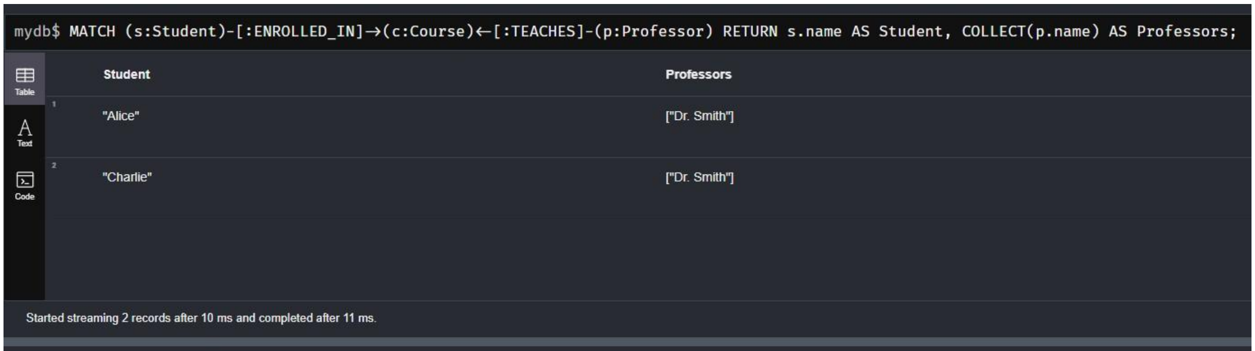
```
mydb$ MATCH (p:Professor), (c:Course) WHERE p.name = 'Dr. Smith' AND c.title = 'Introduction to Programming' CREATE (p)-[:TEACH…
mydb$ MATCH (p:Professor), (c:Course) WHERE p.name = 'Dr. Johnson' AND c.title = 'Calculus I' CREATE (p)-[:TEACHES]→(c)
mydb$ MATCH (p:Professor), (c:Course) WHERE p.name = 'Dr. Smith' AND c.title = 'Data Structures' CREATE (p)-[:TEACHES]→(c)
```



```
mydb$ call db.schema.visualization
```



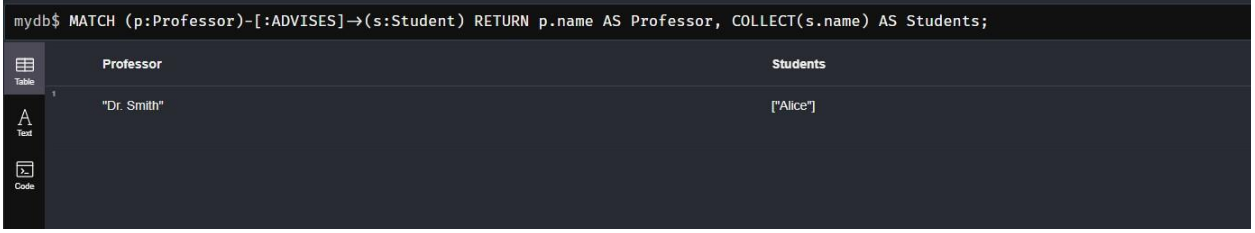
```
MATCH(n) RETURN n LIMIT 100
```



```
mydb$ MATCH (s:Student)-[:ENROLLED_IN]→(c:Course)←[:TEACHES]-(p:Professor) RETURN s.name AS Student, COLLECT(p.name) AS Professors;
```

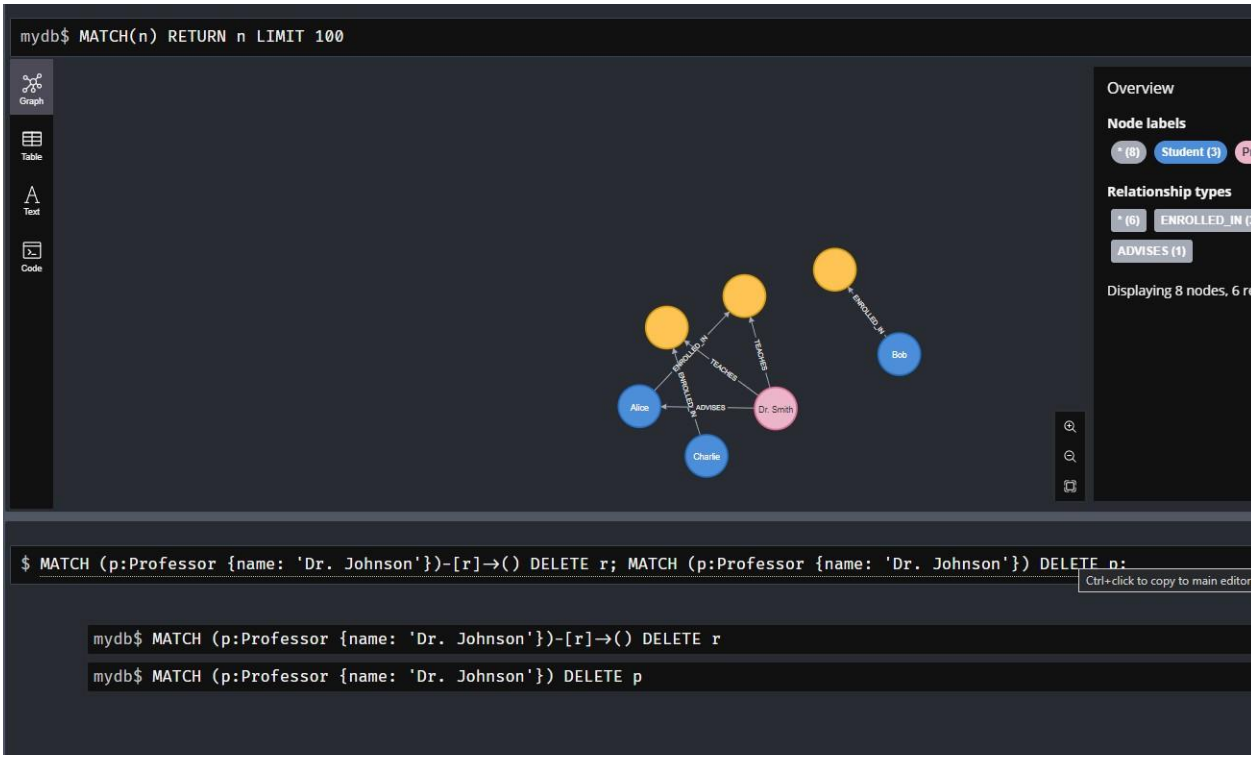
	Student	Professors
1	"Alice"	["Dr. Smith"]
2	"Charlie"	["Dr. Smith"]

Started streaming 2 records after 10 ms and completed after 11 ms.



```
mydb$ MATCH (p:Professor)-[:ADVISES]→(s:Student) RETURN p.name AS Professor, COLLECT(s.name) AS Students;
```

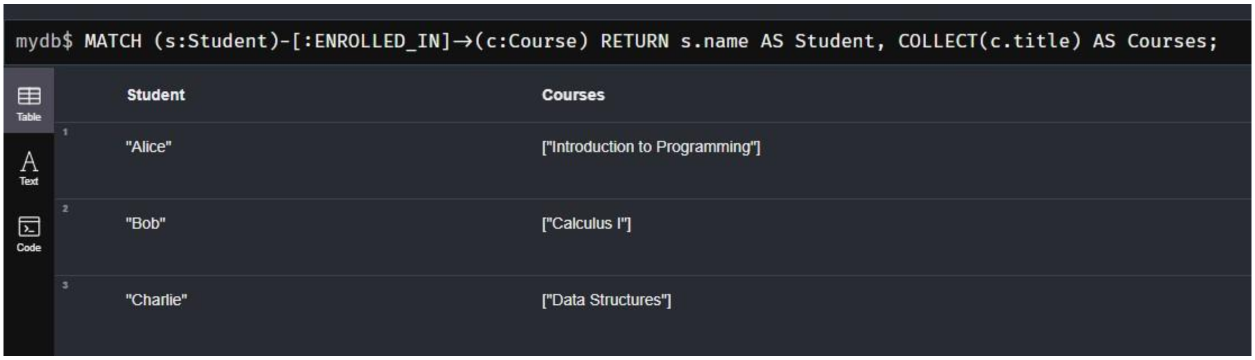
	Professor	Students
1	"Dr. Smith"	["Alice"]



```
mydb$ MATCH(n) RETURN n LIMIT 100
```

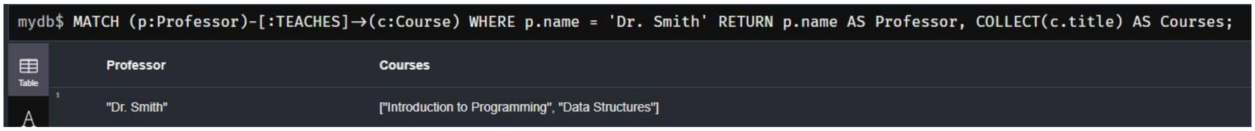
```
$ MATCH (p:Professor {name: 'Dr. Johnson'})-[r]→() DELETE r; MATCH (p:Professor {name: 'Dr. Johnson'}) DELETE p;
```

```
mydb$ MATCH (p:Professor {name: 'Dr. Johnson'})-[r]→() DELETE r
mydb$ MATCH (p:Professor {name: 'Dr. Johnson'}) DELETE p
```



```
mydb$ MATCH (s:Student)-[:ENROLLED_IN]->(c:Course) RETURN s.name AS Student, COLLECT(c.title) AS Courses;
```

	Student	Courses
1	"Alice"	["Introduction to Programming"]
2	"Bob"	["Calculus I"]
3	"Charlie"	["Data Structures"]



```
mydb$ MATCH (p:Professor)-[:TEACHES]->(c:Course) WHERE p.name = 'Dr. Smith' RETURN p.name AS Professor, COLLECT(c.title) AS Courses;
```

	Professor	Courses
1	"Dr. Smith"	["Introduction to Programming", "Data Structures"]

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>;
```

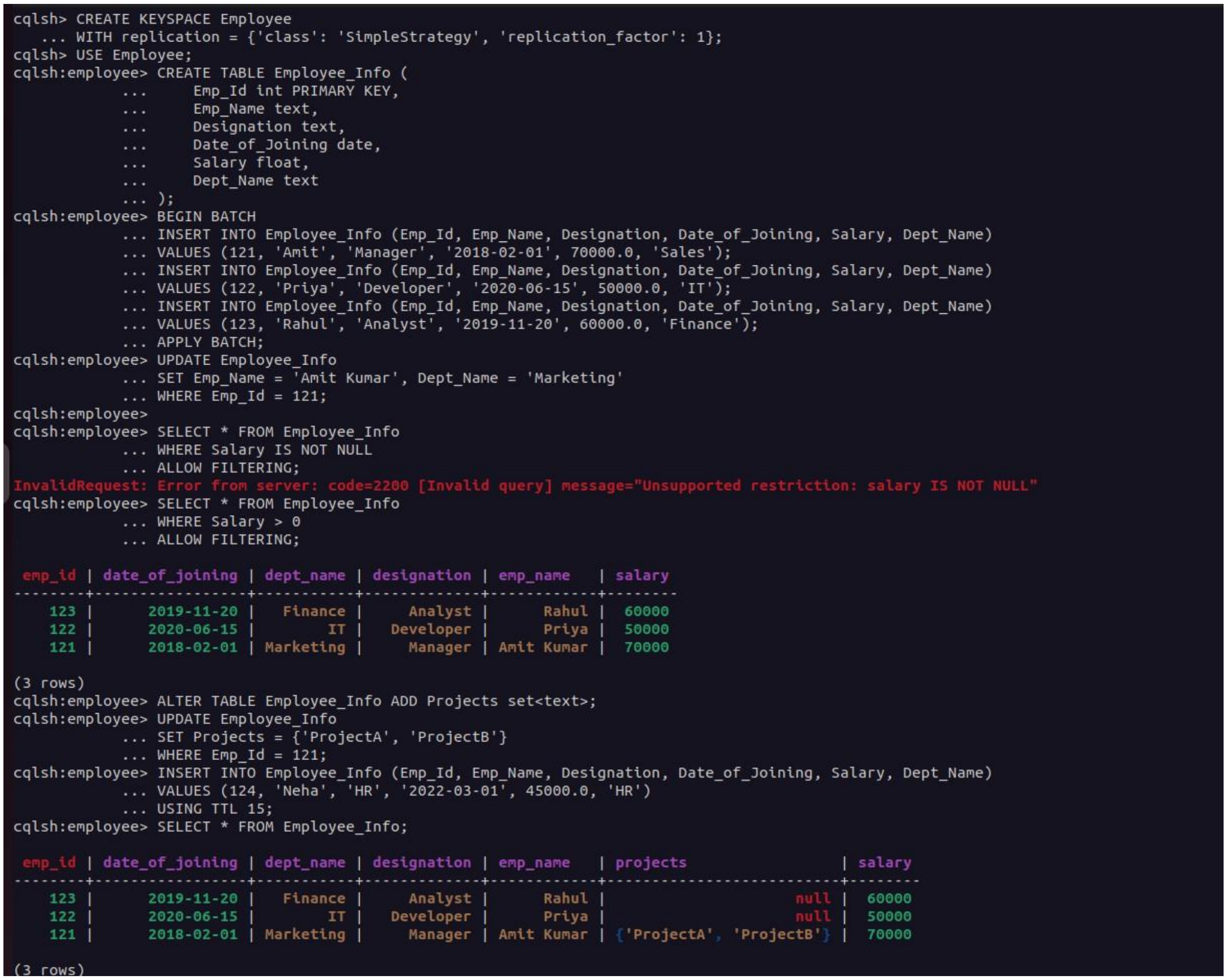
6.How do you update the projects of employee with Emp_Id = 121?

```
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;



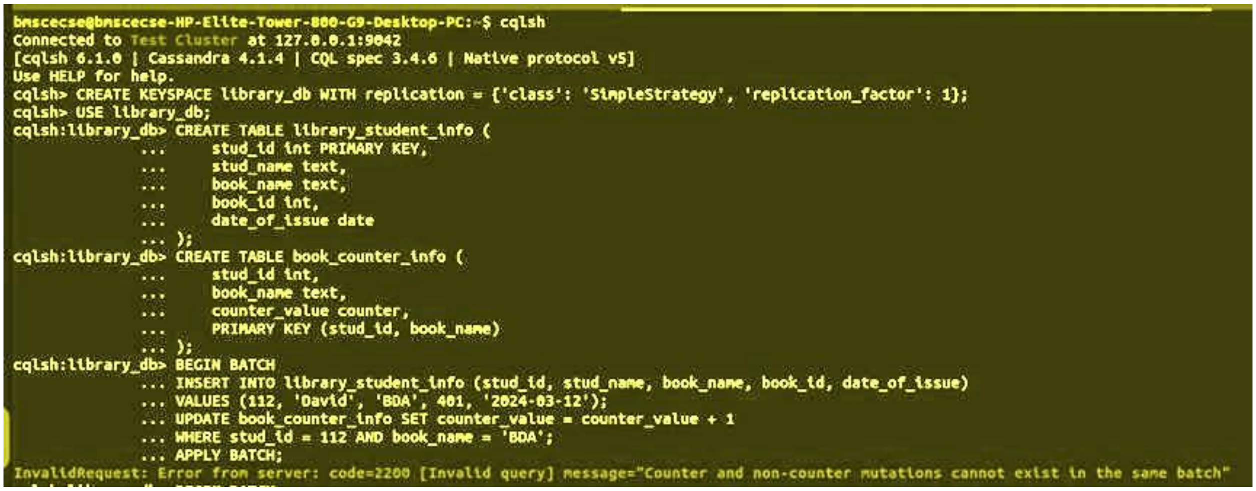
```
cqlsh> CREATE KEYSPACE Employee
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE Employee_Info (
            ...     Emp_Id int PRIMARY KEY,
            ...     Emp_Name text,
            ...     Designation text,
            ...     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

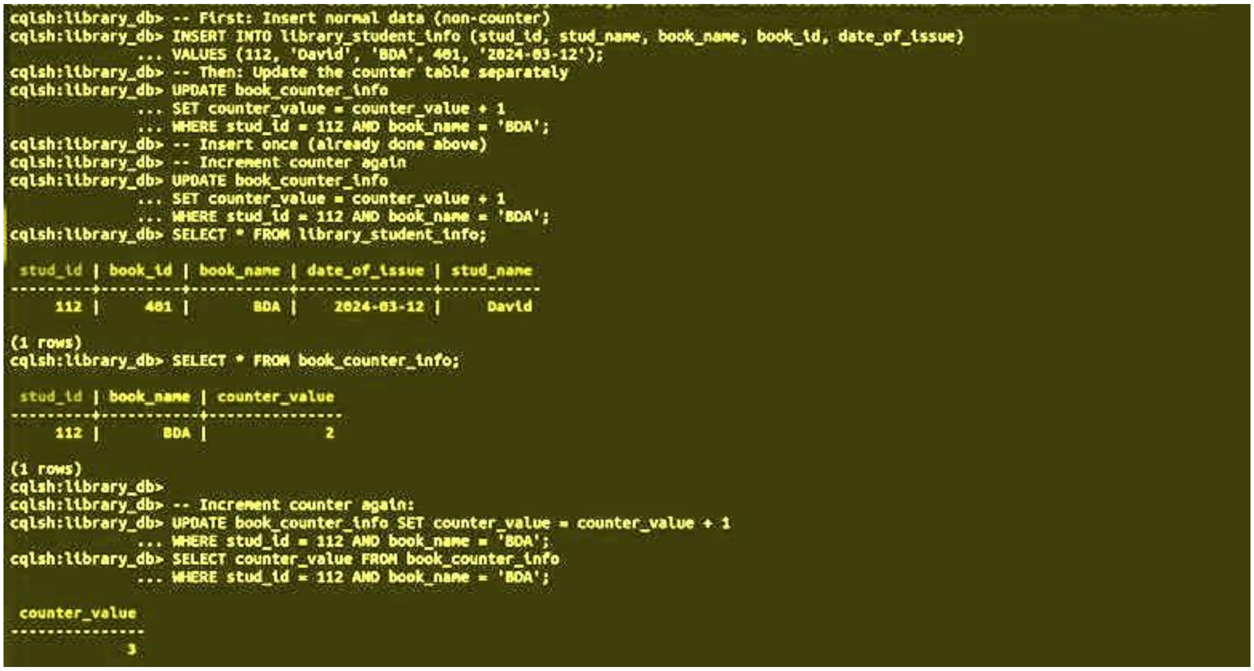


```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE library_db WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE library_db;
cqlsh:library_db> CREATE TABLE library_student_info (
              ...     stud_id int PRIMARY KEY,
              ...     stud_name text,
              ...     book_name text,
              ...     book_id int,
              ...     date_of_issue date
              ... );
cqlsh:library_db> CREATE TABLE book_counter_info (
              ...     stud_id int,
              ...     book_name text,
              ...     counter_value counter,
              ...     PRIMARY KEY (stud_id, book_name)
              ... );
cqlsh:library_db> BEGIN BATCH
              ... INSERT INTO library_student_info (stud_id, stud_name, book_name, book_id, date_of_issue)
              ... VALUES (112, 'David', 'BDA', 401, '2024-03-12');
              ... UPDATE book_counter_info SET counter_value = counter_value + 1
              ... WHERE stud_id = 112 AND book_name = 'BDA';
              ... APPLY BATCH;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Counter and non-counter mutations cannot exist in the same 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



```
cqlsh:library_db> -- First: Insert normal data (non-counter)
cqlsh:library_db> INSERT INTO library_student_info (stud_id, stud_name, book_name, book_id, date_of_issue)
              ... VALUES (112, 'David', 'BDA', 401, '2024-03-12');
cqlsh:library_db> -- Then: Update the counter table separately
cqlsh:library_db> UPDATE book_counter_info
              ... SET counter_value = counter_value + 1
              ... WHERE stud_id = 112 AND book_name = 'BDA';
cqlsh:library_db> -- Insert once (already done above)
cqlsh:library_db> -- Increment counter again
cqlsh:library_db> UPDATE book_counter_info
              ... SET counter_value = counter_value + 1
              ... WHERE stud_id = 112 AND book_name = 'BDA';
cqlsh:library_db> SELECT * FROM library_student_info;

 stud_id | book_id | book_name | date_of_issue | stud_name
---------+---------+-----------+---------------+-----------
     112 |     401 |       BDA |    2024-03-12 |     David

(1 rows)
cqlsh:library_db> SELECT * FROM book_counter_info;

 stud_id | book_name | counter_value
---------+-----------+---------------
     112 |       BDA |             2

(1 rows)
cqlsh:library_db>
cqlsh:library_db> -- Increment counter again:
cqlsh:library_db> UPDATE book_counter_info SET counter_value = counter_value + 1
              ... WHERE stud_id = 112 AND book_name = 'BDA';
cqlsh:library_db> SELECT counter_value FROM book_counter_info
              ... WHERE stud_id = 112 AND book_name = 'BDA';

 counter_value
---------------
             3
```

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.

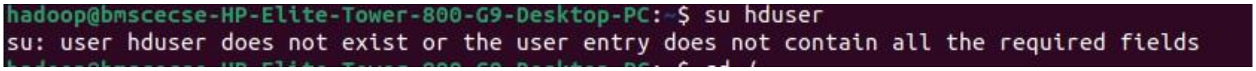
8. mv

Command:
```
hadoop fs -mv /abc /FFF
```
Description: Moves /abc directory in HDFS to /FFF.

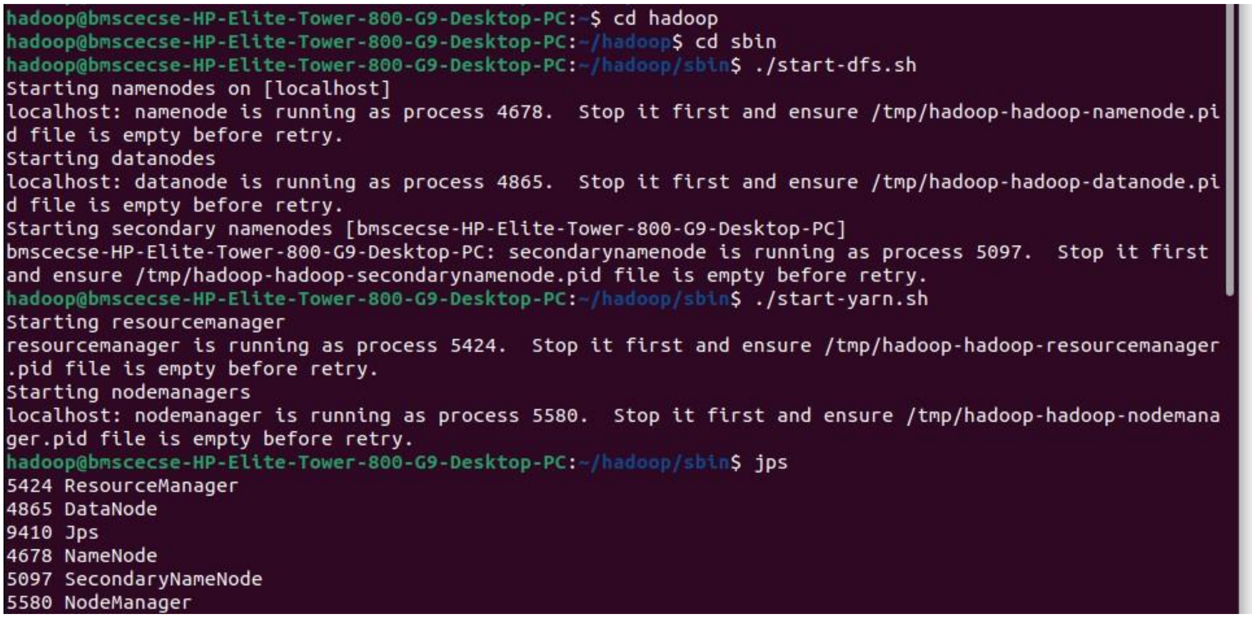
9. cp

Command:
```
hadoop fs -cp /CSE/ /LLL
```
Description: Copies contents from /CSE/ to /LLL within HDFS.

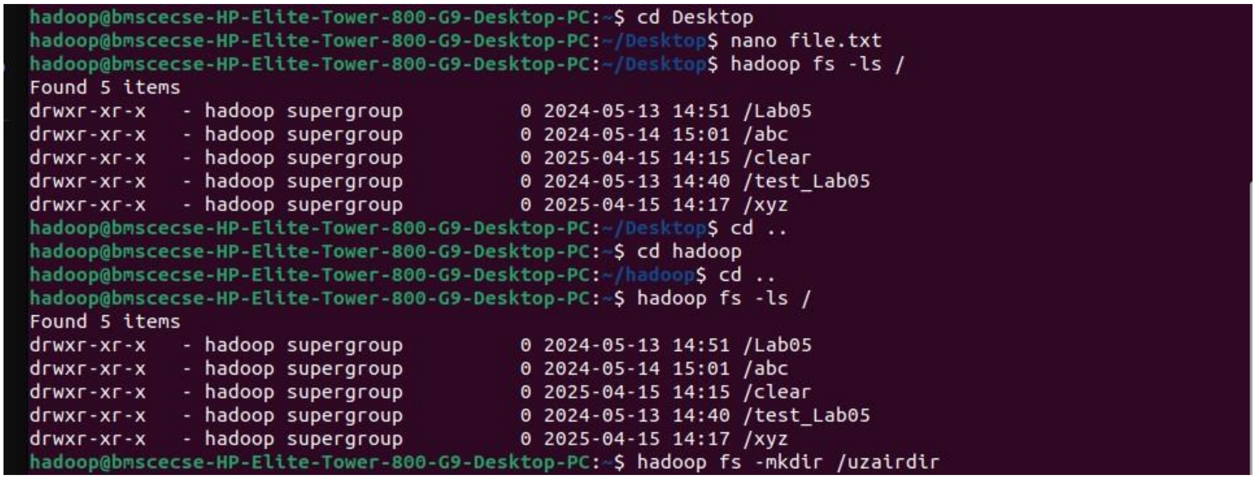
Screenshots



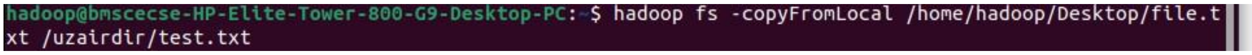
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ su hduser
su: user hduser does not exist or the user entry does not contain all the required fields
```



```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd hadoop
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ cd sbin
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode is running as process 4678.  Stop it first and ensure /tmp/hadoop-hadoop-namenode.pi
d file is empty before retry.
Starting datanodes
localhost: datanode is running as process 4865.  Stop it first and ensure /tmp/hadoop-hadoop-datanode.pi
d file is empty before retry.
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: secondarynamenode is running as process 5097.  Stop it first
and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop/sbin$ ./start-yarn.sh
Starting resourcemanager
resourcemanager is running as process 5424.  Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager
.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 5580.  Stop it first and ensure /tmp/hadoop-hadoop-nodemana
ger.pid file is empty before retry.
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop/sbin$ jps
5424 ResourceManager
4865 DataNode
9410 Jps
4678 NameNode
5097 SecondaryNameNode
5580 NodeManager
```

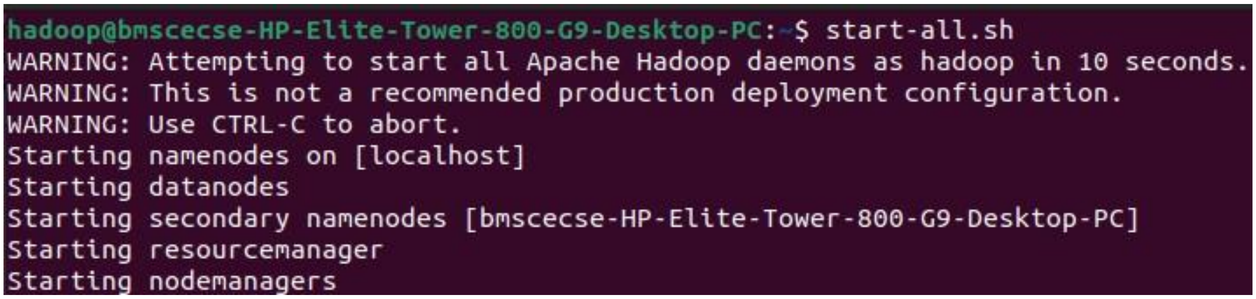


```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd Desktop
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano file.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /
Found 5 items
drwxr-xr-x   - hadoop supergroup          0 2024-05-13 14:51 /Lab05
drwxr-xr-x   - hadoop supergroup          0 2024-05-14 15:01 /abc
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 14:15 /clear
drwxr-xr-x   - hadoop supergroup          0 2024-05-13 14:40 /test_Lab05
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 14:17 /xyz
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ cd ..
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd hadoop
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/hadoop$ cd ..
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
Found 5 items
drwxr-xr-x   - hadoop supergroup          0 2024-05-13 14:51 /Lab05
drwxr-xr-x   - hadoop supergroup          0 2024-05-14 15:01 /abc
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 14:15 /clear
drwxr-xr-x   - hadoop supergroup          0 2024-05-13 14:40 /test_Lab05
drwxr-xr-x   - hadoop supergroup          0 2025-04-15 14:17 /xyz
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /uzairdir
```



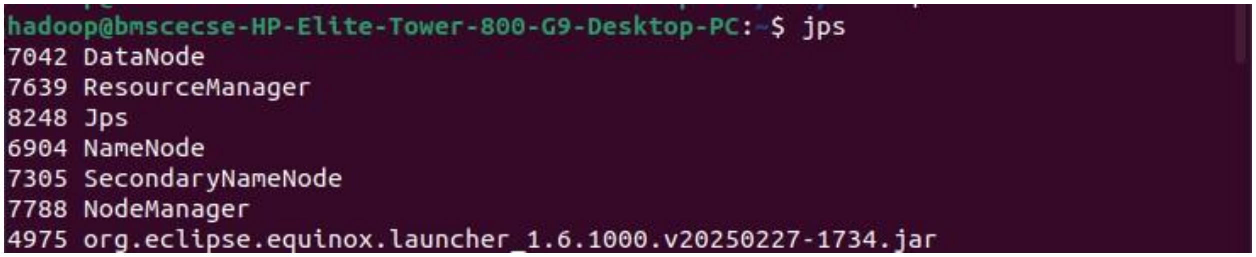
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/file.t
xt /uzairdir/test.txt
```

Lab 7 Word Count using Map-Reduce



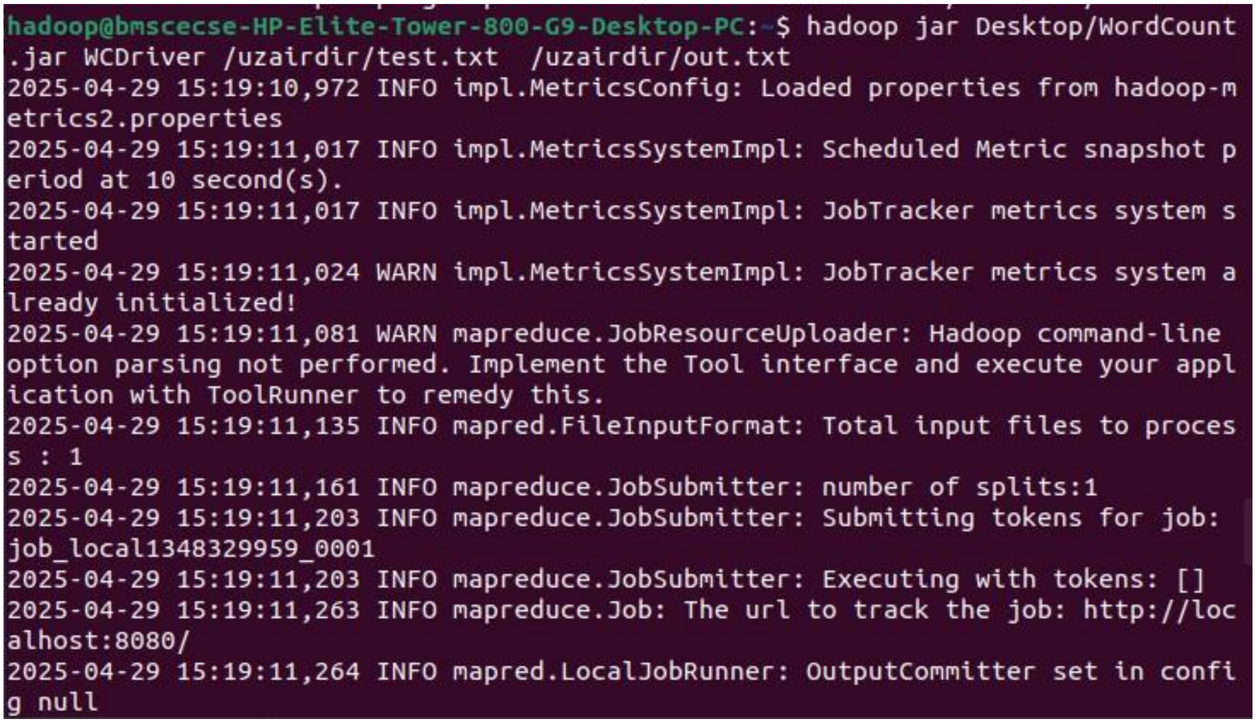
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes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



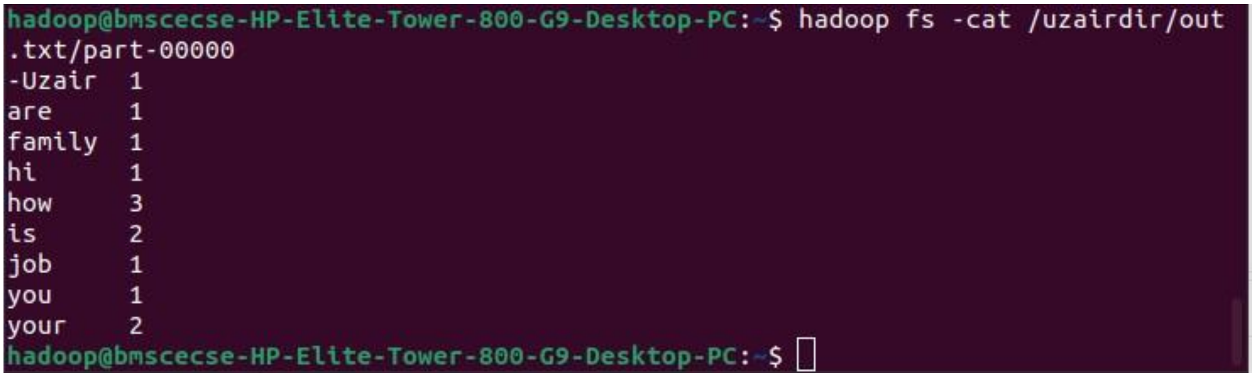
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ jps
7042 DataNode
7639 ResourceManager
8248 Jps
6904 NameNode
7305 SecondaryNameNode
7788 NodeManager
4975 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
```

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



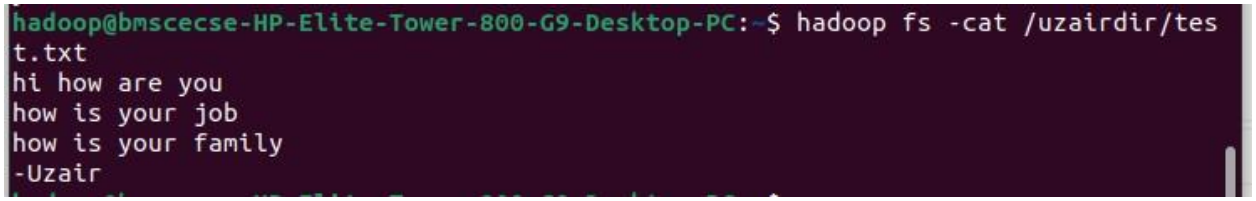
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar Desktop/WordCount
.jar WCDriver /uzairdir/test.txt  /uzairdir/out.txt
2025-04-29 15:19:10,972 INFO impl.MetricsConfig: Loaded properties from hadoop-m
etrics2.properties
2025-04-29 15:19:11,017 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot p
eriod at 10 second(s).
2025-04-29 15:19:11,017 INFO impl.MetricsSystemImpl: JobTracker metrics system s
tarted
2025-04-29 15:19:11,024 WARN impl.MetricsSystemImpl: JobTracker metrics system a
lready initialized!
2025-04-29 15:19:11,081 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your appl
ication with ToolRunner to remedy this.
2025-04-29 15:19:11,135 INFO mapred.FileInputFormat: Total input files to proces
s : 1
2025-04-29 15:19:11,161 INFO mapreduce.JobSubmitter: number of splits:1
2025-04-29 15:19:11,203 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_local1348329959_0001
2025-04-29 15:19:11,203 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-29 15:19:11,263 INFO mapreduce.Job: The url to track the job: http://loc
alhost:8080/
2025-04-29 15:19:11,264 INFO mapred.LocalJobRunner: OutputCommitter set in confi
g null
```

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:~$ hadoop fs -cat /uzairdir/out
.txt/part-00000
-Uzair  1
are     1
family  1
hi      1
how     3
is      2
job     1
you     1
your    2
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ 
```

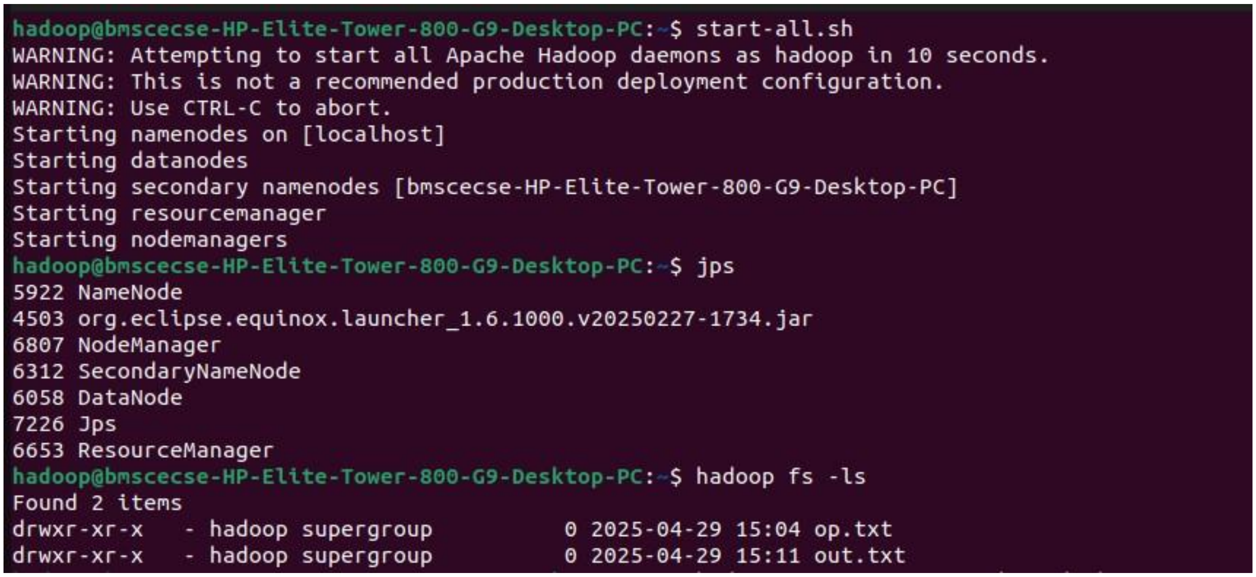
The output of the MapReduce job is displayed using hadoop fs -cat, showing the word count of the input file.



```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /uzairdir/tes
t.txt
hi how are you
how is your job
how is your family
-Uzair
```

The contents of the input file test.txt are displayed using hadoop fs -cat, showing a text conversation.

Lab 8 Mean-Max Temperature using Map-Reduce

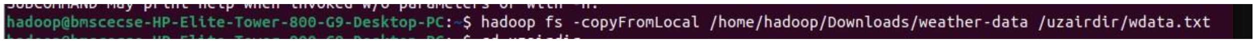


```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes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

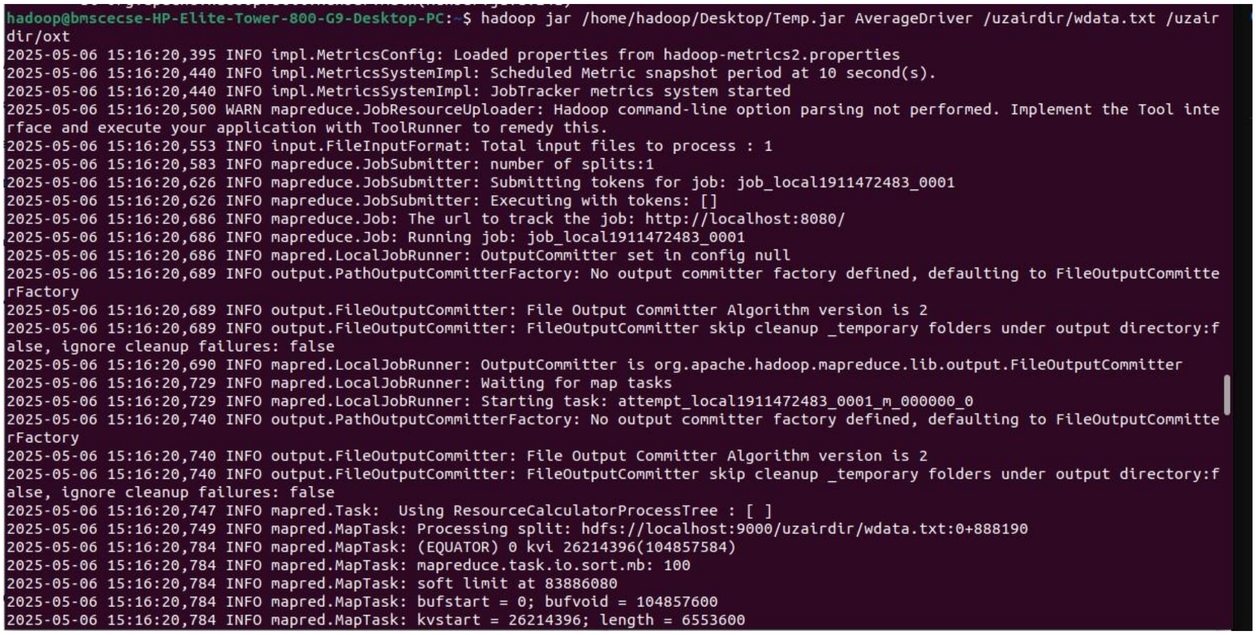
The jps command confirms active Hadoop services such as NameNode, DataNode, and ResourceManager are running.

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



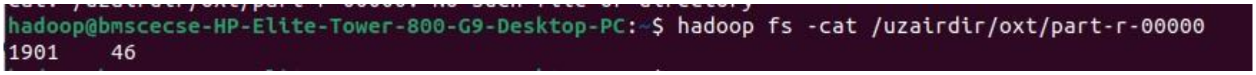
```
SUBCOMMAND may print help when invoked w/o parameters or with -h.
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Downloads/weather-data /uzairdir/wdata.txt
```

A local file weather-data.txt is copied to HDFS at /uzairdir/wdata.txt using the copyFromLocal command.



```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/Temp.jar AverageDriver /uzairdir/wdata.txt /uzair
dir/oxt
2025-05-06 15:16:20,395 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-06 15:16:20,440 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-06 15:16:20,440 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-06 15:16:20,500 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool inte
rface and execute your application with ToolRunner to remedy this.
2025-05-06 15:16:20,553 INFO input.FileInputFormat: Total input files to process : 1
2025-05-06 15:16:20,583 INFO mapreduce.JobSubmitter: number of splits:1
2025-05-06 15:16:20,626 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1911472483_0001
2025-05-06 15:16:20,626 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-05-06 15:16:20,686 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-05-06 15:16:20,686 INFO mapreduce.Job: Running job: job_local1911472483_0001
2025-05-06 15:16:20,686 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-05-06 15:16:20,689 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitte
rFactory
2025-05-06 15:16:20,689 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 15:16:20,689 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2025-05-06 15:16:20,690 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-05-06 15:16:20,729 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-05-06 15:16:20,729 INFO mapred.LocalJobRunner: Starting task: attempt_local1911472483_0001_m_000000_0
2025-05-06 15:16:20,740 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitte
rFactory
2025-05-06 15:16:20,740 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-05-06 15:16:20,740 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2025-05-06 15:16:20,747 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-05-06 15:16:20,749 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/uzairdir/wdata.txt:0+888190
2025-05-06 15:16:20,784 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-05-06 15:16:20,784 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2025-05-06 15:16:20,784 INFO mapred.MapTask: soft limit at 83886080
2025-05-06 15:16:20,784 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2025-05-06 15:16:20,784 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
```

A MapReduce job is executed using the AverageDriver class to process wdata.txt and save results in oxt.

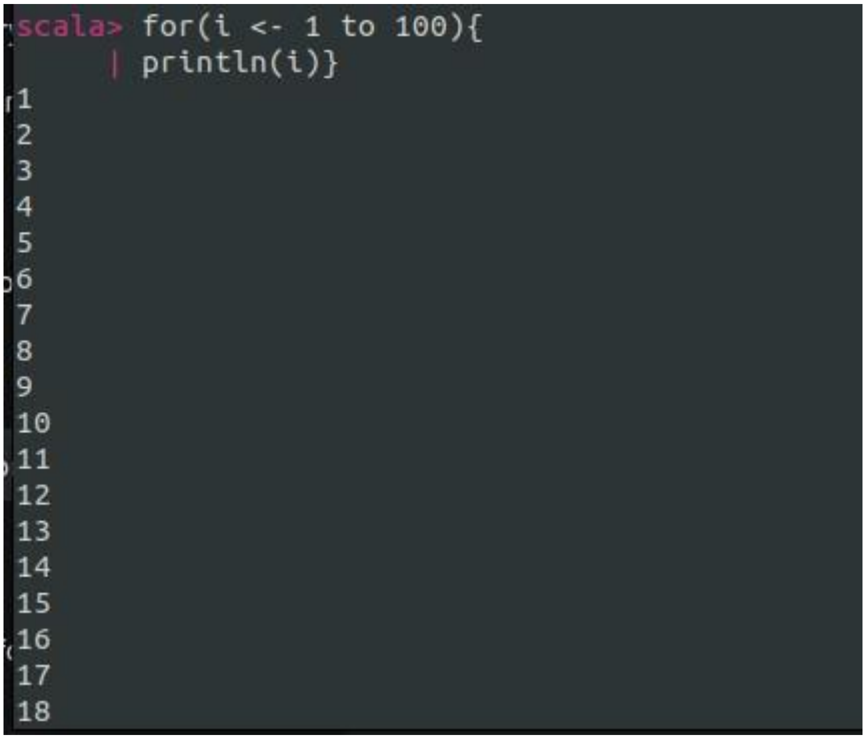


```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /uzairdir/oxt/part-r-00000
1901    46
```

The output of the MapReduce job is viewed using hadoop fs -cat, showing results from the oxt/part-r-00000 file.

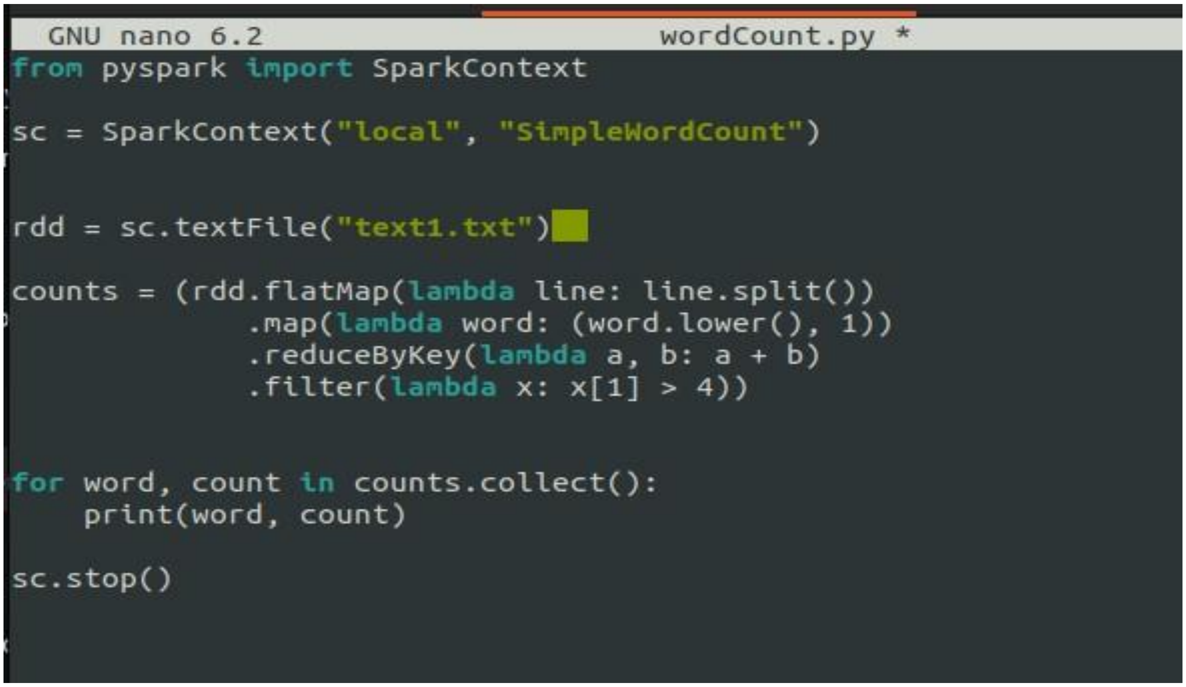
Lab 9 Scala and pySpark

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



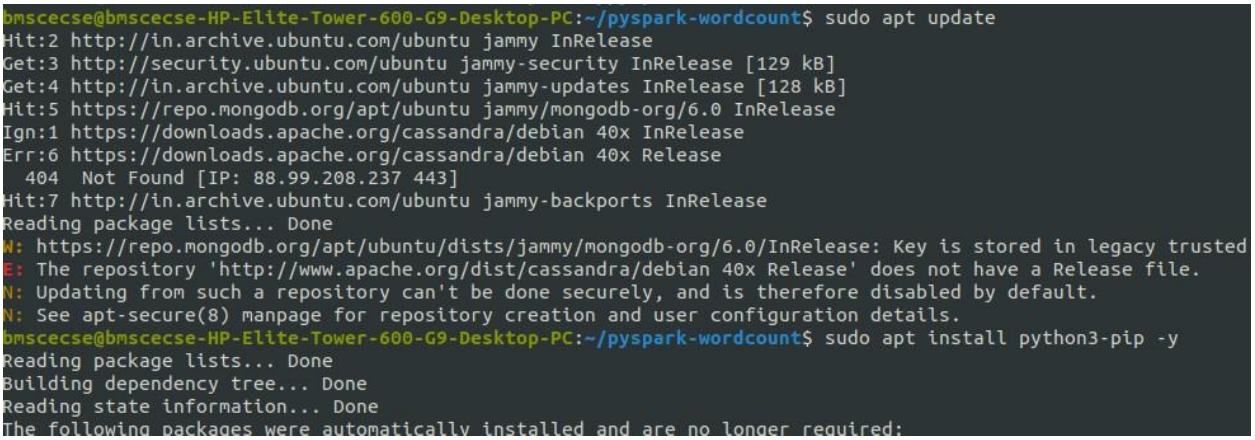
```
scala> for(i <- 1 to 100){
     | println(i)}
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

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.

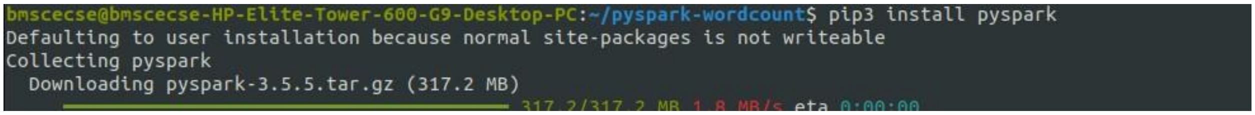


```
  GNU nano 6.2                          wordCount.py *
from pyspark import SparkContext

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()
```

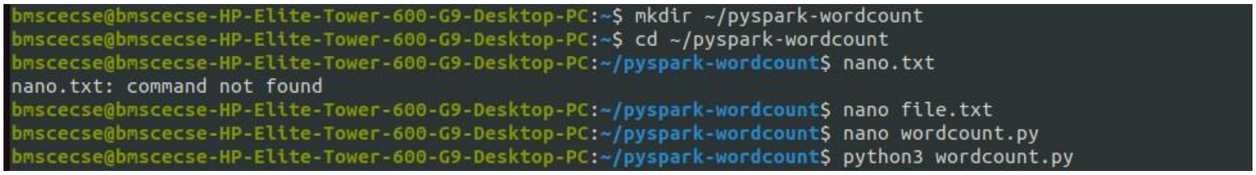
Spark Shell Execution Screenshots



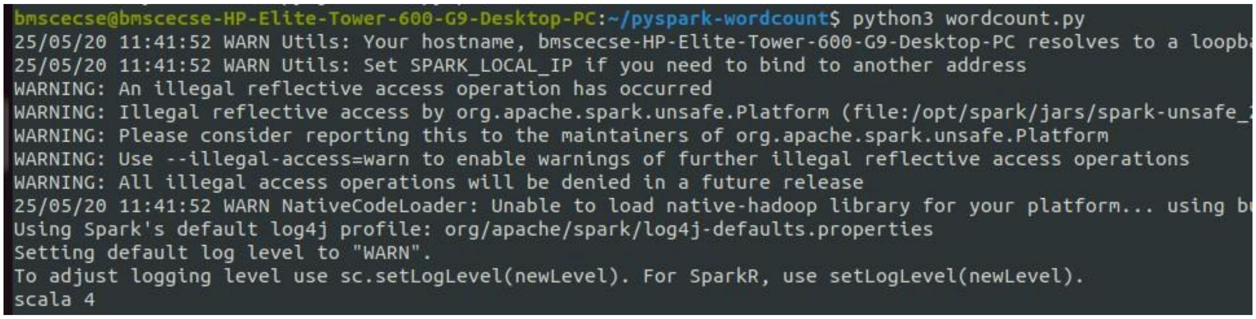
```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ sudo apt update
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:5 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
```

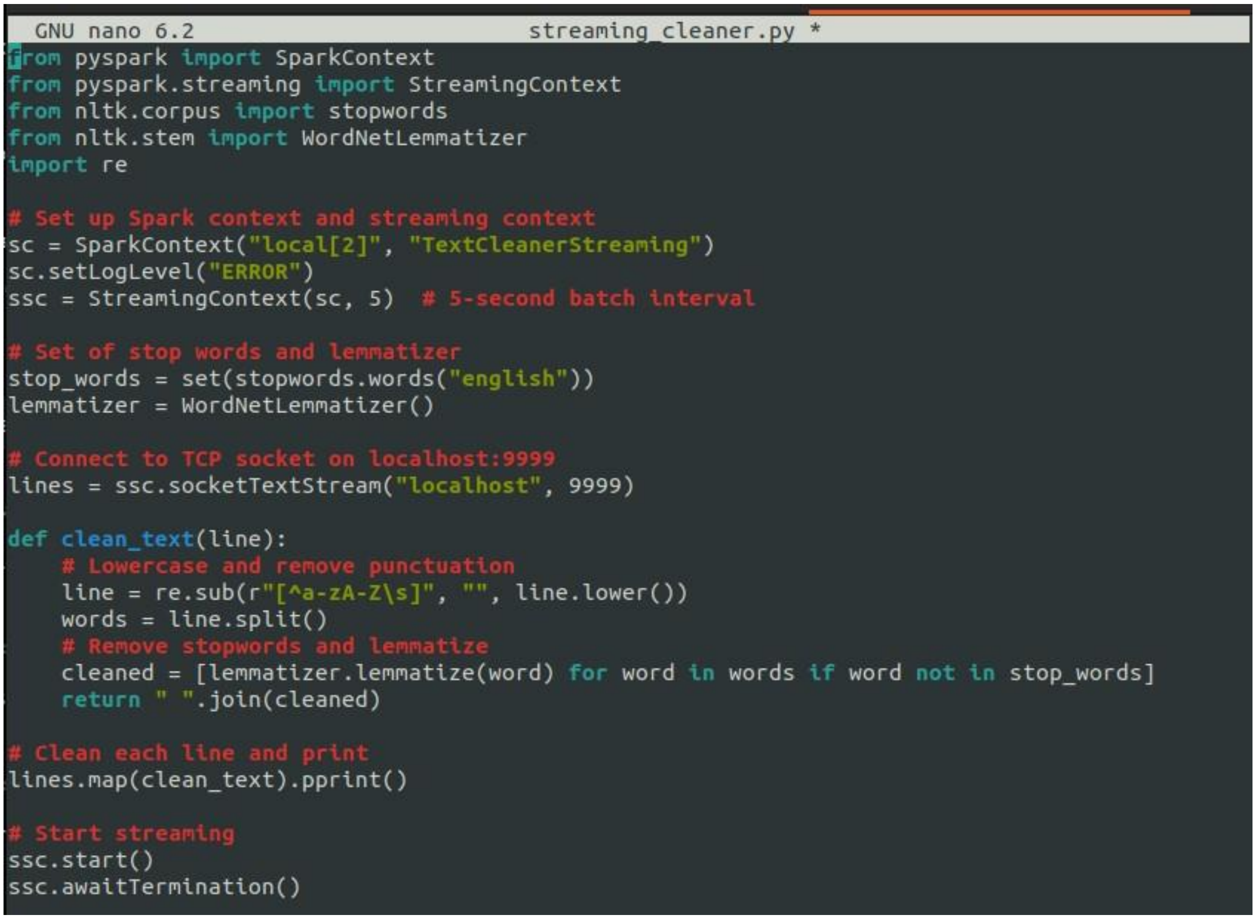


```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ mkdir ~/pyspark-wordcount
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ cd ~/pyspark-wordcount
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ nano.txt
nano.txt: command not found
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$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

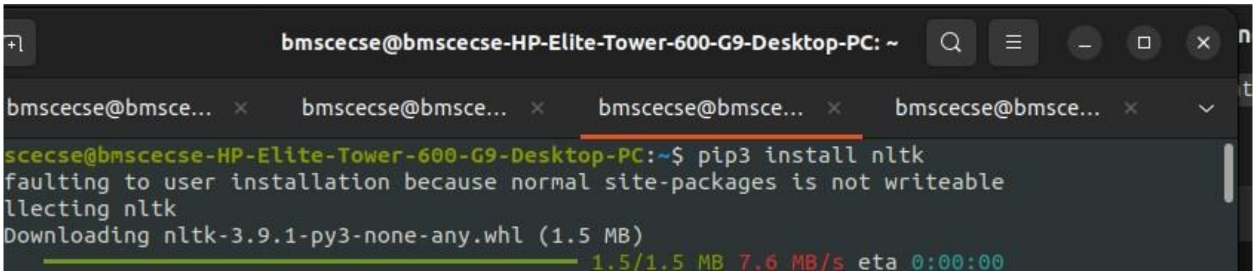


```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/pyspark-wordcount$ python3 wordcount.py
25/05/20 11:41:52 WARN Utils: Your hostname, bmscecse-HP-Elite-Tower-600-G9-Desktop-PC resolves to a loopba
25/05/20 11:41:52 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/opt/spark/jars/spark-unsafe_2
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 bu
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.



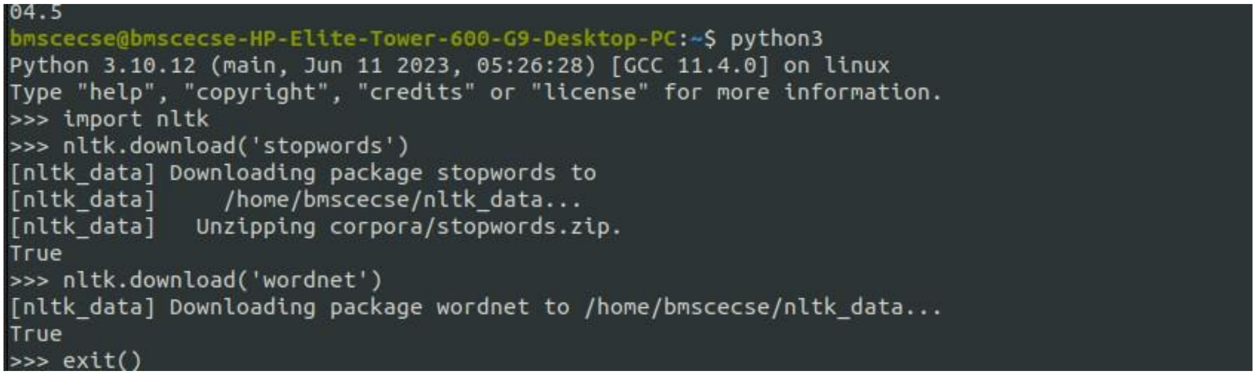
```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re

# 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()
```



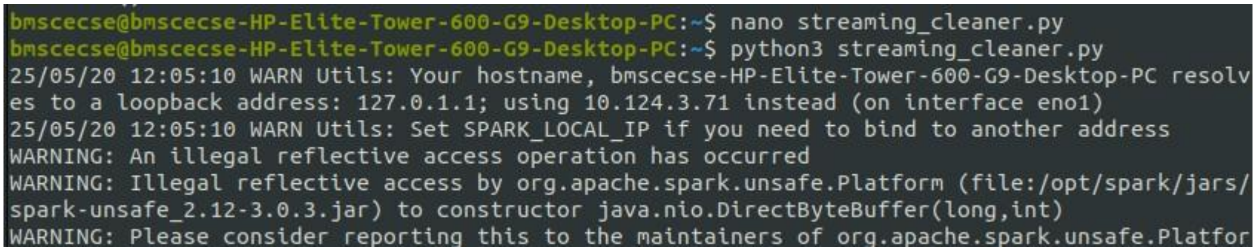
```
scecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ pip3 install nltk
faulting to user installation because normal site-packages is not writeable
llecting nltk
Downloading nltk-3.9.1-py3-none-any.whl (1.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.5/1.5 MB 7.6 MB/s eta 0:00:00
```

Installation of Natural Language Toolkit (nltk)

nlt

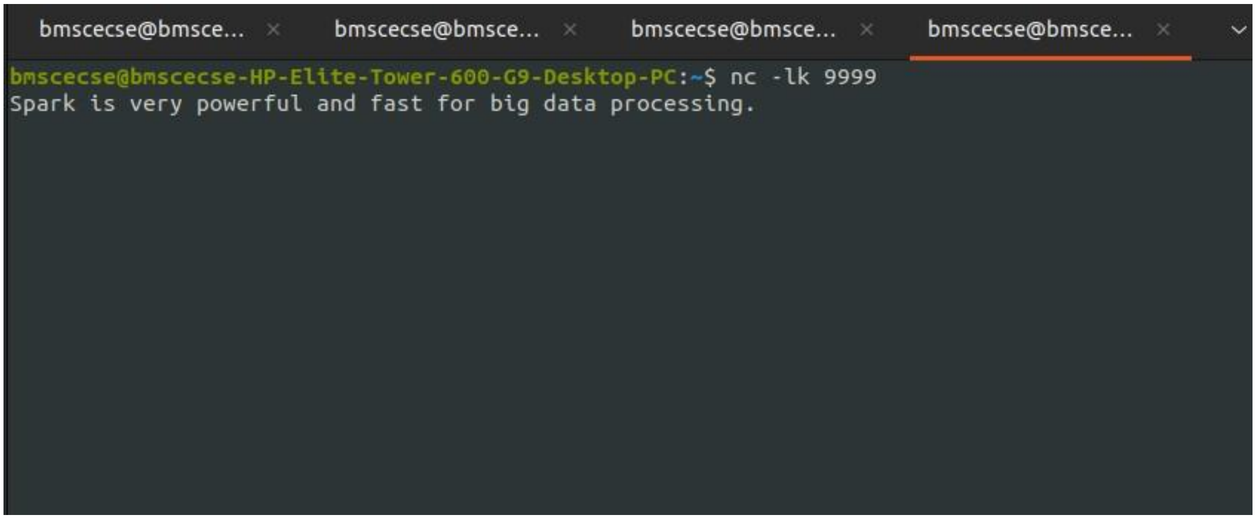


```
04.5
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ python3
Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download('stopwords')
[nltk_data] Downloading package stopwords to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



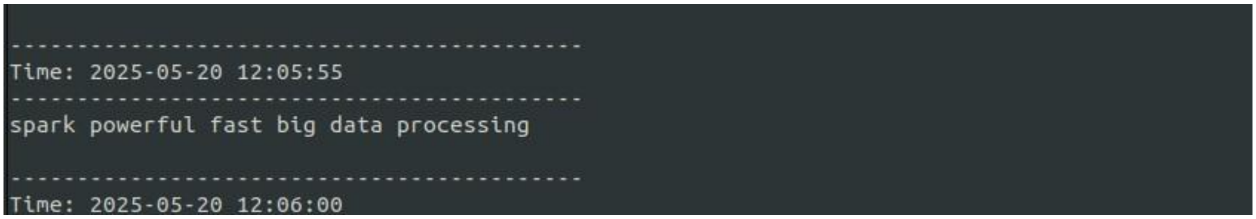
```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ nano streaming_cleaner.py
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ python3 streaming_cleaner.py
25/05/20 12:05:10 WARN Utils: Your hostname, bmscecse-HP-Elite-Tower-600-G9-Desktop-PC resolv
es to a loopback address: 127.0.1.1; using 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



```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ nc -lk 9999
Spark is very powerful and fast for big data processing.
```

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



```
-------------------------------------------
Time: 2025-05-20 12:05:55
-------------------------------------------
spark powerful fast big data processing

-------------------------------------------
Time: 2025-05-20 12:06:00
```

Output- cleaned data