VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT

on

Big Data Analytics

Submitted by

NAVANEETH V N (1BM22CS171)

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 **NAVANEETH V N(1BM22CS171),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics - (23CS6PCBDA)** work prescribed for the said degree.

RAMYA K M	**Dr. Kavitha Sooda**
Asst. Professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

`

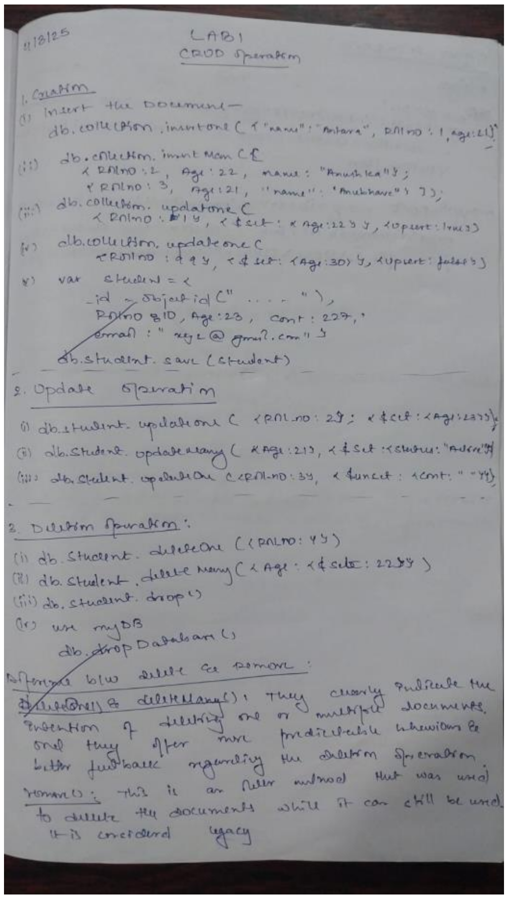
Index Sheet

Sl. No.	Experiment Title	Page No.
1	MongoDB- CRUD Operations Demonstration (Practice and Self Study)	1
2	MongoDB	3
3	Neo4j	9
4	Cassandra exploration	10
5	Cassandra Employee and library	12
6	HDFS commands	16
7	HDFS Word Count	18
8	HDFS Mean Temperature	22
9	SCALA using Spark	25

Course Outcome

CO1	Apply the concepts of NoSQL, Hadoop, Spark for a given task
CO2	Analyse data analytic techniques for a given problem .
CO3	Conduct experiments using data analytics mechanisms for a given problem.

Experiment – 1 Explore MongoDB



11/8/25

LAB1

CRUD Operation

1. Creation

(i) Insert the Document—

db.collection.insertOne ({"name": "Antara", Rollno : 1, age:21}

(ii) db.collection.insertMany ([
 { Rollno : 2 , Age : 22 , name : "Anushka" } ;
 { Rollno : 3, Age : 21, "name" : "Anubhave" }]);

(iii) db.collection. updateOne (
 { Rollno : 1 } , { $set : { Age : 22 } } , {upsert : true})

(iv) db.collection. updateOne (
 { Rollno : 99 } , { $ set : {Age : 30} } , {upsert : false})

(v) var student = {
 _id : Objectid (" . . . . "),
 Rollno : 10, Age : 23, cont : 227,
 email : " xyz@gmail.com" }
 db.student. save (student)

2. Update Operation

(i) db.student. updateOne ({Rollno : 2} ; { $set : {Age : 23} })

(ii) db.Student. updateMany ({Age : 21} , { $ Set : {status : "Active"} })

(iii) db.Student. updateOne ({Roll-no : 3}, { $unset : {cont : ""} })

3. Deletion Operation :

(i) db. Student. deleteOne ({Rollno : 4})

(ii) db. Student . delete Many ({ Age : { $ gte : 22 } })

(iii) db. Student. drop ()

(iv) use myDB
 db.dropDatabase ()

Difference b/w delete & remove :

deleteOne() & deleteMany() : They clearly indicate the intention of deleting one or multiple documents, and they offer more predictable behaviour & better feedback regarding the deletion operation.

remove() : This is an older method that was used to delete the documents while it can still be used, it is considered legacy

Code with Output:

```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.createCollection('Customers')
{ ok: 1 }
```

```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.createCollection('Student')
{ ok: 1 }
```

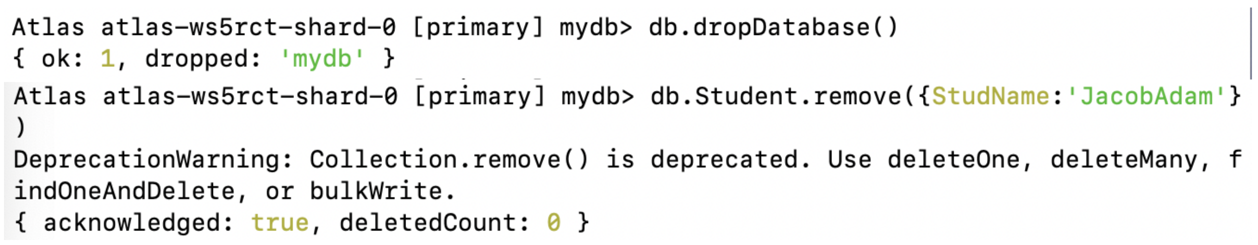
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.createCollection('Student')
{ ok: 1 }
```

```
Atlas atlas-ws5rct-shard-0 [primary] test> use mydb
switched to db mydb
```

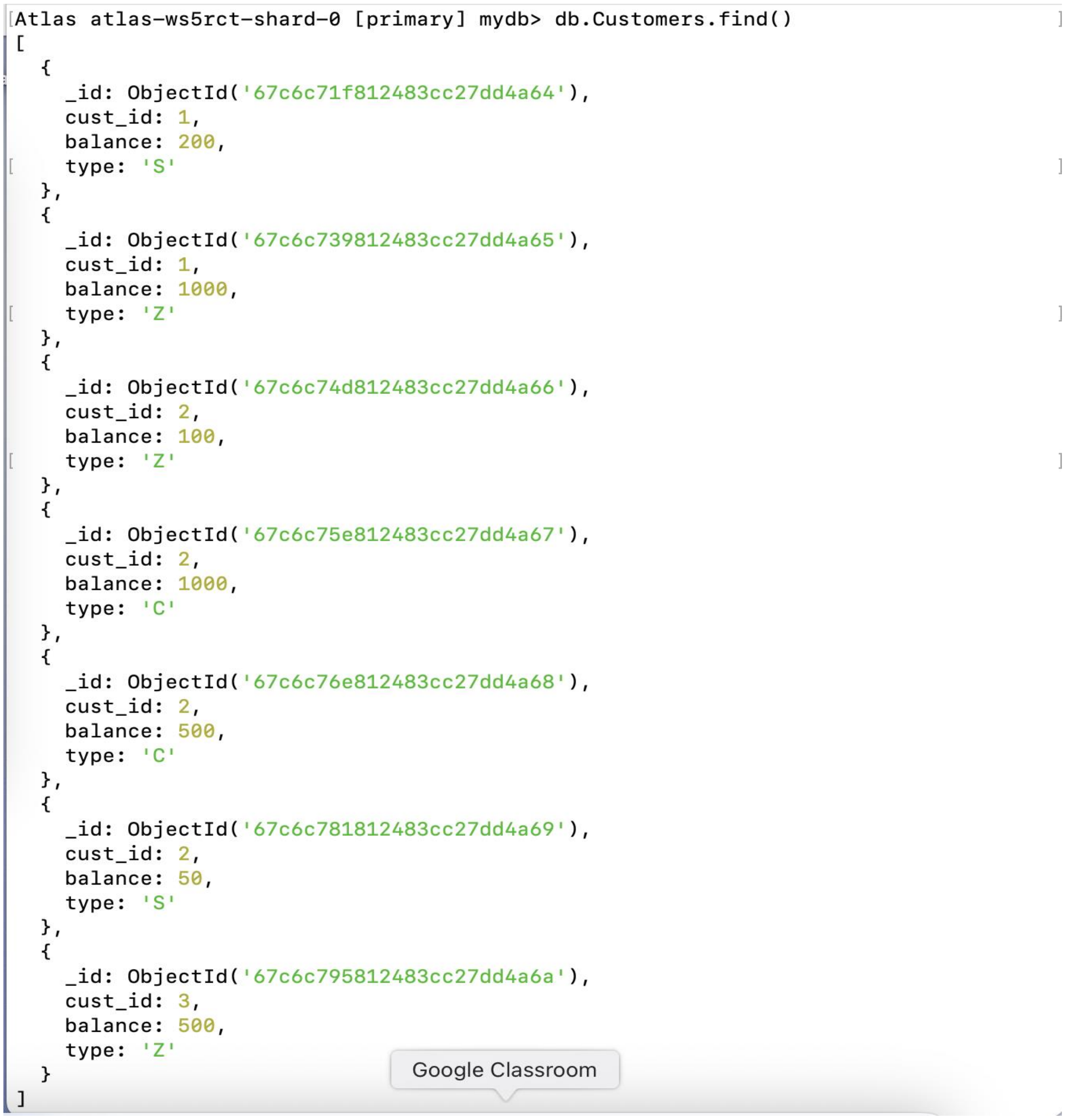
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.deleteMany({Grade:'VII'})
{ acknowledged: true, deletedCount: 3 }
```

```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.deleteOne({StudName:'JacobAda
m'})
{ acknowledged: true, deletedCount: 0 }
```

```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.drop()
true
```

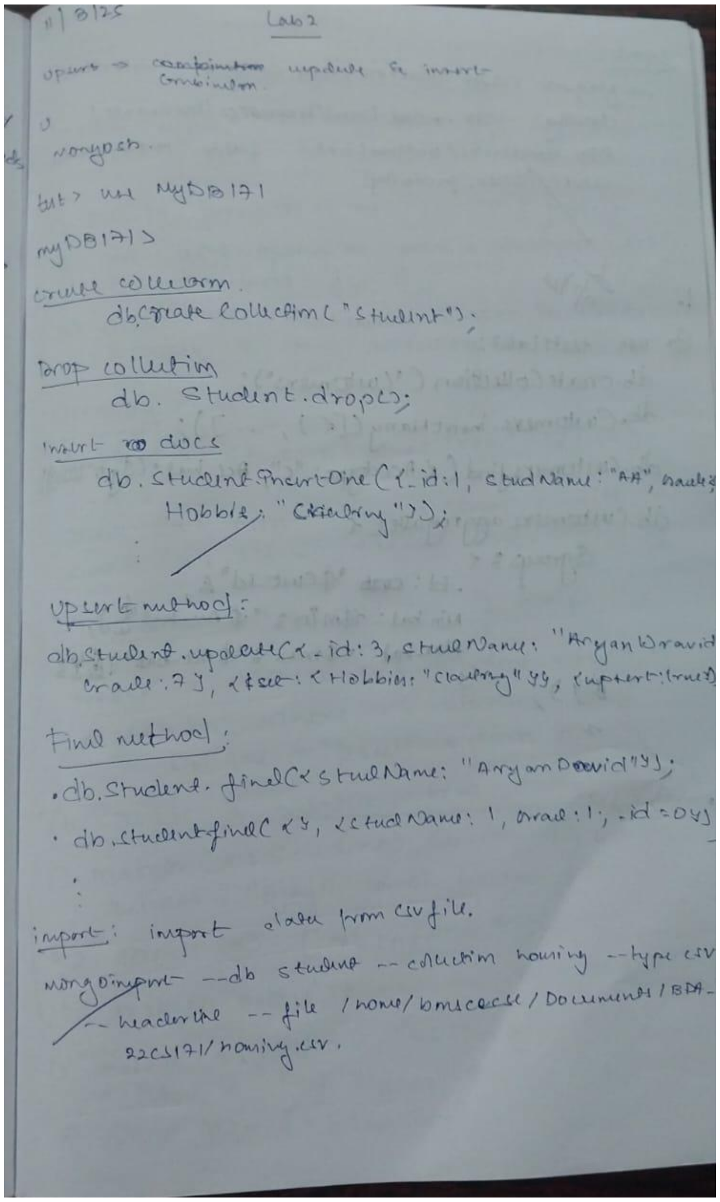


```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.dropDatabase()
{ ok: 1, dropped: 'mydb' }
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.remove({StudName:'JacobAdam'}
)
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, f
indOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 0 }
```



```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.find()
[
  {
    _id: ObjectId('67c6c71f812483cc27dd4a64'),
    cust_id: 1,
    balance: 200,
    type: 'S'
  },
  {
    _id: ObjectId('67c6c739812483cc27dd4a65'),
    cust_id: 1,
    balance: 1000,
    type: 'Z'
  },
  {
    _id: ObjectId('67c6c74d812483cc27dd4a66'),
    cust_id: 2,
    balance: 100,
    type: 'Z'
  },
  {
    _id: ObjectId('67c6c75e812483cc27dd4a67'),
    cust_id: 2,
    balance: 1000,
    type: 'C'
  },
  {
    _id: ObjectId('67c6c76e812483cc27dd4a68'),
    cust_id: 2,
    balance: 500,
    type: 'C'
  },
  {
    _id: ObjectId('67c6c781812483cc27dd4a69'),
    cust_id: 2,
    balance: 50,
    type: 'S'
  },
  {
    _id: ObjectId('67c6c795812483cc27dd4a6a'),
    cust_id: 3,
    balance: 500,
    type: 'Z'
  }
]
```

Experiment – 2 MongoDB commands



11/3/25

Lab 2

upsert → ~~[illegible]~~ update & insert
combination

mongosh.

test > use MyDB171

myDB171 >

create collection.

```
db.createCollection("Student");
```

Drop collection

```
db.Student.drop();
```

insert ~~[illegible]~~ docs

```
db.Student.insertOne({_id:1, studName:"AA", grade:
    Hobbie: "Cricket"});
```

upsert method:

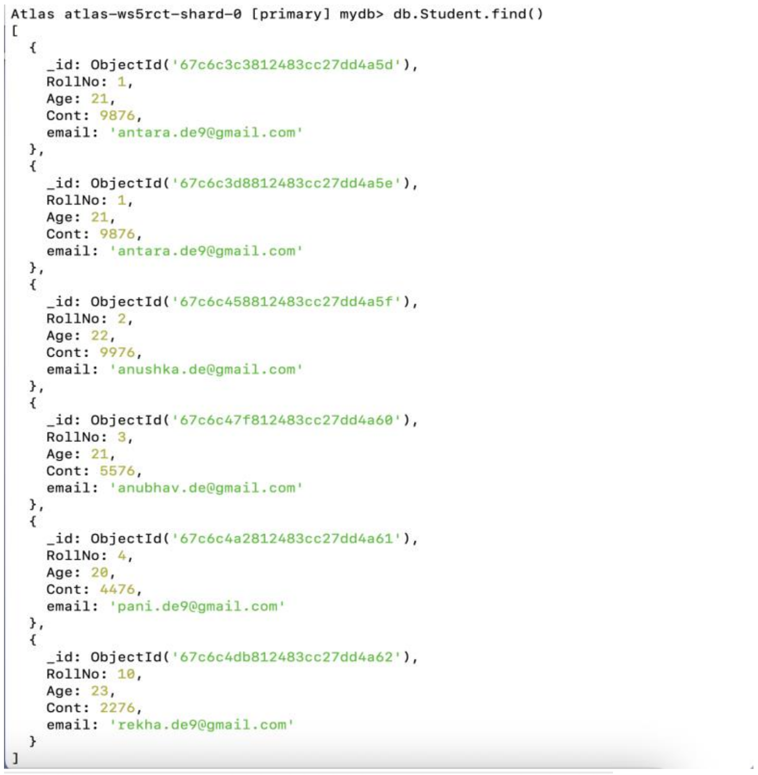
```
db.Student.update({_id:3, studName: "AryanDravid
  Grade:7}, {$set: {Hobbies:"Cricket"}}, {upsert:true})
```

Find method:

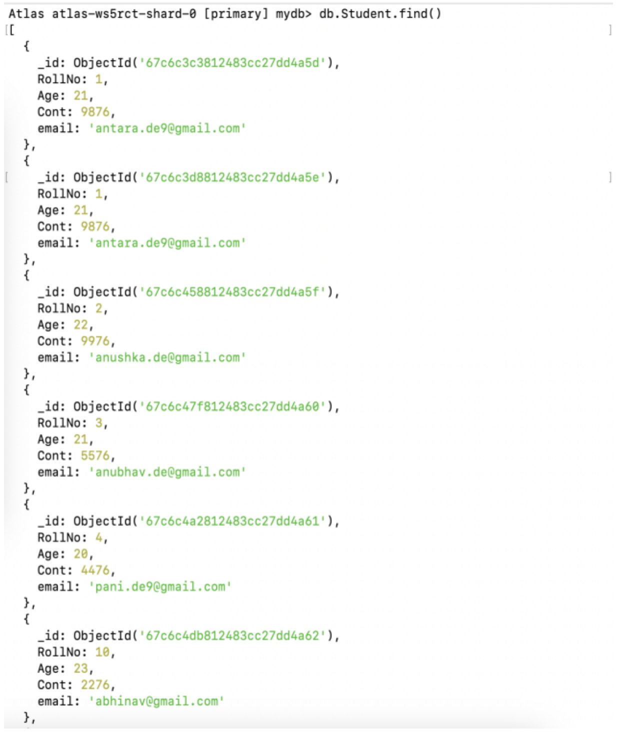
```
• db.Student.find({studName: "AryanDavid"});
• db.Student.find({}, {studName: 1, Grade:1, _id=0})
```

import: import data from csv file.

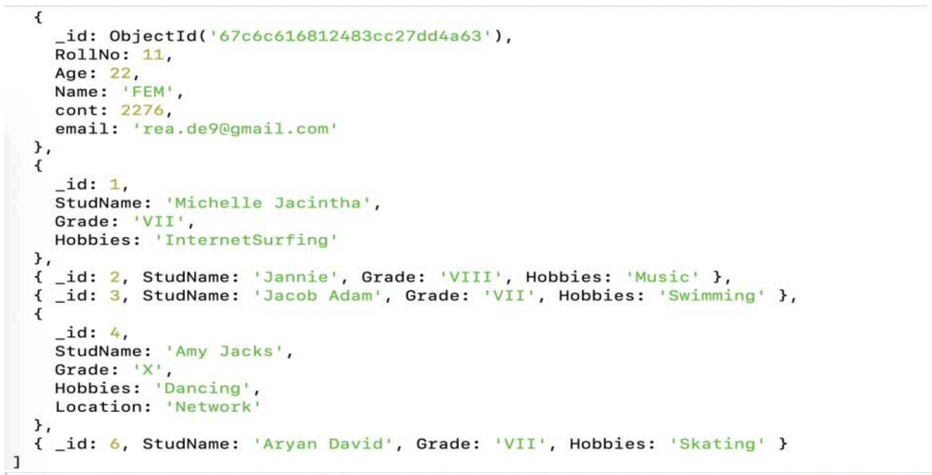
```
mongoimport --db student --collection housing --type csv
  --headerline --file /home/bmscecse/Documents/BDA-
    22CS171/housing.csv.
```



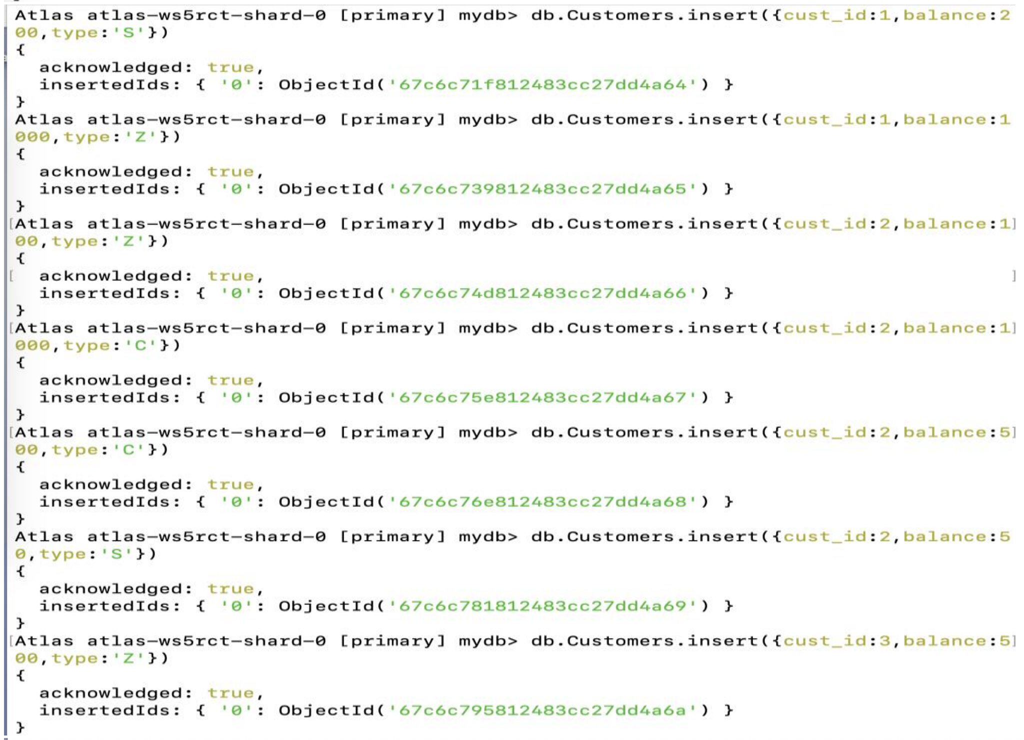
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.find()
[
  {
    _id: ObjectId('67c6c3c3812483cc27dd4a5d'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c3d8812483cc27dd4a5e'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c458812483cc27dd4a5f'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de@gmail.com'
  },
  {
    _id: ObjectId('67c6c47f812483cc27dd4a60'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de@gmail.com'
  },
  {
    _id: ObjectId('67c6c4a2812483cc27dd4a61'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c4db812483cc27dd4a62'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
```



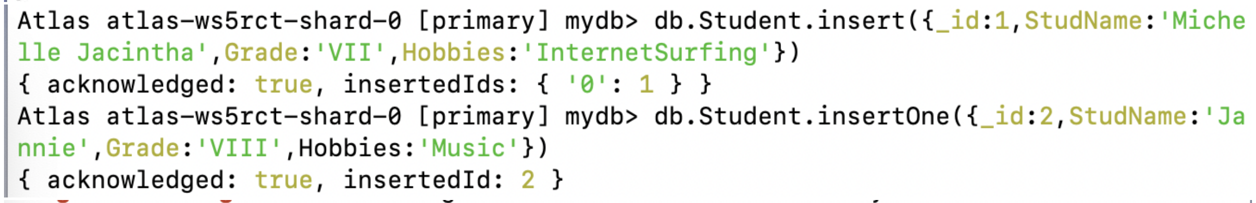
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.find()
[
  {
    _id: ObjectId('67c6c3c3812483cc27dd4a5d'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c3d8812483cc27dd4a5e'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c458812483cc27dd4a5f'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de@gmail.com'
  },
  {
    _id: ObjectId('67c6c47f812483cc27dd4a60'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de@gmail.com'
  },
  {
    _id: ObjectId('67c6c4a2812483cc27dd4a61'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c4db812483cc27dd4a62'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'abhinav@gmail.com'
  },
```



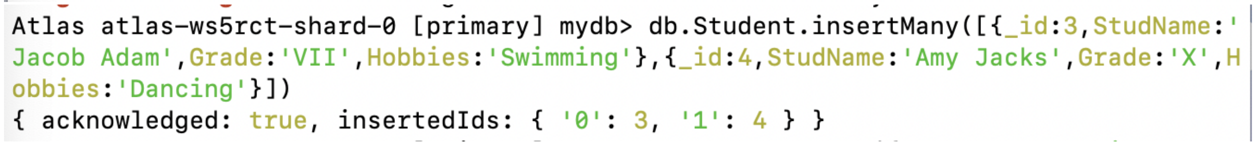
```
  {
    _id: ObjectId('67c6c616812483cc27dd4a63'),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    cont: 2276,
    email: 'rea.de9@gmail.com'
  },
  {
    _id: 1,
    StudName: 'Michelle Jacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 2, StudName: 'Jannie', Grade: 'VIII', Hobbies: 'Music' },
  { _id: 3, StudName: 'Jacob Adam', Grade: 'VII', Hobbies: 'Swimming' },
  {
    _id: 4,
    StudName: 'Amy Jacks',
    Grade: 'X',
    Hobbies: 'Dancing',
    Location: 'Network'
  },
  { _id: 6, StudName: 'Aryan David', Grade: 'VII', Hobbies: 'Skating' }
]
```



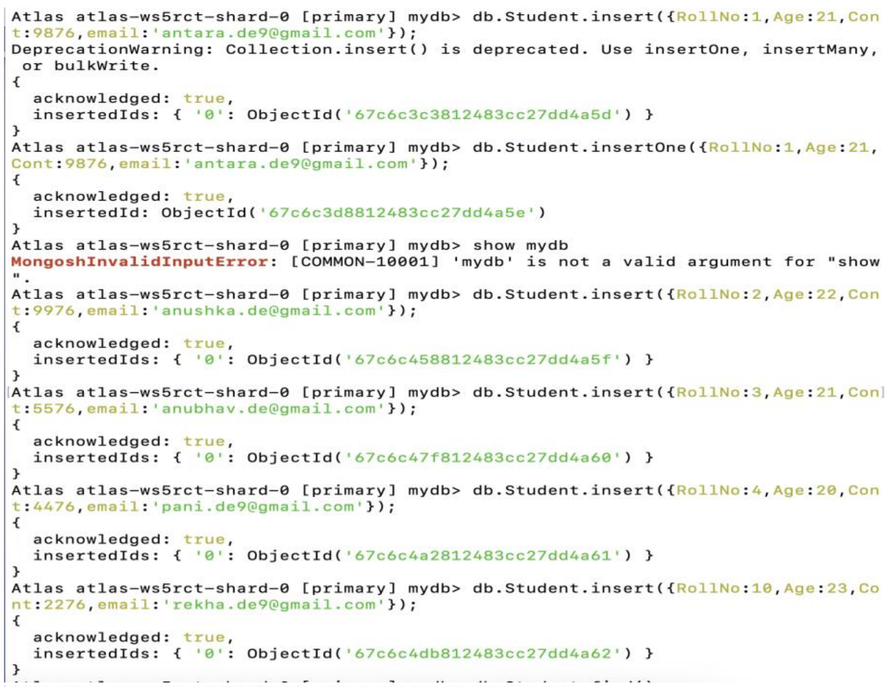
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.insert({cust_id:1,balance:2
00,type:'S'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c71f812483cc27dd4a64') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.insert({cust_id:1,balance:1
000,type:'Z'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c739812483cc27dd4a65') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.insert({cust_id:2,balance:1
00,type:'Z'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c74d812483cc27dd4a66') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.insert({cust_id:2,balance:1
000,type:'C'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c75e812483cc27dd4a67') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.insert({cust_id:2,balance:5
00,type:'C'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c76e812483cc27dd4a68') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.insert({cust_id:2,balance:5
0,type:'S'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c781812483cc27dd4a69') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Customers.insert({cust_id:3,balance:5
00,type:'Z'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c795812483cc27dd4a6a') }
}
```



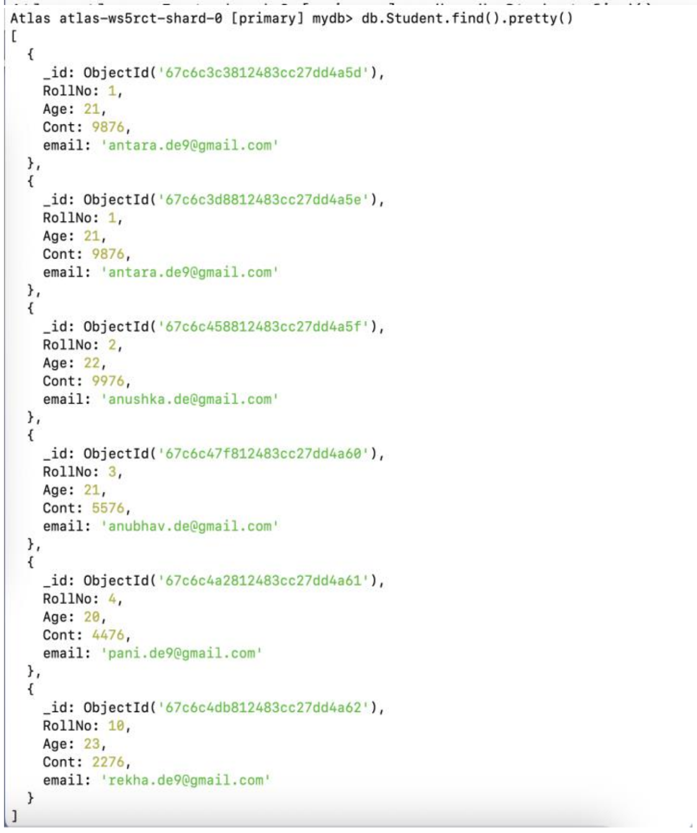
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insert({_id:1,StudName:'Miche
lle Jacintha',Grade:'VII',Hobbies:'InternetSurfing'})
{ acknowledged: true, insertedIds: { '0': 1 } }
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insertOne({_id:2,StudName:'Ja
nnie',Grade:'VIII',Hobbies:'Music'})
{ acknowledged: true, insertedId: 2 }
```



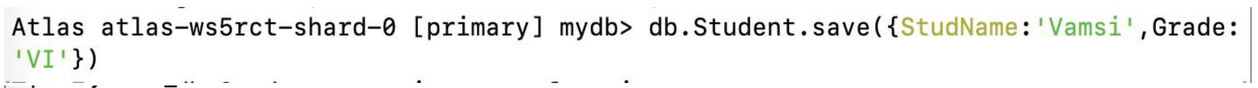
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insertMany([{_id:3,StudName:'
Jacob Adam',Grade:'VII',Hobbies:'Swimming'},{_id:4,StudName:'Amy Jacks',Grade:'X',H
obbies:'Dancing'}])
{ acknowledged: true, insertedIds: { '0': 3, '1': 4 } }
```



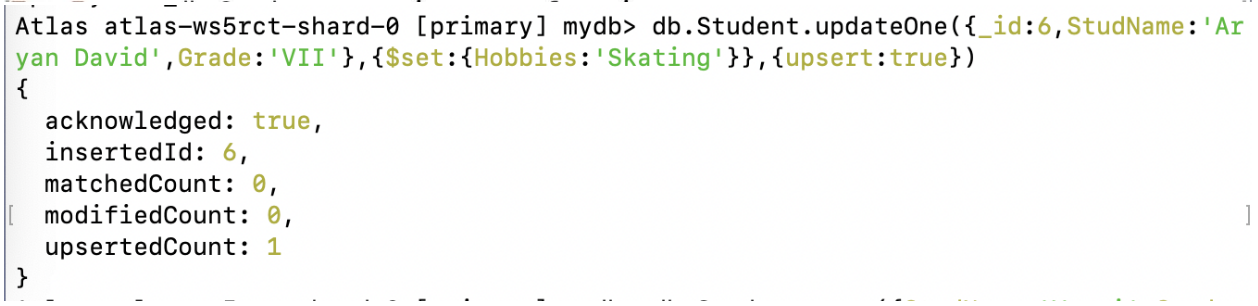
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insert({RollNo:1,Age:21,Con
t:9876,email:'antara.de9@gmail.com'});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany,
 or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c3c3812483cc27dd4a5d') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insertOne({RollNo:1,Age:21,
Cont:9876,email:'antara.de9@gmail.com'});
{
  acknowledged: true,
  insertedId: ObjectId('67c6c3d8812483cc27dd4a5e')
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> show mydb
MongoshInvalidInputError: [COMMON-10001] 'mydb' is not a valid argument for "show
".
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insert({RollNo:2,Age:22,Con
t:9976,email:'anushka.de@gmail.com'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c458812483cc27dd4a5f') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insert({RollNo:3,Age:21,Con
t:5576,email:'anubhav.de@gmail.com'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c47f812483cc27dd4a60') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insert({RollNo:4,Age:20,Con
t:4476,email:'pani.de9@gmail.com'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c4a2812483cc27dd4a61') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insert({RollNo:10,Age:23,Co
nt:2276,email:'rekha.de9@gmail.com'});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c4db812483cc27dd4a62') }
}
```



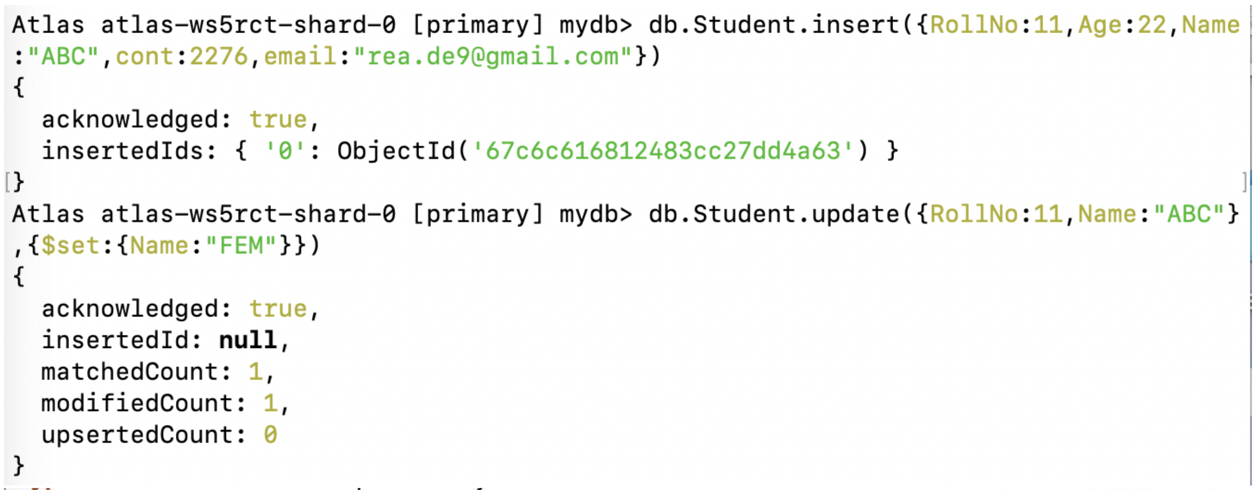
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.find().pretty()
[
  {
    _id: ObjectId('67c6c3c3812483cc27dd4a5d'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c3d8812483cc27dd4a5e'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c458812483cc27dd4a5f'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de@gmail.com'
  },
  {
    _id: ObjectId('67c6c47f812483cc27dd4a60'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de@gmail.com'
  },
  {
    _id: ObjectId('67c6c4a2812483cc27dd4a61'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c4db812483cc27dd4a62'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
```



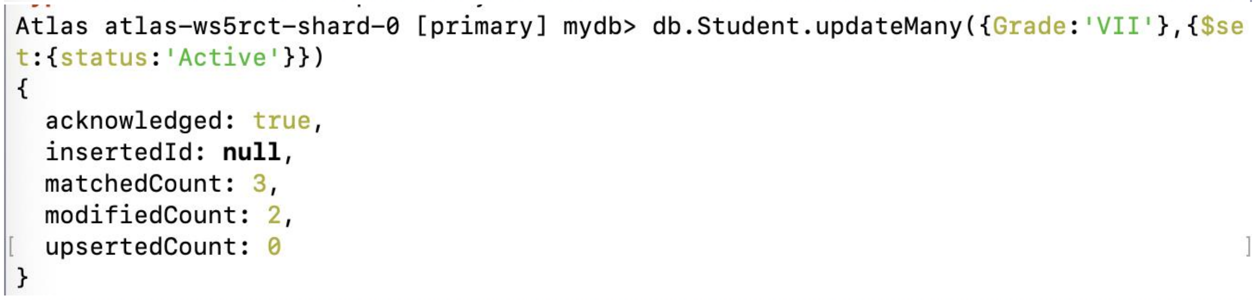
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.save({StudName:'Vamsi',Grade:
'VI'})
```



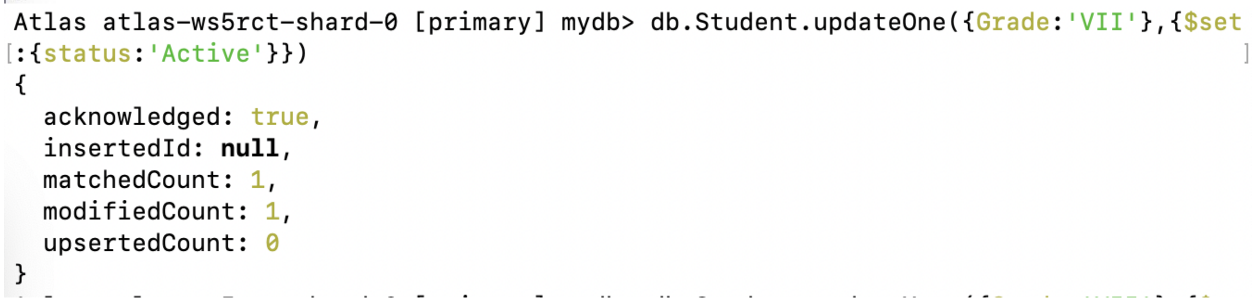
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.updateOne({_id:6,StudName:'Ar
yan David',Grade:'VII'},{$set:{Hobbies:'Skating'}},{upsert:true})
{
  acknowledged: true,
  insertedId: 6,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
```



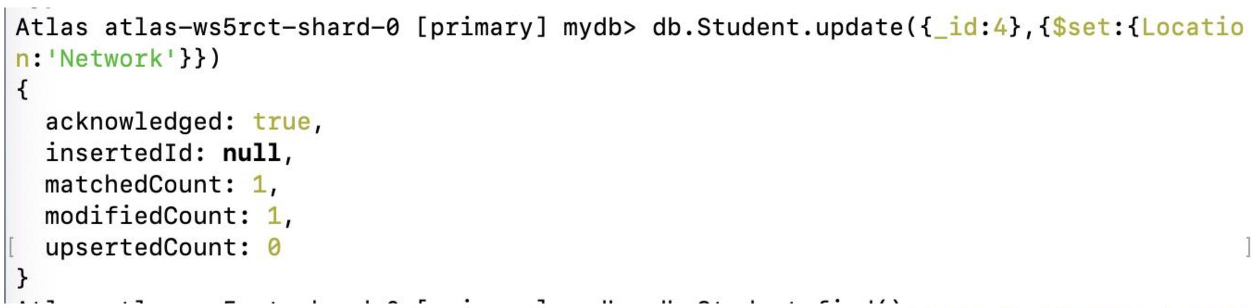
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.insert({RollNo:11,Age:22,Name
:"ABC",cont:2276,email:"rea.de9@gmail.com"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67c6c616812483cc27dd4a63') }
}
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.update({RollNo:11,Name:"ABC"}
,{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



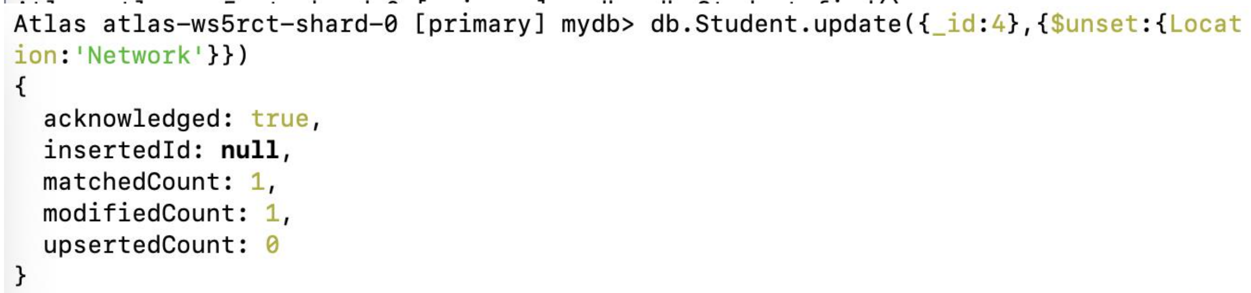
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.updateMany({Grade:'VII'},{$se
t:{status:'Active'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 2,
  upsertedCount: 0
}
```



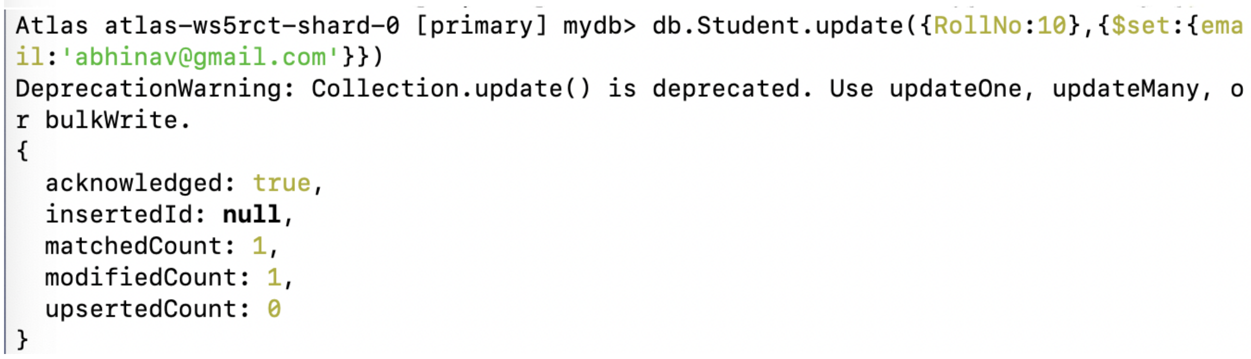
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.updateOne({Grade:'VII'},{$set
:{status:'Active'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.update({_id:4},{$set:{Locatio
n:'Network'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



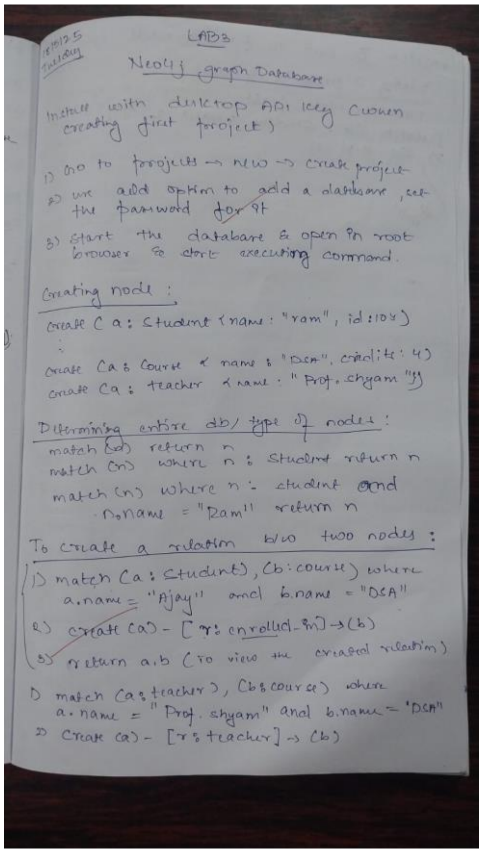
```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.update({_id:4},{$unset:{Locat
ion:'Network'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



```
Atlas atlas-ws5rct-shard-0 [primary] mydb> db.Student.update({RollNo:10},{$set:{ema
il:'abhinav@gmail.com'}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, o
r bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Experiment – 3

Neo4j



18/3/25
Tuesday

LAB3

Neo4j graph Database

Install with desktop API key (when creating first project)

1) Go to projects → new → create project
2) use add option to add a database, set the password for it
3) Start the database & open in root browser to start executing command.

Creating node :

create (a : Student {name : "ram", id :10})

create (a : Course {name : "DSA", credits : 4)
create (a : teacher {name : " Prof. shyam "})

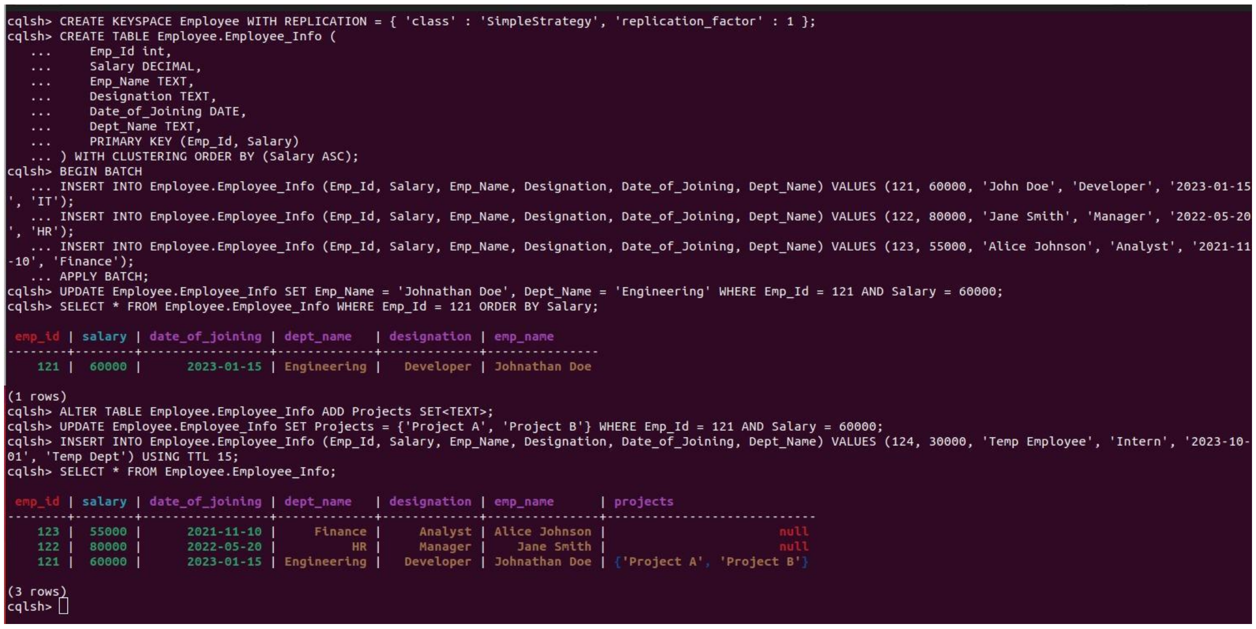
Determining entire db/ type of nodes :

match (n) return n
match (n) where n : Student return n
match (n) where n : student and n.name = "Ram" return n

To create a relation b/w two nodes :

1) match (a : Student), (b : course) where a.name = "Ajay" and b.name = "DSA"
2) create (a) - [r : enrolled-in] → (b)
3) return a,b (to view the created relation)

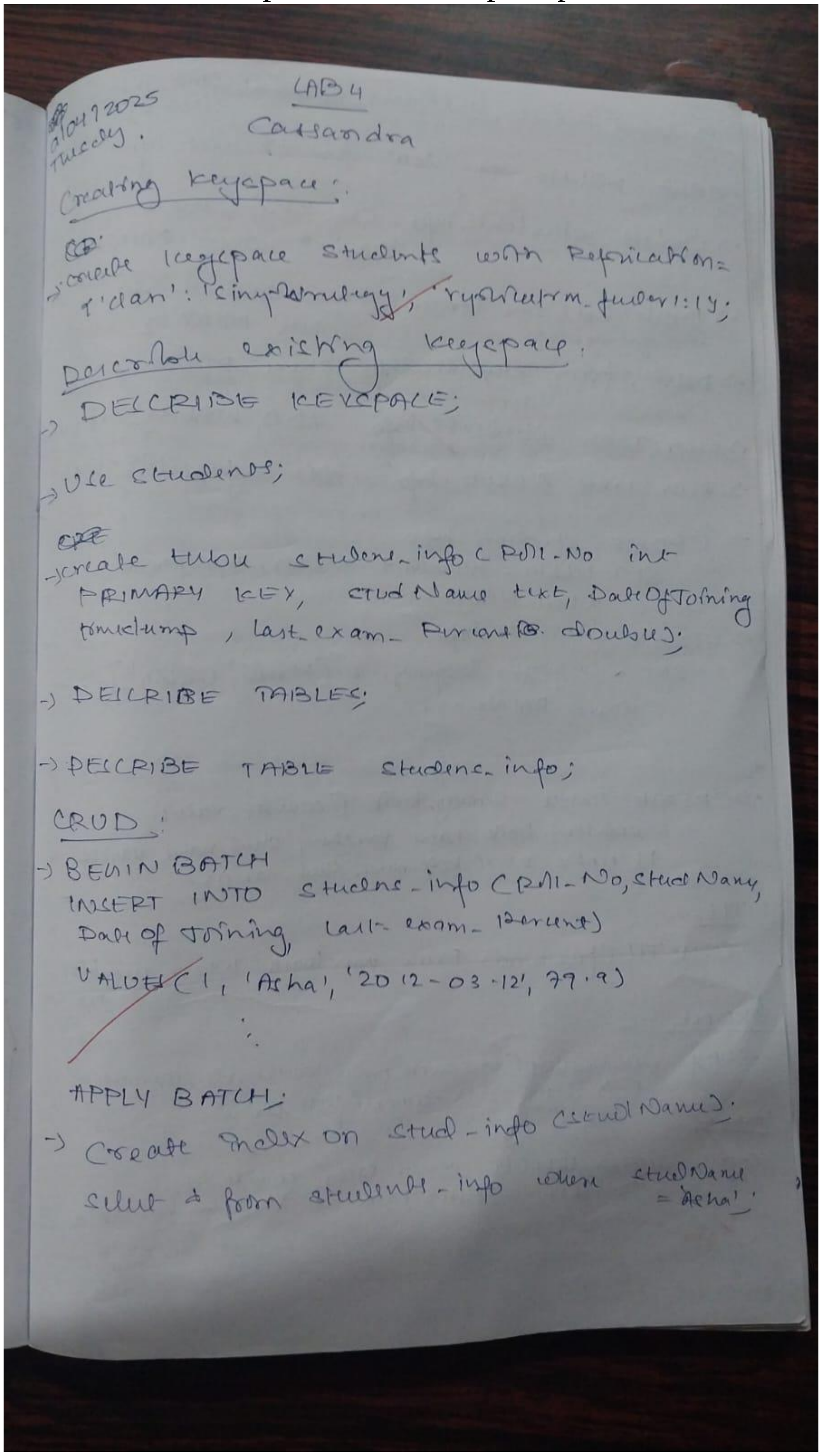
1) match (a : teacher), (b : course) where a. name = " Prof. shyam" and b.name = "DSA"
2) Create (a) - [r : teacher] → (b)



```
cqlsh> CREATE KEYSPACE Employee WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
cqlsh> CREATE TABLE Employee.Employee_Info (
   ...     Emp_Id int,
   ...     Salary DECIMAL,
   ...     Emp_Name TEXT,
   ...     Designation TEXT,
   ...     Date_of_Joining DATE,
   ...     Dept_Name TEXT,
   ...     PRIMARY KEY (Emp_Id, Salary)
   ... ) WITH CLUSTERING ORDER BY (Salary ASC);
cqlsh> BEGIN BATCH
   ... INSERT INTO Employee.Employee_Info (Emp_Id, Salary, Emp_Name, Designation, Date_of_Joining, Dept_Name) VALUES (121, 60000, 'John Doe', 'Developer', '2023-01-15
', 'IT');
   ... INSERT INTO Employee.Employee_Info (Emp_Id, Salary, Emp_Name, Designation, Date_of_Joining, Dept_Name) VALUES (122, 80000, 'Jane Smith', 'Manager', '2022-05-20
', 'HR');
   ... INSERT INTO Employee.Employee_Info (Emp_Id, Salary, Emp_Name, Designation, Date_of_Joining, Dept_Name) VALUES (123, 55000, 'Alice Johnson', 'Analyst', '2021-11
-10', 'Finance');
   ... APPLY BATCH;
cqlsh> UPDATE Employee.Employee_Info SET Emp_Name = 'Johnathan Doe', Dept_Name = 'Engineering' WHERE Emp_Id = 121 AND Salary = 60000;
cqlsh> SELECT * FROM Employee.Employee_Info WHERE Emp_Id = 121 ORDER BY Salary;

 emp_id | salary | date_of_joining | dept_name   | designation | emp_name
--------+--------+-----------------+-------------+-------------+---------------
    121 |  60000 |      2023-01-15 | Engineering |   Developer | Johnathan Doe

(1 rows)
cqlsh> ALTER TABLE Employee.Employee_Info ADD Projects SET<TEXT>;
cqlsh> UPDATE Employee.Employee_Info SET Projects = {'Project A', 'Project B'} WHERE Emp_Id = 121 AND Salary = 60000;
cqlsh> INSERT INTO Employee.Employee_Info (Emp_Id, Salary, Emp_Name, Designation, Date_of_Joining, Dept_Name) VALUES (124, 30000, 'Temp Employee', 'Intern', '2023-10-
01', 'Temp Dept') USING TTL 15;
cqlsh> SELECT * FROM Employee.Employee_Info;

 emp_id | salary | date_of_joining | dept_name   | designation | emp_name      | projects
--------+--------+-----------------+-------------+-------------+---------------+----------------------------
    123 |  55000 |      2021-11-10 |     Finance |     Analyst | Alice Johnson |                       null
    122 |  80000 |      2022-05-20 |          HR |     Manager |    Jane Smith |                       null
    121 |  60000 |      2023-01-15 | Engineering |   Developer | Johnathan Doe | {'Project A', 'Project B'}

(3 rows)
cqlsh> 
```

Experiment – 4

Explore Cassandra prompts



9/04/2025
Tuesday.

LAB 4

Cassandra

Creating keyspace:

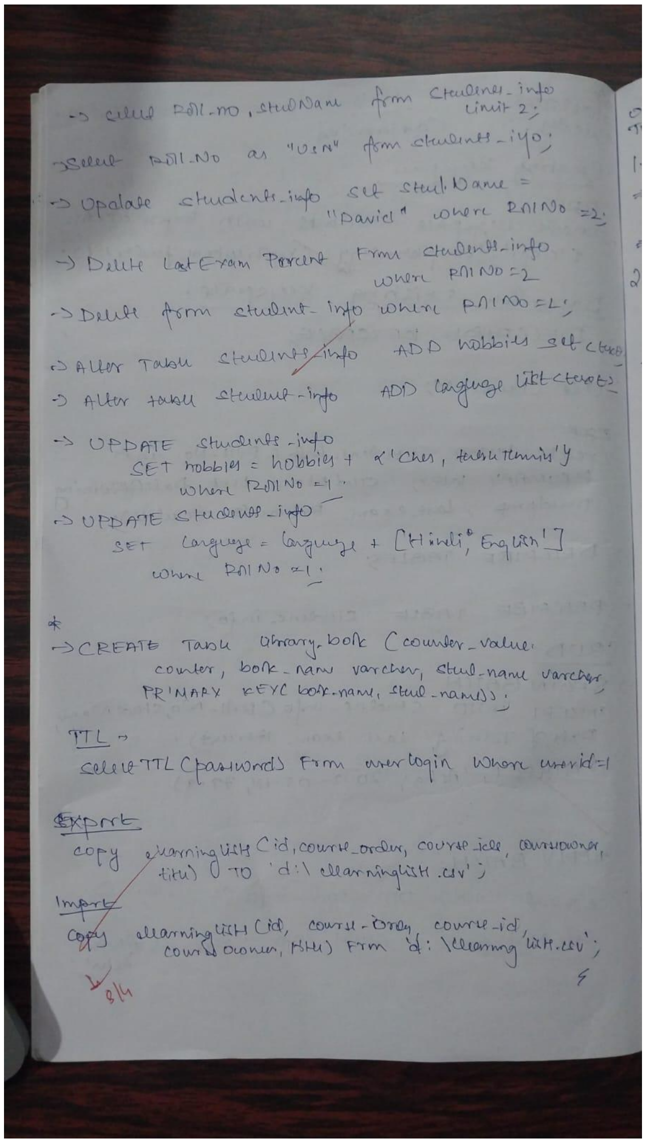
```
-> create keyspace students with Replication=
   {'class': 'SimpleStrategy', 'replication_factor':1};
```

Describe existing keyspace:

```
-> DESCRIBE KEYSPACE;

-> Use students;

-> create table student_info (Roll_No int
   PRIMARY KEY, StudName text, DateOfJoining
   timestamp, last_exam_Percent double);

-> DESCRIBE TABLES;

-> DESCRIBE TABLE student_info;
```

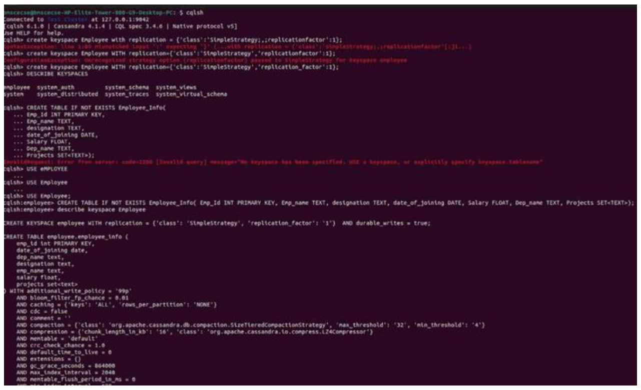
CRUD:

```
-> BEGIN BATCH
   INSERT INTO student_info (Roll_No, StudName,
   DateOfJoining, last_exam_Percent)
   VALUES (1, 'Asha', '2012-03-12', 79.9)
   :
   APPLY BATCH;
-> Create index on stud_info (StudName);
   select * from students_info where StudName
                                      = 'Asha';
```



```
-> select Roll-no, studName from students_info
                                  limit 2;
-> select Roll_No as "USN" from students_info;
-> Update students_info set studName =
                  "David" where RollNo =2;
-> Delete LastExam Percent From students_info
                         where RollNo=2
-> Delete from student_info where RollNo=1;
-> Alter Table students_info ADD hobbies set<text>
-> Alter table student_info ADD language list<text>
-> UPDATE students_info
      SET hobbies = hobbies + {'Chess, table tennis'}
         where RollNo =1;
-> UPDATE students_info
      SET language = language + ['Hindi', English']
         where RollNo =1;

*
-> CREATE Table library_book (counter_value
      counter, book_name varchar, stud_name varchar,
      PRIMARY KEY (book_name, stud_name));

TTL ->
  select TTL (password) From userlogin where userid=1

Export
  copy elearninglists (id, course_order, course_id, courseowner,
        title) TO 'd:\elearninglist.csv';
Import
  copy elearninglists (id, course_order, course_id,
        courseowner, title) From 'd:\elearninglist.csv';
                                                  ~~~
   8/4
```

Code Outputs:

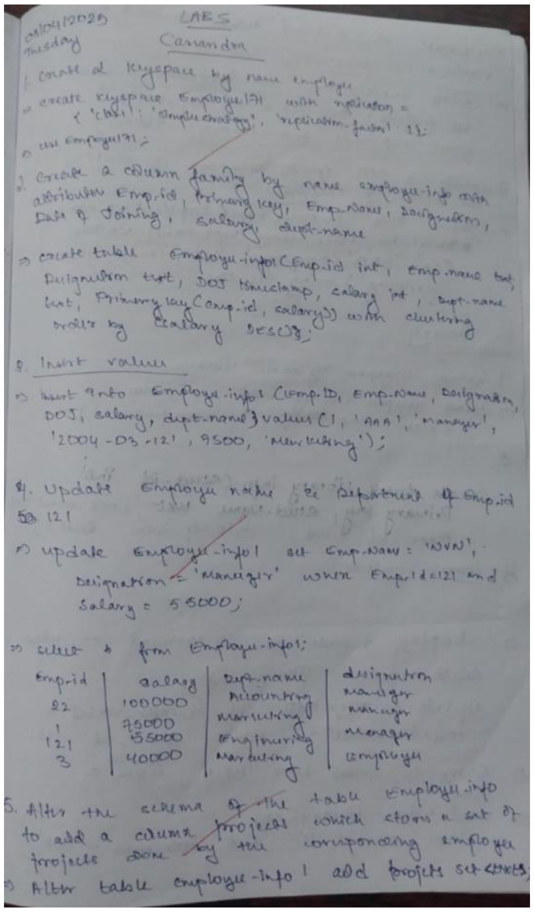


```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> create keyspace Employee with replication = {'class':'SimpleStrategy;,;replicationfactor':1};
SyntaxException: line 1:80 mismatched input ':' expecting '}' (...with replication = {'class':'SimpleStrategy;,;replicationfactor'[:]1...)
cqlsh> create keyspace Employee WITH replication={'class':'SimpleStrategy','replicationfactor':1};
ConfigurationException: Unrecognized strategy option {replicationfactor} passed to SimpleStrategy for keyspace employee
cqlsh> create keyspace Employee WITH replication={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES

employee  system_auth         system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

cqlsh> CREATE TABLE IF NOT EXISTS Employee_Info(
   ... Emp_Id INT PRIMARY KEY,
   ... Emp_name TEXT,
   ... designation TEXT,
   ... date_of_joining DATE,
   ... Salary FLOAT,
   ... Dep_name TEXT,
   ... Projects SET<TEXT>);
InvalidRequest: Error from server: code=2200 [Invalid query] message="No keyspace has been specified. USE a keyspace, or explicitly specify keyspace.tablename"
cqlsh> USE eMPLOYEE
   ...
cqlsh> USE Employee
   ...
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_Info( Emp_Id INT PRIMARY KEY, Emp_name TEXT, designation TEXT, date_of_joining DATE, Salary FLOAT, Dep_name TEXT, Projects SET<TEXT>);
cqlsh:employee> describe keyspace Employee

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

CREATE TABLE employee.employee_info (
    emp_id int PRIMARY KEY,
    date_of_joining date,
    dep_name text,
    designation text,
    emp_name text,
    salary float,
    projects set<text>
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
```

Experiment – 5

Perform the following DB operations using Cassandra.

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



08/04/2025
Tuesday

LAB 5

Cassandra

1. create a keyspace by name employee
=> create keyspace Employee171 with replication = { 'class': 'SimpleStrategy', 'replication_factor': 1};
=> use Employee171;

2. Create a column family by name employee-info with attributes Emp-id, Primary key, Emp-Name, Designation, Date of Joining, salary, dept-name

=> create table Employee-info1 (Emp-id int, emp-name text, Designation text, DOJ timestamp, salary int, dept-name text, Primary key (emp-id, salary)) with clustering order by (salary DESC);

3. Insert values
=> insert into Employee-info1 (Emp-ID, Emp-Name, Designation, DOJ, salary, dept-name) values (1, 'AAA', 'manager', '2004-03-12', 9500, 'Marketing');

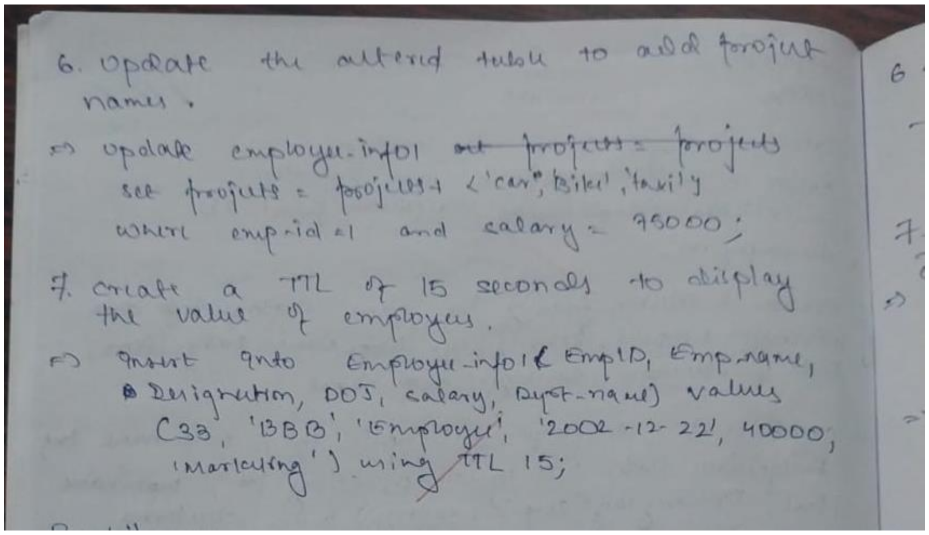
4. Update Employee name & Department of Emp-id 121

=> update Employee-info1 set Emp-Name = 'NVN', Designation = 'Manager' where Emp-id=121 and Salary = 55000;

=> select * from Employee-info1;

```
Emp-id | salary | Dept-name   | designation
  22   | 100000 | Accounting  | Manager
   1   | 75000  | Marketing   | Manager
 121   | 55000  | Engineering | Manager
   3   | 40000  | Marketing   | Employee
```

5. Alter the schema of the table Employee-info to add a column projects which stores a set of projects done by the corresponding employee
=> Alter table employee-info1 add projects set<text>;



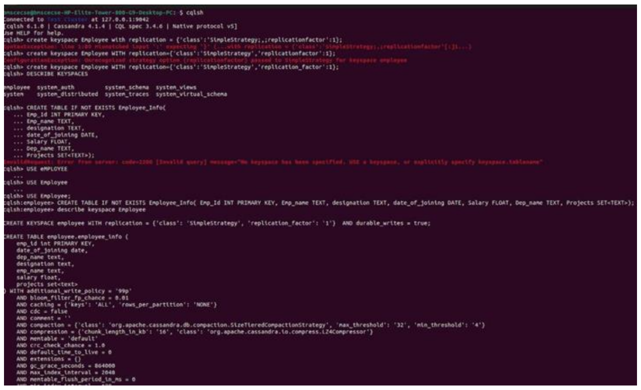
6. Update the altered table to add project names.

=> update employee-info1 ~~set projects = projects~~
set projects = projects + {'car', 'bike', 'taxi'}
where emp-id =1 and salary = 75000;

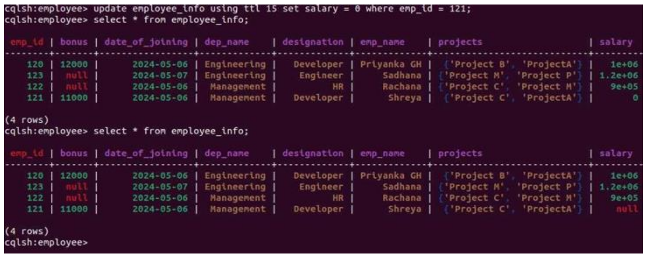
7. Create a TTL of 15 seconds to display the value of employees.

=> Insert into Employee-info1 (EmpID, Emp-name, Designation, DOJ, salary, Dept-name) values (33, 'BBB', 'Employee', '2002-12-22', 40000, 'Marketing') using TTL 15;

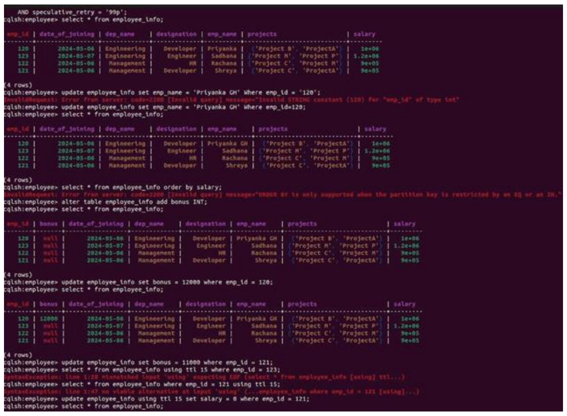
Codes Output:



```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> create keyspace Employee with replication = {'class':'SimpleStrategy;,;replicationfactor':1};
SyntaxException: line 1:80 mismatched input ':' expecting '}' (...with replication = {'class':'SimpleStrategy;,;replicationfactor'[:]1...)
cqlsh> create keyspace Employee WITH replication={'class':'SimpleStrategy','replicationfactor':1};
ConfigurationException: Unrecognized strategy option {replicationfactor} passed to SimpleStrategy for keyspace employee
cqlsh> create keyspace Employee WITH replication={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES

employee  system_auth         system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

cqlsh> CREATE TABLE IF NOT EXISTS Employee_Info(
   ... Emp_Id INT PRIMARY KEY,
   ... Emp_name TEXT,
   ... designation TEXT,
   ... date_of_joining DATE,
   ... Salary FLOAT,
   ... Dep_name TEXT,
   ... Projects SET<TEXT>);
InvalidRequest: Error from server: code=2200 [Invalid query] message="No keyspace has been specified. USE a keyspace, or explicitly specify keyspace.tablename"
cqlsh> USE eMPLOYEE
   ...
cqlsh> USE Employee
   ...
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE IF NOT EXISTS Employee_Info( Emp_Id INT PRIMARY KEY, Emp_name TEXT, designation TEXT, date_of_joining DATE, Salary FLOAT, Dep_name TEXT, Projects SET<TEXT>);
cqlsh:employee> describe keyspace Employee

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

CREATE TABLE employee.employee_info (
    emp_id int PRIMARY KEY,
    date_of_joining date,
    dep_name text,
    designation text,
    emp_name text,
    salary float,
    projects set<text>
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
```



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

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

(4 rows)
cqlsh:employee> select * from employee_info;

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

(4 rows)
cqlsh:employee>
```



```
    AND speculative_retry = '99p';
cqlsh:employee> select * from employee_info;

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

(4 rows)
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id = '120';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid STRING constant (120) for "emp_id" of type int"
cqlsh:employee> update employee_info set emp_name = 'Priyanka GH' Where emp_id=120;
cqlsh:employee> select * from employee_info;

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

(4 rows)
cqlsh:employee> select * from employee_info order by salary;
InvalidRequest: Error from server: code=2200 [Invalid query] message="ORDER BY is only supported when the partition key is restricted by an EQ or an IN."
cqlsh:employee> alter table employee_info add bonus INT;
cqlsh:employee> select * from employee_info;

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

(4 rows)
cqlsh:employee> update employee_info set bonus = 12000 where emp_id = 120;
cqlsh:employee> select * from employee_info;

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

(4 rows)
cqlsh:employee> update employee_info set bonus = 11000 where emp_id = 121;
cqlsh:employee> select * from employee_info using ttl 15 where emp_id = 123;
SyntaxException: line 1:28 mismatched input 'using' expecting EOF (select * from employee_info [using] ttl...)
cqlsh:employee> select * from employee_info where emp_id = 121 using ttl 15;
SyntaxException: line 1:47 no viable alternative at input 'using' (...employee_info where emp_id = 121 [using]...)
cqlsh:employee> update employee_info using ttl 15 set salary = 0 where emp_id = 121;
cqlsh:employee> select * from employee_info;
```

Perform the following DB operations using Cassandra:

● Create a keyspace by name Library

● Create a column family by name Library-Info with attributes Stud_Id Primary Key, Counter_value of type Counter, Stud_Name, Book-Name, Book-Id, Date_of_issue

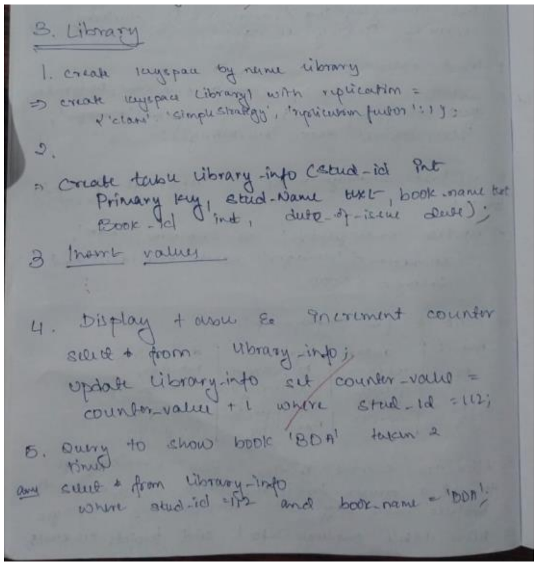
● Insert the values into the table in batch

● Display the details of the table created and increase the value of the counter

● Write a query to show that a student with id 112 has taken a book "BDA" 2 times.

● Export the created column to a csv file

● Import a given csv dataset from local file system into Cassandra column family.



3. Library

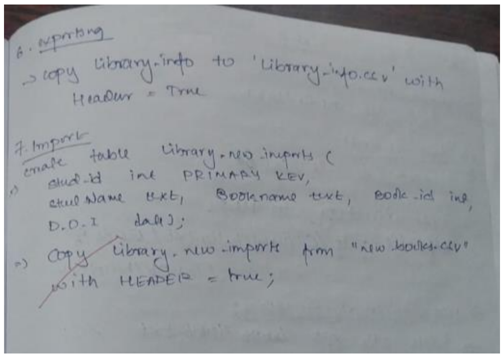
1. create keyspace by name library
=> create keyspace Library1 with replication =
{'class' : 'simpleStrategy', 'replication factor' : 1 };

2.
=> Create table library-info (stud-id int Primary key, stud-Name text, book-name text, Book-Id int, date-of-issue date);

3 Insert values

4. Display + also & increment counter
select * from library-info;
update library-info set counter-value = counter-value + 1 where stud-id = 112;

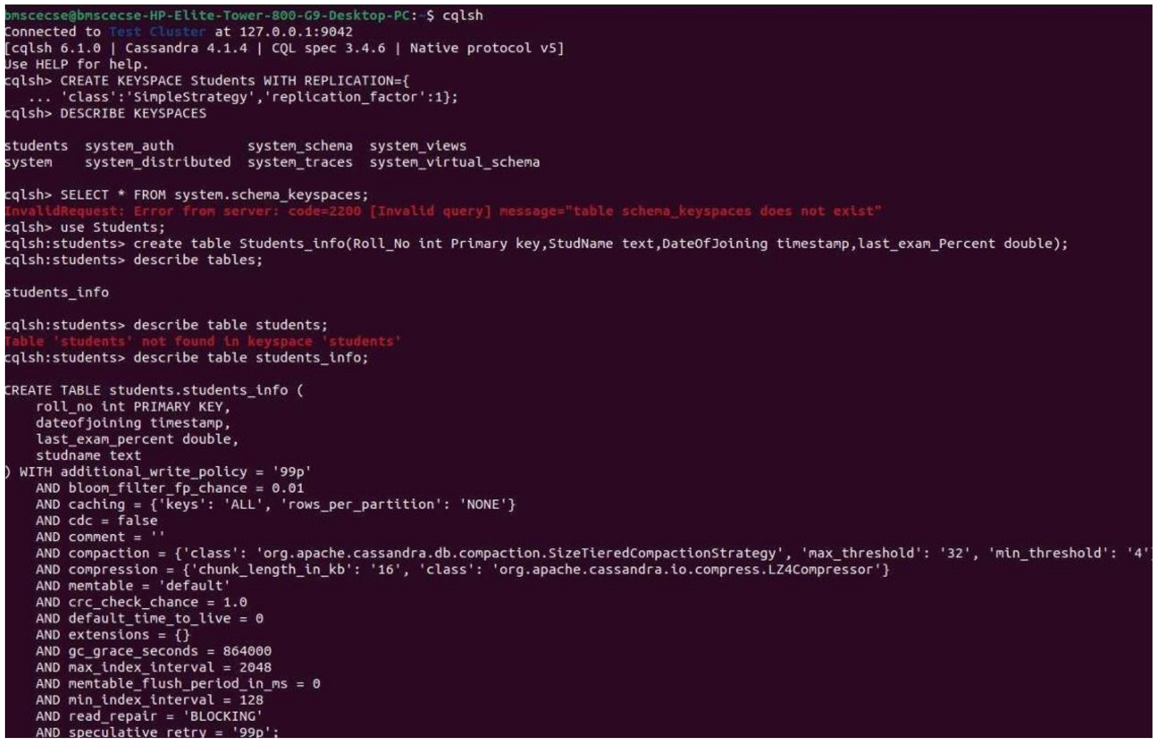
5. Query to show book 'BDA' taken 2 times
ans select * from library-info
where stud-id = 112 and book-name = 'BDA';



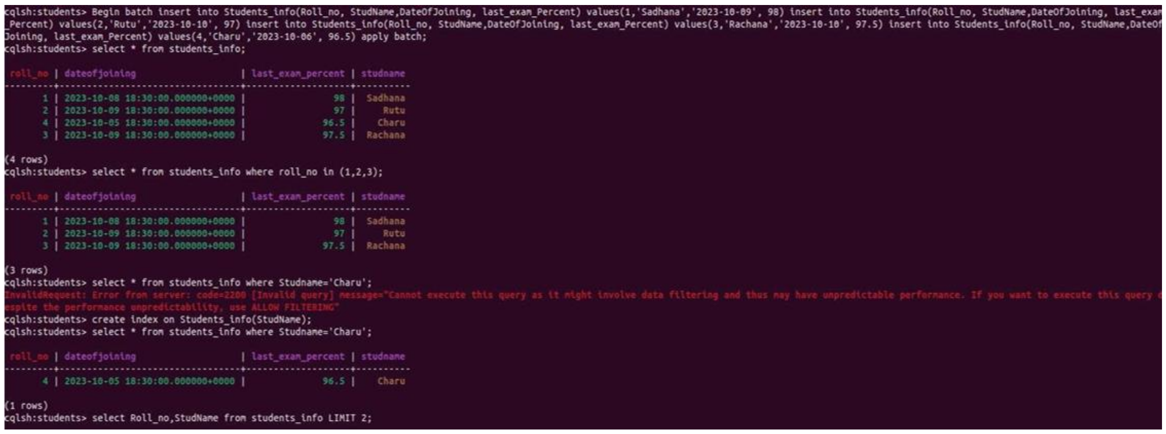
6. exporting
=> copy library-info to 'Library-info.csv' with Header = True

7. Import
create table library.new imports (
=> stud-id int PRIMARY KEY,
stud Name text, Bookname text, Book-id int,
D.O.I date);
=> copy library.new-imports from "new-books.csv"
with HEADER = true;

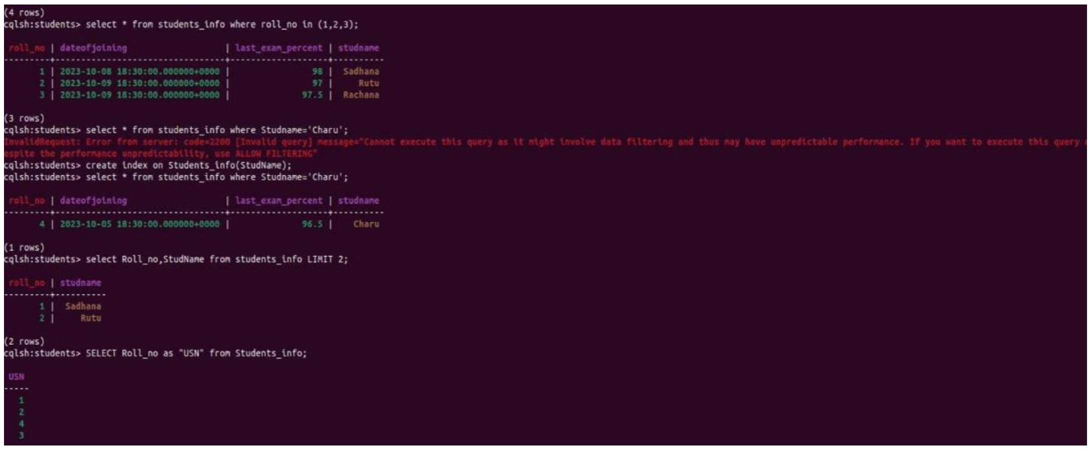
Codes Output:



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

students  system_auth         system_schema  system_views
system    system_distributed  system_traces  system_virtual_schema

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

students_info

cqlsh:students> describe table students;
Table 'students' not found in keyspace 'students'
cqlsh:students> describe table students_info;

CREATE TABLE students.students_info (
    roll_no int PRIMARY KEY,
    dateofjoining timestamp,
    last_exam_percent double,
    studname text
) WITH additional_write_policy = '99p'
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';
```



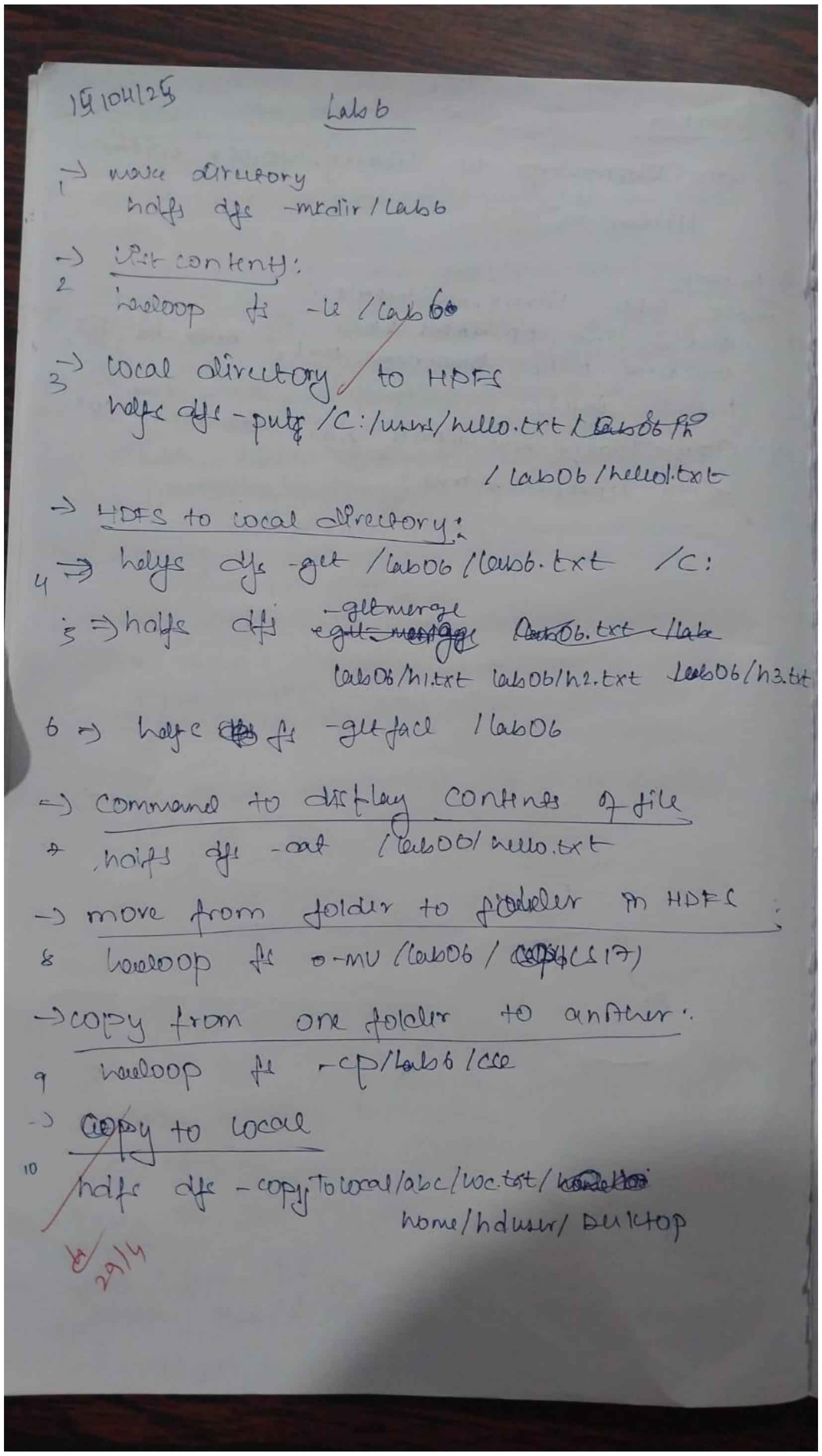
```
cqlsh:students> Begin batch insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent) values(1,'Sadhana','2023-10-09', 98) insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam
_Percent) values(2,'Rutu','2023-10-10', 97) insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent) values(3,'Rachana','2023-10-10', 97.5) insert into Students_info(Roll_no, StudName,DateOf
Joining, last_exam_Percent) values(4,'Charu','2023-10-06', 96.5) apply batch;
cqlsh:students> select * from students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2023-10-08 18:30:00.000000+0000 |                98 |  Sadhana
       2 | 2023-10-09 18:30:00.000000+0000 |                97 |     Rutu
       4 | 2023-10-05 18:30:00.000000+0000 |              96.5 |    Charu
       3 | 2023-10-09 18:30:00.000000+0000 |              97.5 |  Rachana

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

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2023-10-08 18:30:00.000000+0000 |                98 |  Sadhana
       2 | 2023-10-09 18:30:00.000000+0000 |                97 |     Rutu
       3 | 2023-10-09 18:30:00.000000+0000 |              97.5 |  Rachana

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

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

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



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

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2023-10-08 18:30:00.000000+0000 |                98 |  Sadhana
       2 | 2023-10-09 18:30:00.000000+0000 |                97 |     Rutu
       3 | 2023-10-09 18:30:00.000000+0000 |              97.5 |  Rachana

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

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

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

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

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

 USN
-----
   1
   2
   4
   3
```

Experiment-6

Execution of HDFS Commands for interaction with Hadoop Environment.



15/04/25

Lab 6

1 → make directory
hdfs dfs -mkdir /Lab6

2 → list contents:
hadoop fs -ls /Lab6

3 → local directory to HDFS
hdfs dfs -put /C:/users/hello.txt ~~/Lab06/h~~
/Lab06/hello1.txt

→ HDFS to local directory:

4 ⇒ hdfs dfs -get /Lab06/Lab6.txt /C:

5 ⇒ hdfs dfs -getmerge ~~-get-merge~~ ~~Lab06.txt~~ /Lab
Lab06/h1.txt Lab06/h2.txt Lab06/h3.txt

6 → hdfs ~~[illegible]~~ fs -getfacl /Lab06

→ Command to display contents of file

7 hdfs dfs -cat /Lab00/hello.txt

→ move from folder to folder in HDFS:

8 hadoop fs -mv /Lab06/ ~~[illegible]~~(S17)

→ copy from one folder to another:

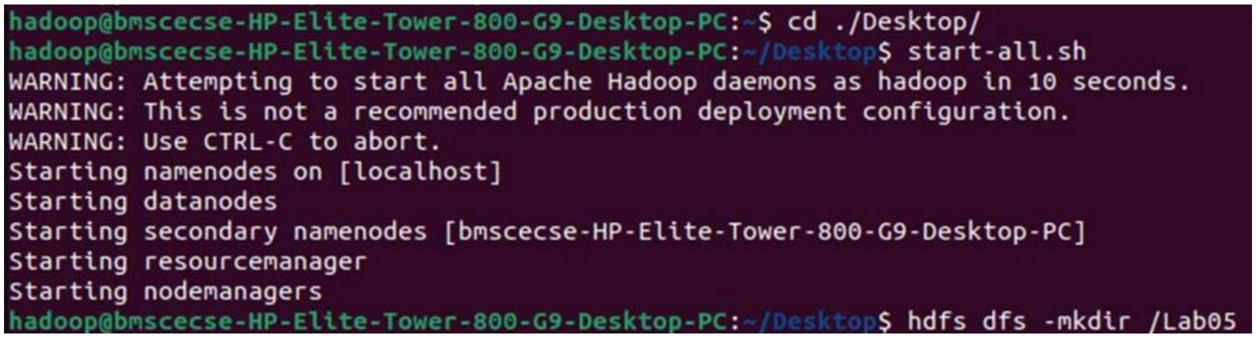
9 hadoop fs -cp /Lab6 /cce

→ copy to local

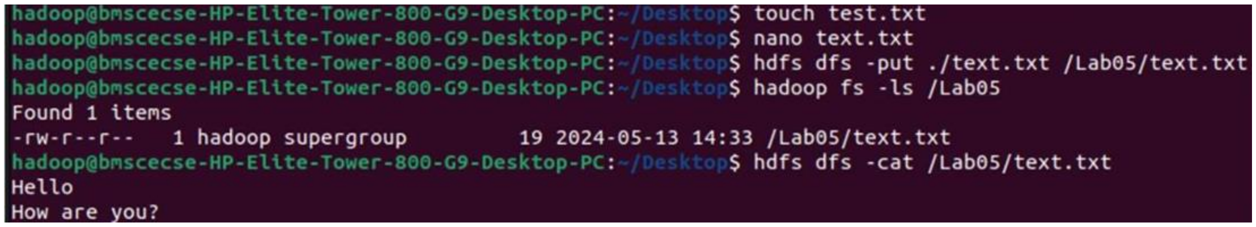
10 hdfs dfs -copyToLocal /abc/wc.txt/ ~~[illegible]~~
home/hduser/Desktop

29/4

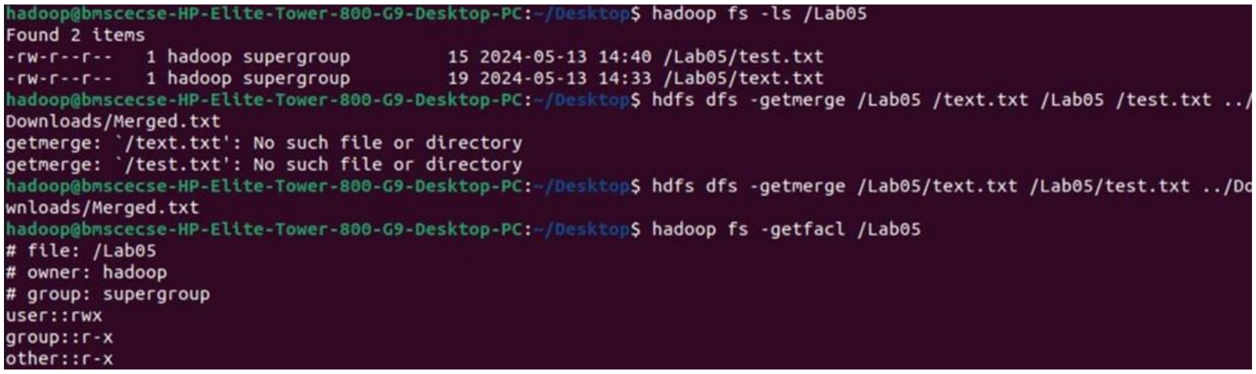
Codes Output:



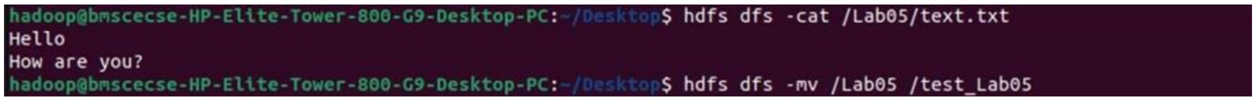
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: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



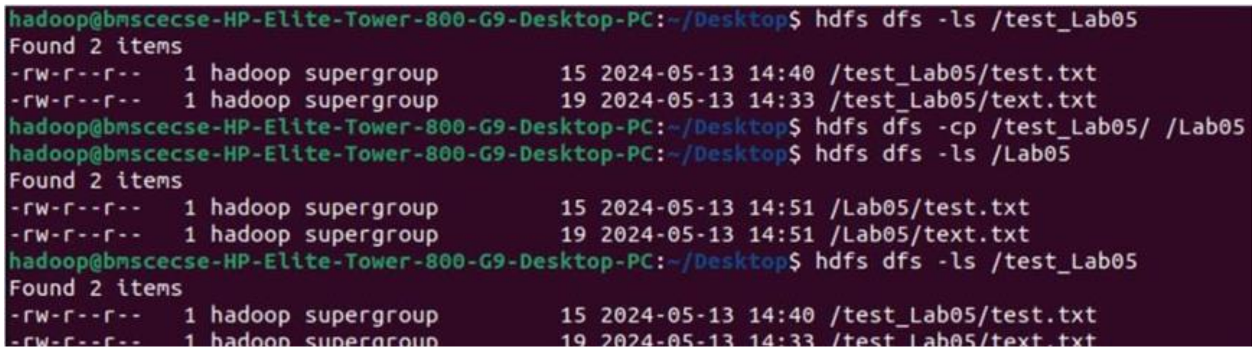
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ touch test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -put ./text.txt /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$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



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



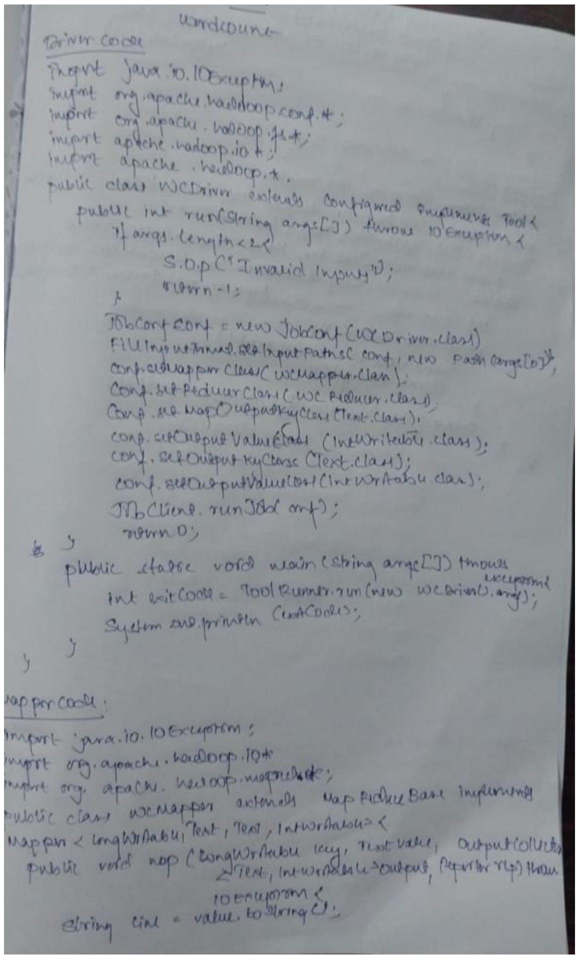
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -cat /Lab05/text.txt
Hello
How are you?
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -mv /Lab05 /test_Lab05
```



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

Experiment-7

Implement Wordcount program on Hadoop framework



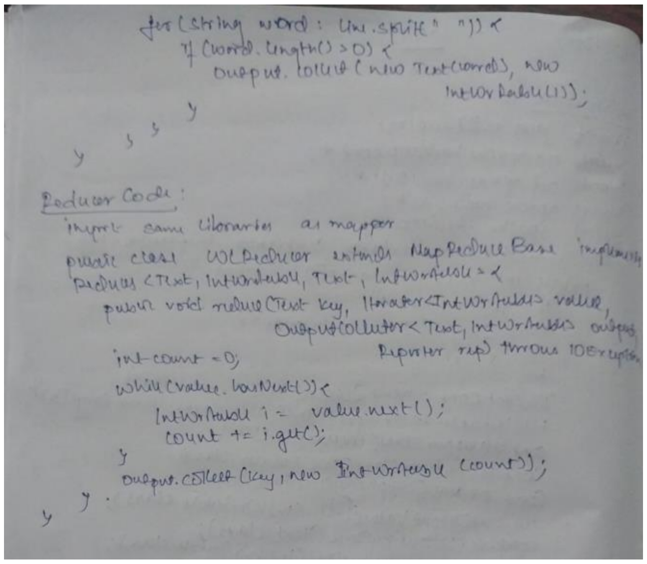
Wordcount

Driver code

```
import java.io.IOException;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.fs.*;
import apache.hadoop.io.*;
import apache.hadoop.*.
public class WCDriver extends Configured implements Tool{
    public int run(String args[]) throws IOException {
        if args.length<2{
            S.O.p("Invalid Inputs");
            return -1;
        }
        JobConf conf = new JobConf(WCDriver.class)
        FileInputFormat.setInputPaths( conf, new Path(args[0]),
        conf.setMapperClass( WCMapper.Class);
        Conf.setReducerClass( WC Reducer.class),
        Conf.setMapOutputKeyClass(Text.Class),
        conf.setOutputValueClass (IntWritable.class);
        conf.setOutputKeyClass (Text.class);
        conf.setOutputValueClass(IntWritable class);
        JobClient.runJob( conf);
        return 0;
    }
    public static void main(String args[]) throws
                                                  exception{
        int exitCode = ToolRunner.run(new WCDriver(),args);
        System.out.println (exitCode);
    }
}
```

Mapper code:

```
import java.io.IOException;
import org.apache.hadoop.io*
import org.apache.hadoop.mapred.*;
public class WCMapper extends MapReduceBase implements
Mapper < LongWritable, Text, Text, IntWritable> {
  public void map (LongWritable key, Text value, OutputCollector
                   <Text, IntWritable> output, Reporter rep) throws
                          IOException {
    String line = value.toString();
```



```
      for(String word : line.split(" ")) {
          if (word.length() > 0) {
              output.collect( new Text(word), new
                                            IntWritable(1));
          }
      }
    }
}
```

Reducer Code:

```
import same libraries as mapper
public class WCReducer extends MapReduceBase implements
Reducer <Text, IntWritable, Text, IntWritable> {
   public void reduce(Text key, Iterator<IntWritable> value,
                      OutputCollector< Text, IntWritable> output,
     int count = 0;                     Reporter rep) throws IOException
     while(value.hasNext()){
         IntWritable i = value.next();
         count += i.get();
     }
     output.collect(key, new IntWritable (count));
   }.
}
```

Mapper:

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

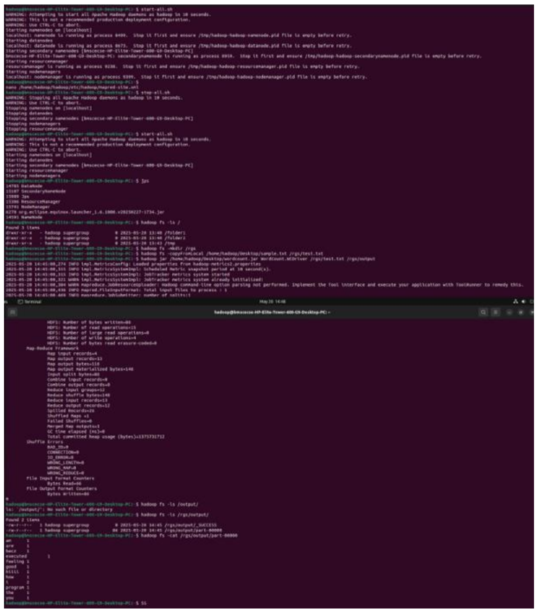
Reducer:

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

Driver:

```
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
public class WCDriver extends Configured implements Tool {
public int run(String args[]) throws IOException
{
if (args.length < 2)
{
System.out.println("Please give valid inputs");
return -1;
}
JobConf conf = new JobConf(WCDriver.class);
FileInputFormat.setInputPaths(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));
conf.setMapperClass(WCMapper.class);
conf.setReducerClass(WCReducer.class);
conf.setMapOutputKeyClass(Text.class);
conf.setMapOutputValueClass(IntWritable.class);
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
JobClient.runJob(conf);
return 0;
}
public static void main(String args[]) throws Exception
{
int exitCode = ToolRunner.run(new WCDriver(), args);
System.out.println(exitCode);
}
}
```

Codes Output:



```
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]
localhost: namenode is running as process 8499.  Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 8673.  Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid 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 8959.  Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
Starting resourcemanager
resourcemanager is running as process 9238.  Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 9399.  Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$
nano /home/hadoop/hadoop/etc/hadoop/mapred-site.xml
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
Stopping nodemanagers
Stopping resourcemanager
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
14785 DataNode
15107 SecondaryNameNode
15989 Jps
15306 ResourceManager
15741 NodeManager
6278 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
14593 NameNode
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2025-05-28 13:40 /folder1
drwxr-xr-x   - hadoop supergroup          0 2025-05-28 13:40 /folder2
drwxr-xr-x   - hadoop supergroup          0 2025-05-28 13:43 /tmp
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mkdir /rgs
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/sample.txt /rgs/test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/wordcount.jar WordCount.WCDriver /rgs/test.txt /rgs/output
2025-05-28 14:45:08,274 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-28 14:45:08,315 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-28 14:45:08,315 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-28 14:45:08,321 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2025-05-28 14:45:08,384 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-28 14:45:08,436 INFO mapred.FileInputFormat: Total input files to process : 1
2025-05-28 14:45:08,469 INFO mapreduce.JobSubmitter: number of splits:1
```

Terminal   May 28 14:48

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC: ~

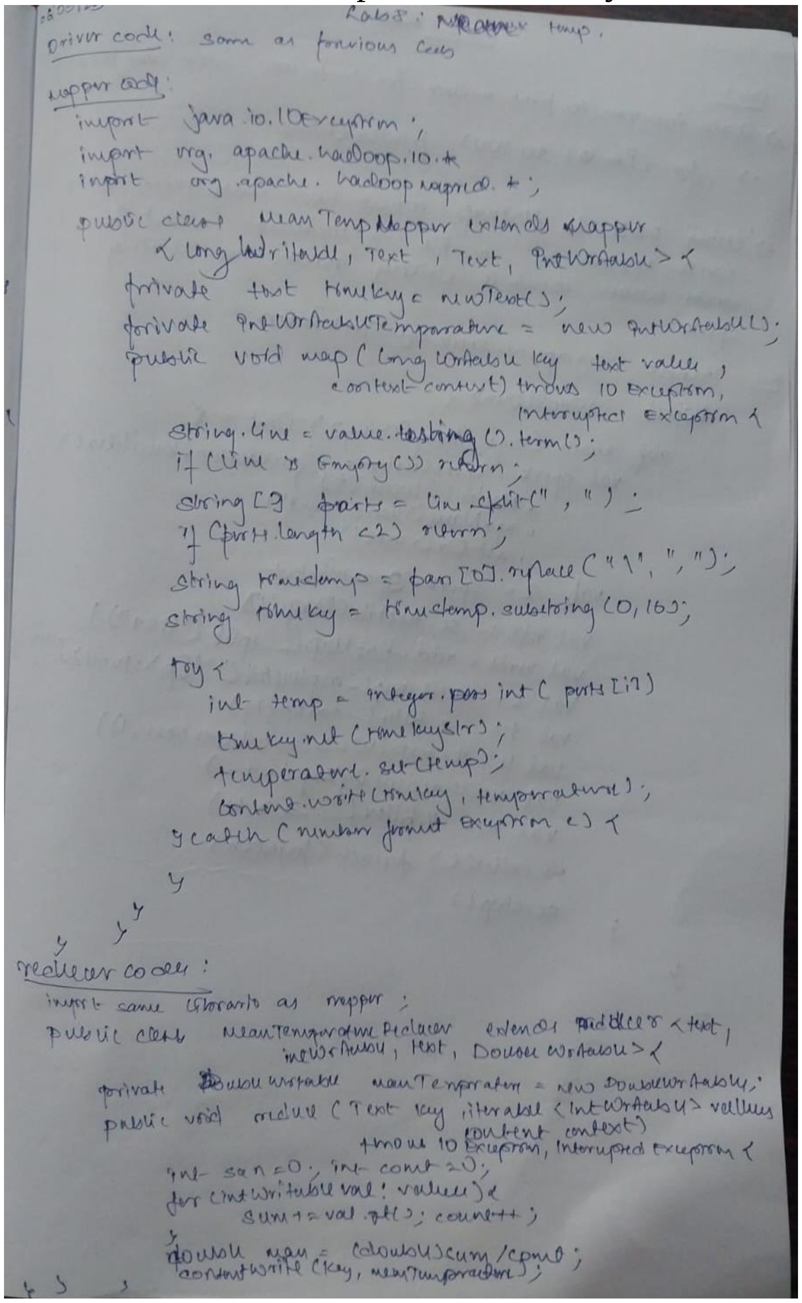
```
                HDFS: Number of bytes written=86
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=4
                Map output records=13
                Map output bytes=118
                Map output materialized bytes=148
                Input split bytes=88
                Combine input records=0
                Combine output records=0
                Reduce input groups=12
                Reduce shuffle bytes=148
                Reduce input records=13
                Reduce output records=12
                Spilled Records=26
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1375731712
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=66
        File Output Format Counters
                Bytes Written=86
0
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /output/
ls: `/output/': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /rgs/output/
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-28 14:45 /rgs/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup         86 2025-05-28 14:45 /rgs/output/part-00000
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cat /rgs/output/part-00000
am      1
are     1
beca    1
executed        1
feeling 1
good    1
hiiii   1
how     1
i       2
program 1
the     1
you     1
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ SS
```

Experiment-8

From the following link extract the weather data:
https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all

Create a Map Reduce program to:

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



Lab 8: Mean max temp.

driver code: same as previous code

mapper code:

```
import java.io.IOException;
import org.apache.hadoop.io.*
import org.apache.hadoop.mapred.*;
public class MeanTempMapper extends Mapper
    <LongWritable, Text , Text, IntWritable> {
  private text timekey = new Text();
  private IntWritableTemperature = new IntWritable();
  public void map ( LongWritable key  text value ,
                  context context) throws IO Exception,
                           Interrupted Exception {
    String.line = value.toString ().trim();
    if (line is Empty()) return;
    String [] parts = line.split(" , ");
    if (parts.length <2) return;
    String timestamp = parts[0].replace ("\", ",");
    String timekey = timestamp. substring (0, 16);
    try {
      int temp = integer.parseInt ( parts [1])
      timekey.set (timekeystr);
      temperature. set(temp);
      context.write(timekey , temperature);
    } catch ( number format Exception e) {
    }
  }
}
```

reducer code:

```
import same libraries as mapper ;
public class MeanTemperatureReducer extends Reducer <text,
                IntWritable, text, Double Writable> {
  private DoubleWritable meanTemprature = new DoubleWritable;
  public void reduce ( Text key ,iterable <IntWritable> values
                      content context)
                throws IO Exception, Interrupted Exception {
    int sum =0 , int count =0;
    for (intWritable val : values){
          sum += val.get(); count++;
    }
    double mean = (double)sum /count ;
    contextwrite (key, meanTemprature);
} }
```

Mapper:

```
#!/usr/bin/env python3
import sys

for line in sys.stdin:
    line = line.strip()

    parts = line.split()
    date, temp = parts
    temp = float(temp)
    print(f"{date}\t{temp}")
```

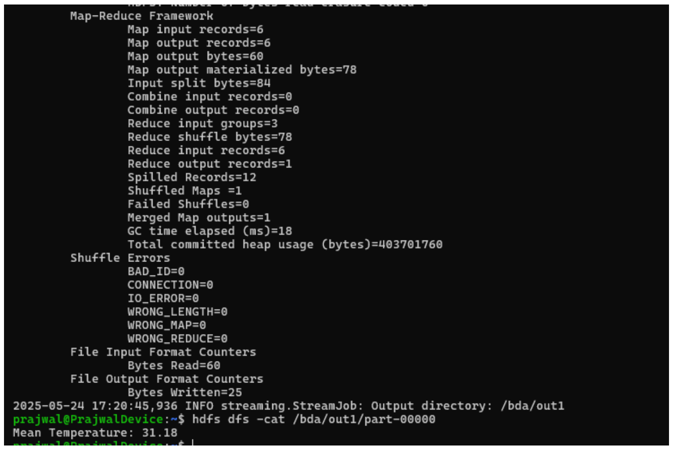
Reducer1:

```
#!/usr/bin/env python3
import sys
count = 0
total_temp = 0.0
for line in sys.stdin:
    line = line.strip()
    key, value = line.split("\t")
    try:
        total_temp += float(value)
        count += 1
    except ValueError:
        continue

if count > 0:
    mean_temp = total_temp / count
    print(f"Mean Temperature: {mean_temp:.2f}")
else:
    print("No valid temperature records.")
```

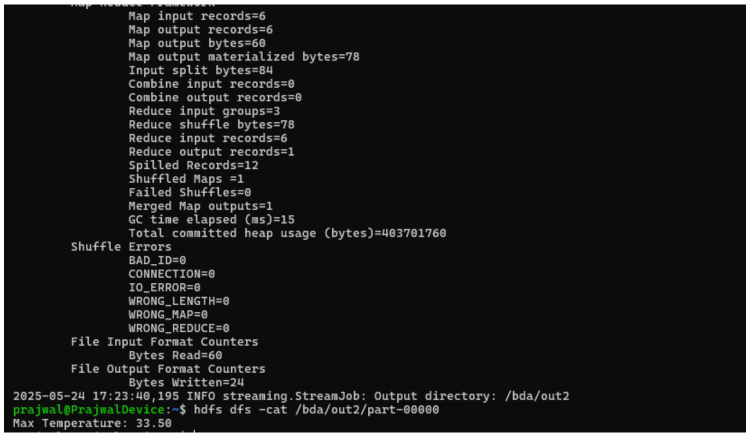
Reducer2:

```
#!/usr/bin/env python3
import sys

max_temp = float('-inf')

for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    try:
        key, value = line.split("\t")
        temp = float(value)
        if temp > max_temp:
            max_temp = temp
    except ValueError:
        continue

if max_temp != float('-inf'):
    print(f"Max Temperature: {max_temp:.2f}")
else:
    print("No valid temperature records.")
```

Codes Output:



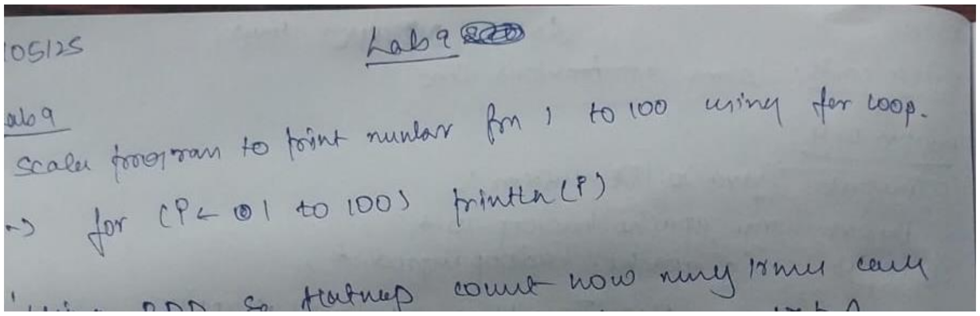
```
        Map-Reduce Framework
                Map input records=6
                Map output records=6
                Map output bytes=60
                Map output materialized bytes=78
                Input split bytes=84
                Combine input records=0
                Combine output records=0
                Reduce input groups=3
                Reduce shuffle bytes=78
                Reduce input records=6
                Reduce output records=1
                Spilled Records=12
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=18
                Total committed heap usage (bytes)=403701760
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=60
        File Output Format Counters
                Bytes Written=25
2025-05-24 17:20:45,936 INFO streaming.StreamJob: Output directory: /bda/out1
prajwal@PrajwalDevice:~$ hdfs dfs -cat /bda/out1/part-00000
Mean Temperature: 31.18
```



```
                Map input records=6
                Map output records=6
                Map output bytes=60
                Map output materialized bytes=78
                Input split bytes=84
                Combine input records=0
                Combine output records=0
                Reduce input groups=3
                Reduce shuffle bytes=78
                Reduce input records=6
                Reduce output records=1
                Spilled Records=12
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=15
                Total committed heap usage (bytes)=403701760
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=60
        File Output Format Counters
                Bytes Written=24
2025-05-24 17:23:40,195 INFO streaming.StreamJob: Output directory: /bda/out2
prajwal@PrajwalDevice:~$ hdfs dfs -cat /bda/out2/part-00000
Max Temperature: 33.50
```

Experiment-9

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



05/25 Lab 9

ab 9

Scala program to print number from 1 to 100 using for loop.

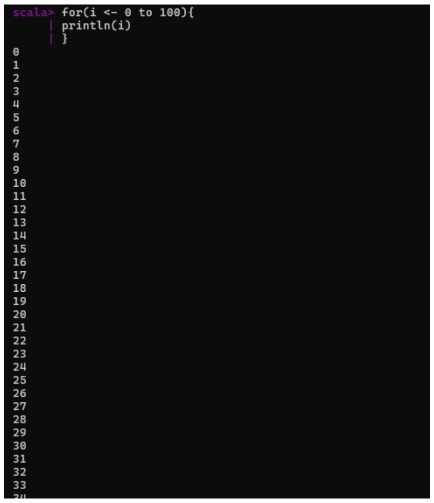
→ for (i ← 1 to 100) println(i)

Hashmap count how many times each

Scala Code:

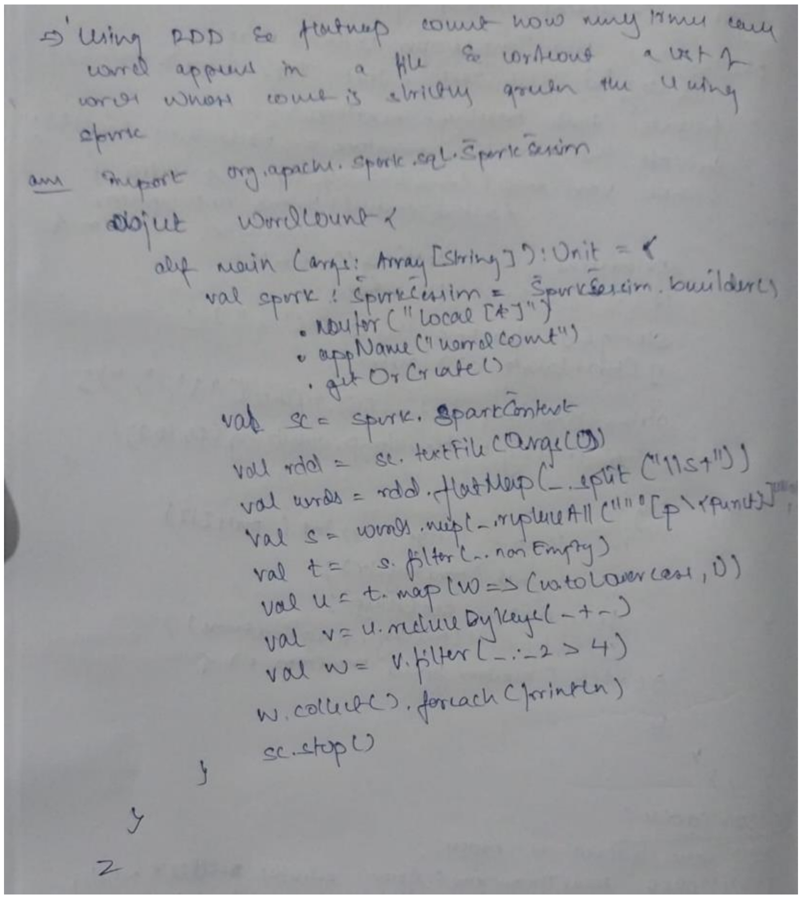
```
Scala> for(i <- 0  to 100){
          println(i)
          }
0
1
2
.
```

.Codes Output:



```
scala> for(i <- 0 to 100){
     | println(i)
     | }
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
```

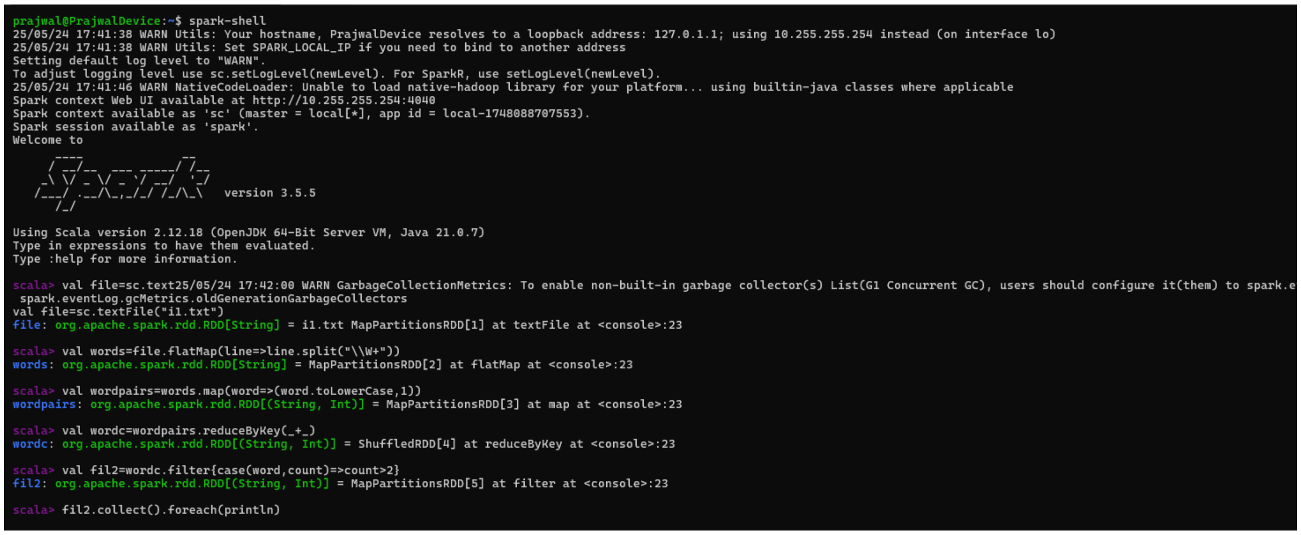
Using RDD and FlatMap count how many times each word appears in a file and write out a list of words whose count is strictly greater than 4 using Spark.



⇒ Using RDD & flatmap count how many times each word appears in a file & writeout a list of word whose count is strictly greater the 4 using spark

```
ans  import org.apache.spark.sql.SparkSession
   object WordCount {
      def main (args: Array[String]): Unit = {
        val spark : SparkSession = SparkSession.builder()
              .master("local[*]")
              .appName("word count")
              .getOrCreate()
          val sc = spark.sparkContext
          val rdd = sc.textFile(args(0))
          val words = rdd.flatMap(_.split("\\s+"))
          val s = words.map(_.replaceAll("""[\p{Punct}]""",
          val t =  s.filter(_.nonEmpty)
          val u = t.map(w => (w.toLowerCase, 1))
          val v = u.reduceByKey(_+_)
          val w = v.filter(_._2 > 4)
          w.collect().foreach(println)
          sc.stop()
      }
   }
 }
```

Codes Output:



```
prajwal@PrajwalDevice:~$ spark-shell
25/05/24 17:41:38 WARN Utils: Your hostname, PrajwalDevice resolves to a loopback address: 127.0.1.1; using 10.255.255.254 instead (on interface lo)
25/05/24 17:41:38 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/05/24 17:41:46 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://10.255.255.254:4040
Spark context available as 'sc' (master = local[*], app id = local-1748088707553).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.5
      /_/

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 21.0.7)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val file=sc.text25/05/24 17:42:00 WARN GarbageCollectionMetrics: To enable non-built-in garbage collector(s) List(G1 Concurrent GC), users should configure it(them) to spark.e
 spark.eventLog.gcMetrics.oldGenerationGarbageCollectors
val file=sc.textFile("i1.txt")
file: org.apache.spark.rdd.RDD[String] = i1.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> val words=file.flatMap(line=>line.split("\\W+"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:23

scala> val wordpairs=words.map(word=>(word.toLowerCase,1))
wordpairs: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:23

scala> val wordc=wordpairs.reduceByKey(_+_)
wordc: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23

scala> val fil2=wordc.filter{case(word,count)=>count>2}
fil2: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:23

scala> fil2.collect().foreach(println)
```