VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Big Data Analytics (23CS6PCBDA)

Submitted by:

Nikhilesh C (1BM22CS181)

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

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
March 2024 - June 2024

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

CERTIFICATE

This is to certify that the Lab work entitled “**Big Data Analytics**” carried out by **Nikhilesh C (1BM22CS181),** 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 **Big Data Analytics – (23CS6PCBDA)** work prescribed for the said degree.

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

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

Table Of Contents

Sl.no	Program details	Pg no
1	MongoDB- CRUD Operations Demonstration (Practice and Self Study)	4-7
2	Perform the following DB operations using Cassandra.	8-12
3	Perform the following DB operations using Cassandra	13-20
4	Execution of HDFS Commands for interaction with Hadoop Environment. (Minimum 10 commands to be executed)	21-25
5	Implement Wordcount program on Hadoop framework	26-31
6	Create a MapReduce program to find average temperature for each year from NCDC data set. b) find the mean max temperature for every month. For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words.	32-32
7	Write a Scala program to print numbers from 1 to 100 using loop.	33-34
8	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.	35-39
9	Write a simple streaming program in Spark to receive text data streams on a particular port, perform basic text cleaning (like white space removal, stop words removal, lemmatization, etc.), and print the cleaned text on the screen. (Open Ended Question).	40-40

Course Outcomes

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

CO2: Analyze data analytic techniques for a given problem.

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

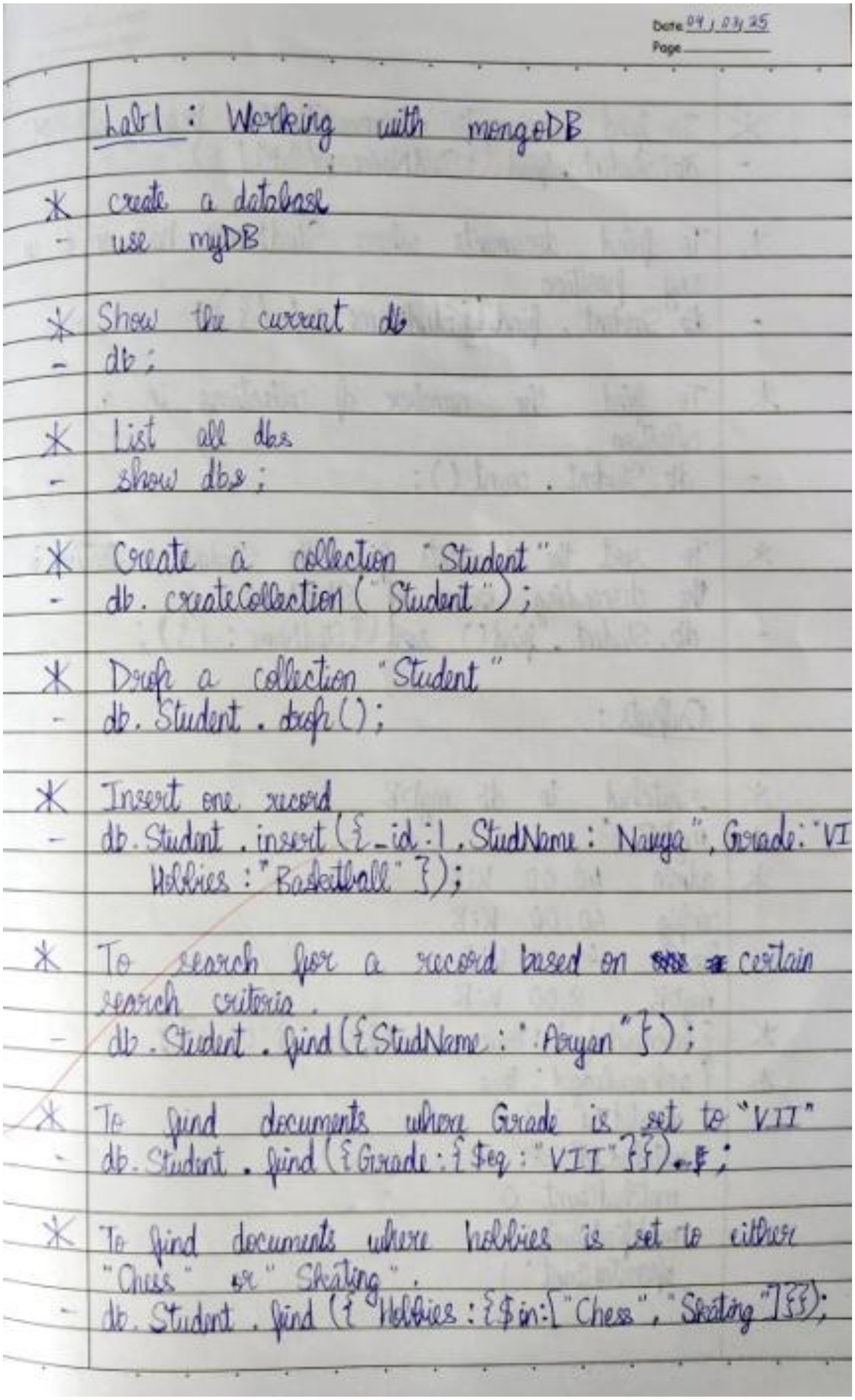
1. Experiments

Experiment - 1

Question:

Perform the following DB operations using Cassandra.

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



Date 04/03/25
Page

Lab1 : Working with mongoDB

* create a database
- use myDB

* Show the current db
- db;

* List all dbs
- show dbs;

* Create a collection "Student"
- db. createCollection ("Student");

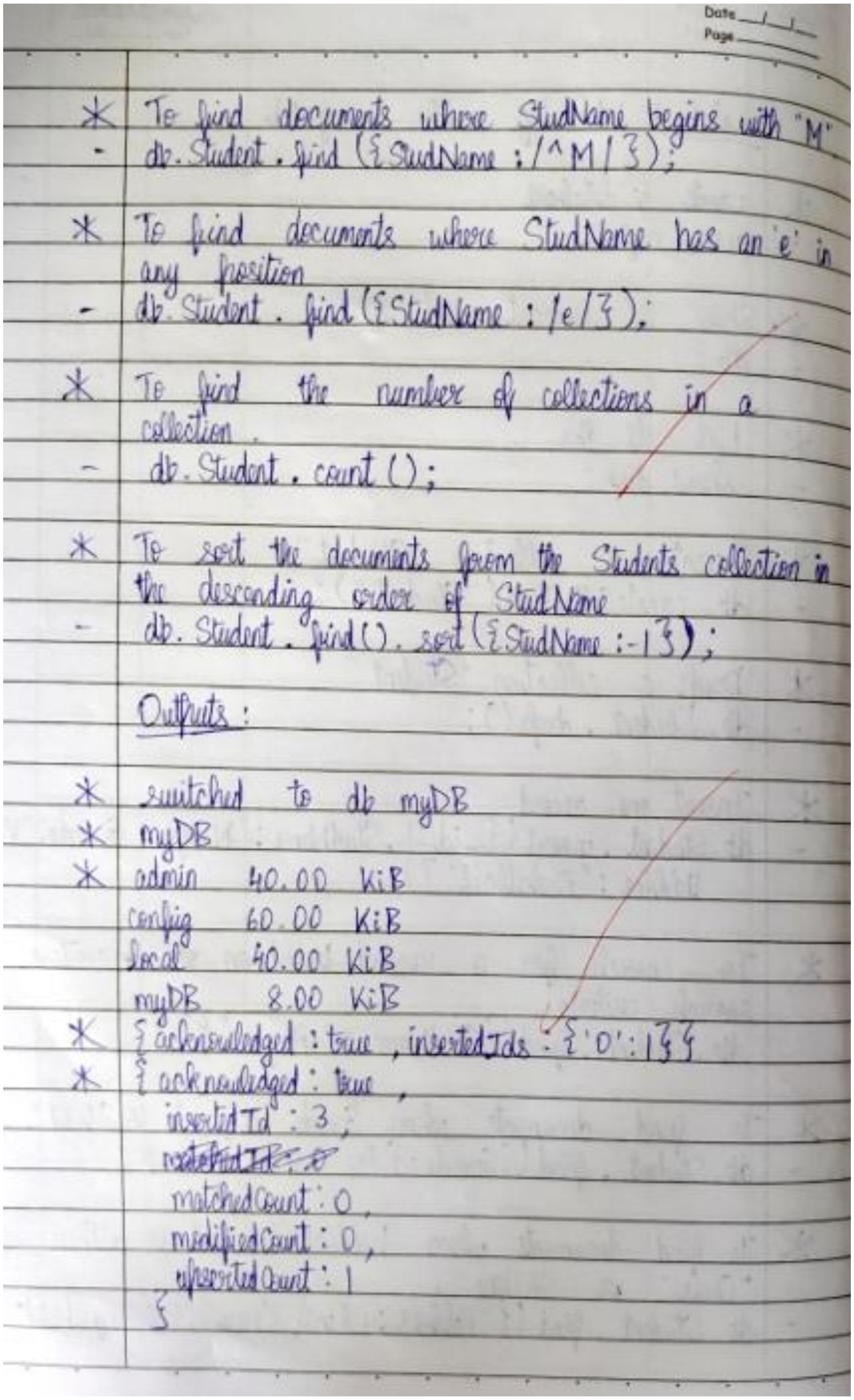
* Drop a collection "Student"
- db. Student . drop();

* Insert one record
- db. Student . insert ({_id:1, StudName : "Navya", Grade: "VI
  Hobbies : "Basketball" });

* To search for a record based on certain search criteria.
- db. Student . find ({StudName : "Aryan"});

* To find documents where Grade is set to "VII"
- db. Student . find ({Grade : {$eq : "VII"}});

* To find documents where hobbies is set to either "Chess" or "Skating".
- db. Student . find ({Hobbies : {$in: ["Chess", "Skating"]}});



Date / /
Page

* To find documents where StudName begins with "M"

```
- db.Student.find({StudName:/^M/});
```

* To find documents where StudName has an 'e' in any position

```
- db.Student.find({StudName:/e/});
```

* To find the number of collections in a collection.

```
- db.Student.count();
```

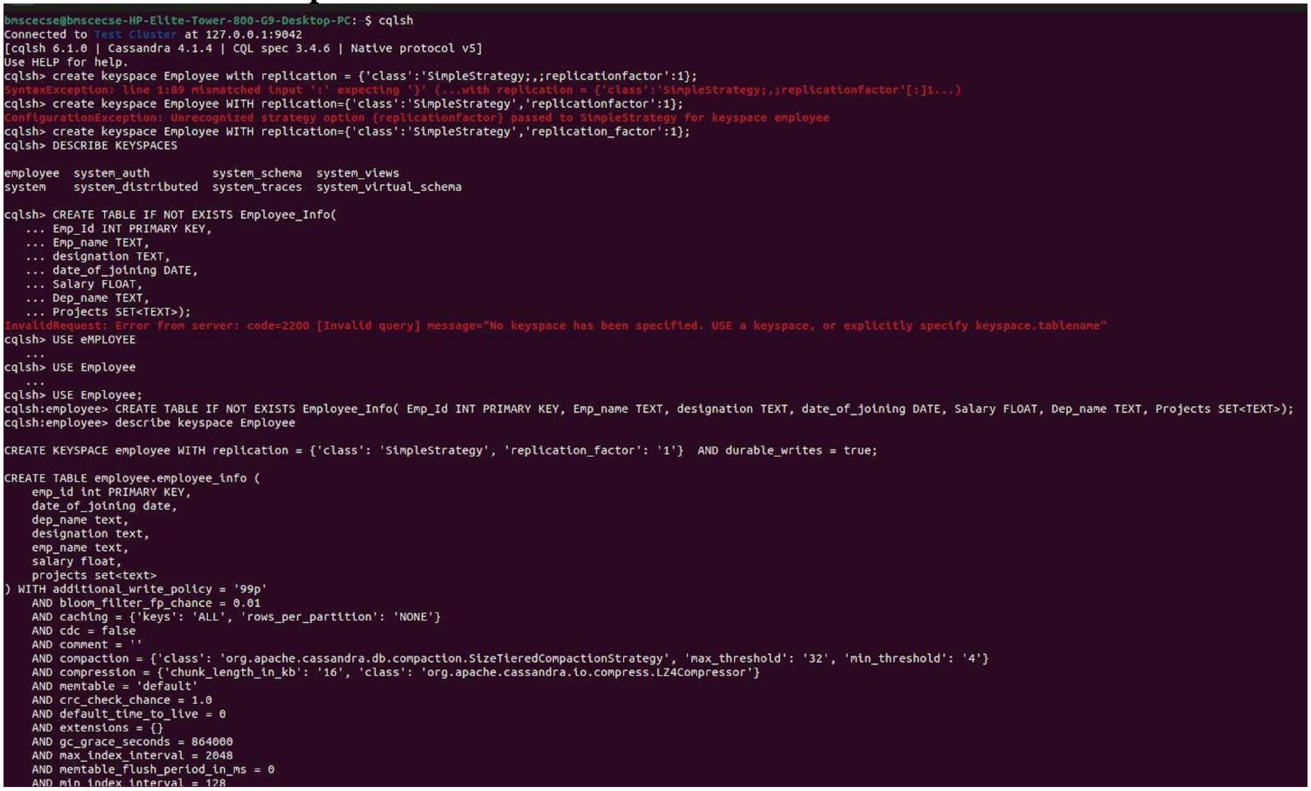
* To sort the documents from the Students collection in the descending order of StudName

```
- db.Student.find().sort({StudName:-1});
```

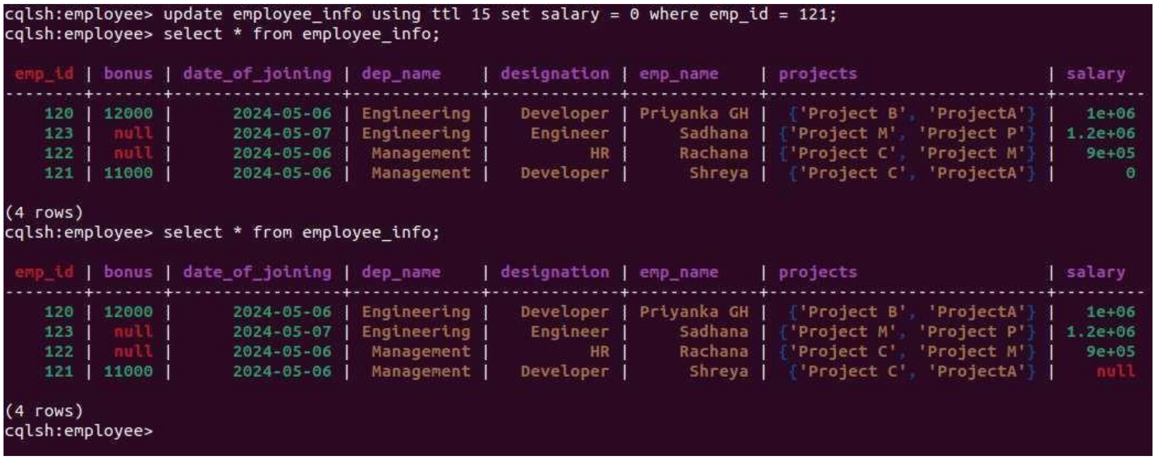
Outputs:

```
* switched to db myDB
* myDB
* admin    40.00 KiB
  config   60.00 KiB
  local    40.00 KiB
  myDB      8.00 KiB
* {acknowledged : true , insertedIds : {'0':1}}
* {acknowledged : true ,
    insertedId : 3 ,
    ~~matchedId : 0~~
    matchedCount : 0 ,
    modifiedCount : 0 ,
    upsertedCount : 1
  }
```

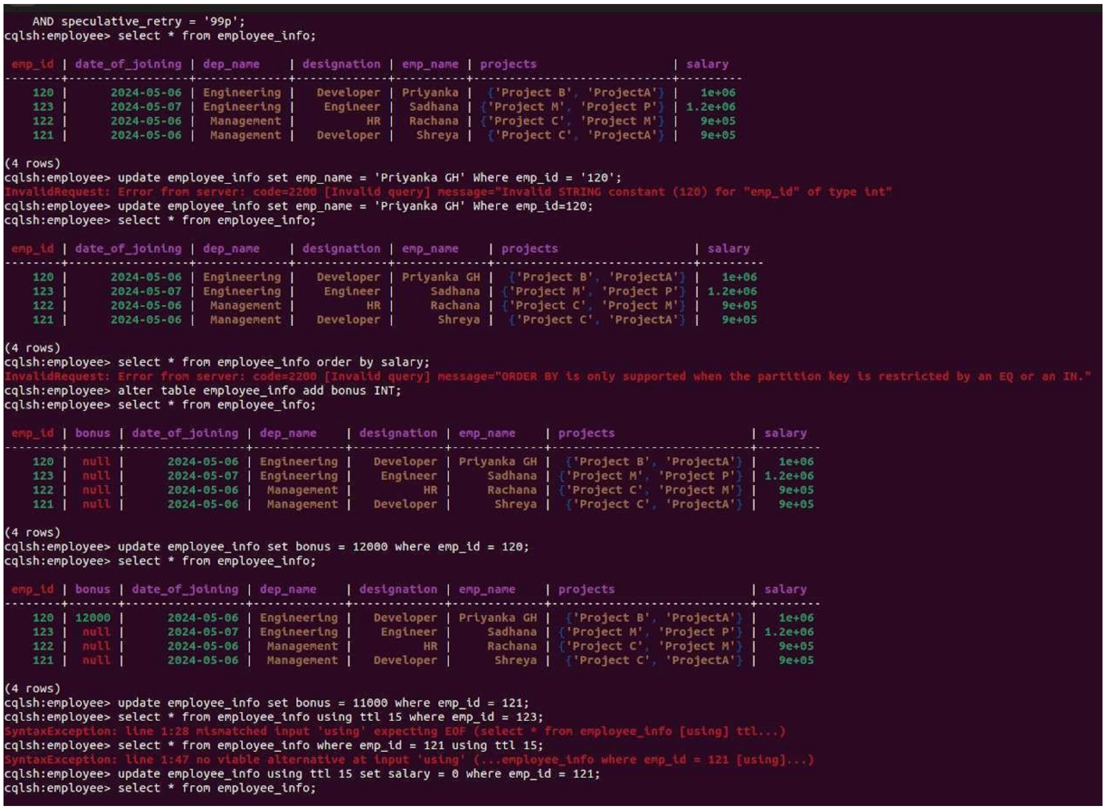
1.1.2 Code with Output:



```
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 1:89 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
    AND min_index_interval = 128
```



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

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

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

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

(4 rows)
cqlsh:employee>
```



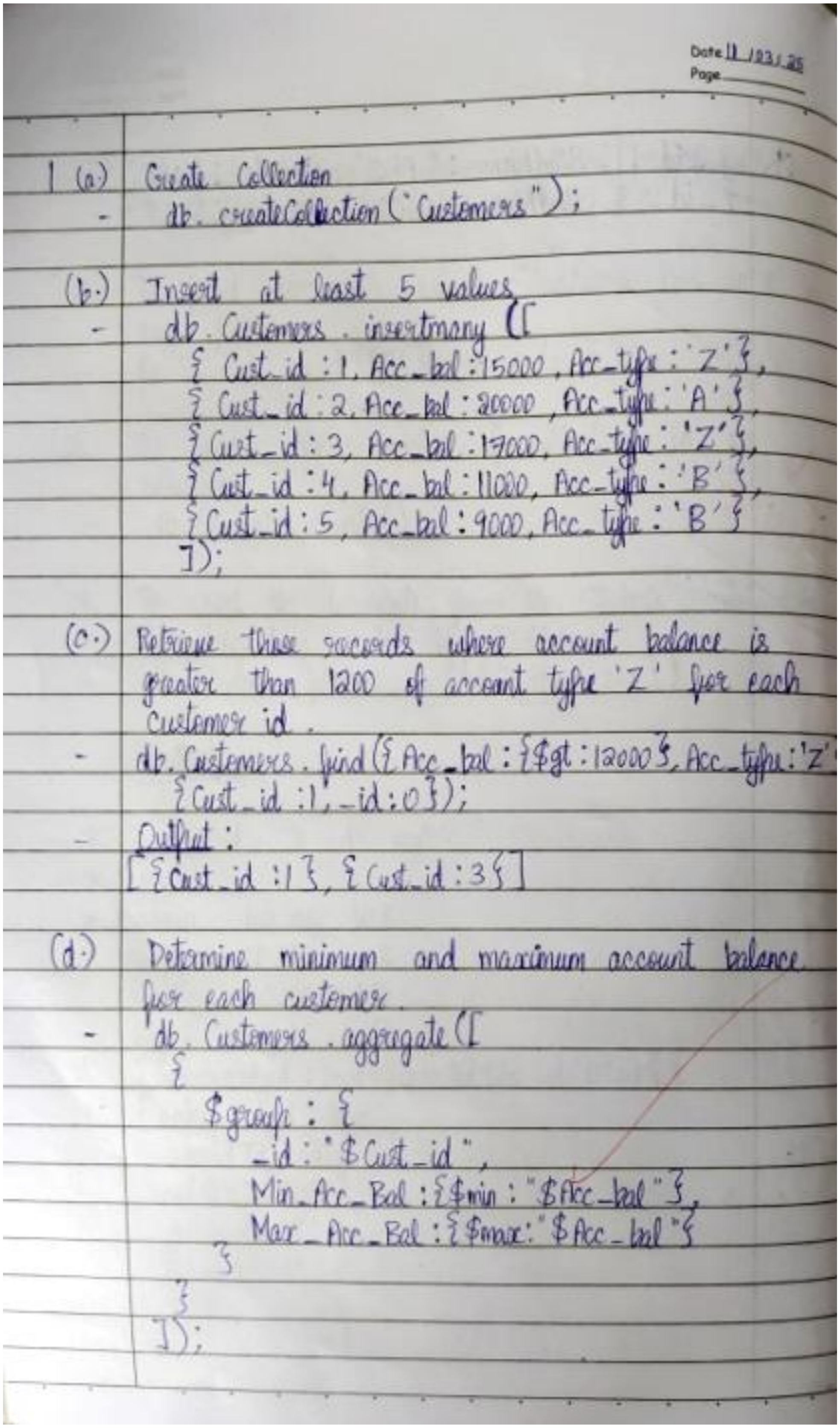
```
    AND speculative_retry = '99p';
cqlsh:employee> select * from employee_info;

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

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

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

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

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

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

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

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

1.2 Experiment - 2

1.2.1 Question:

Perform the following DB operations using Cassandra:

- Create a keyspace by name Library
- Create a column family by name Library-Info with attributes Stud_Id Primary Key, Counter_value of type Counter, Stud_Name, Book-Name, Book-Id, Date_of_issue
- Insert the values into the table in batch
- Display the details of the table created and increase the value of the counter
- Write a query to show that a student with id 112 has taken a book “BDA” 2 times.
- Export the created column to a csv file
- Import a given csv dataset from local file system into Cassandra column family**.**



Date 11/03/25
Page

1 (a) Create Collection

```
- db. createCollection ("Customers");
```

(b.) Insert at least 5 values

```
- db. Customers . insertmany ([
    { Cust_id : 1, Acc_bal : 15000 , Acc_type : 'Z' },
    { Cust_id : 2, Acc_bal : 20000 , Acc_type : 'A' },
    { Cust_id : 3, Acc_bal : 17000, Acc_type : 'Z' },
    { Cust_id : 4, Acc_bal : 11000, Acc_type : 'B' },
    { Cust_id : 5, Acc_bal : 9000, Acc_type : 'B' }
  ]);
```

(c.) Retrieve those records where account balance is greater than 1200 of account type 'Z' for each customer id.

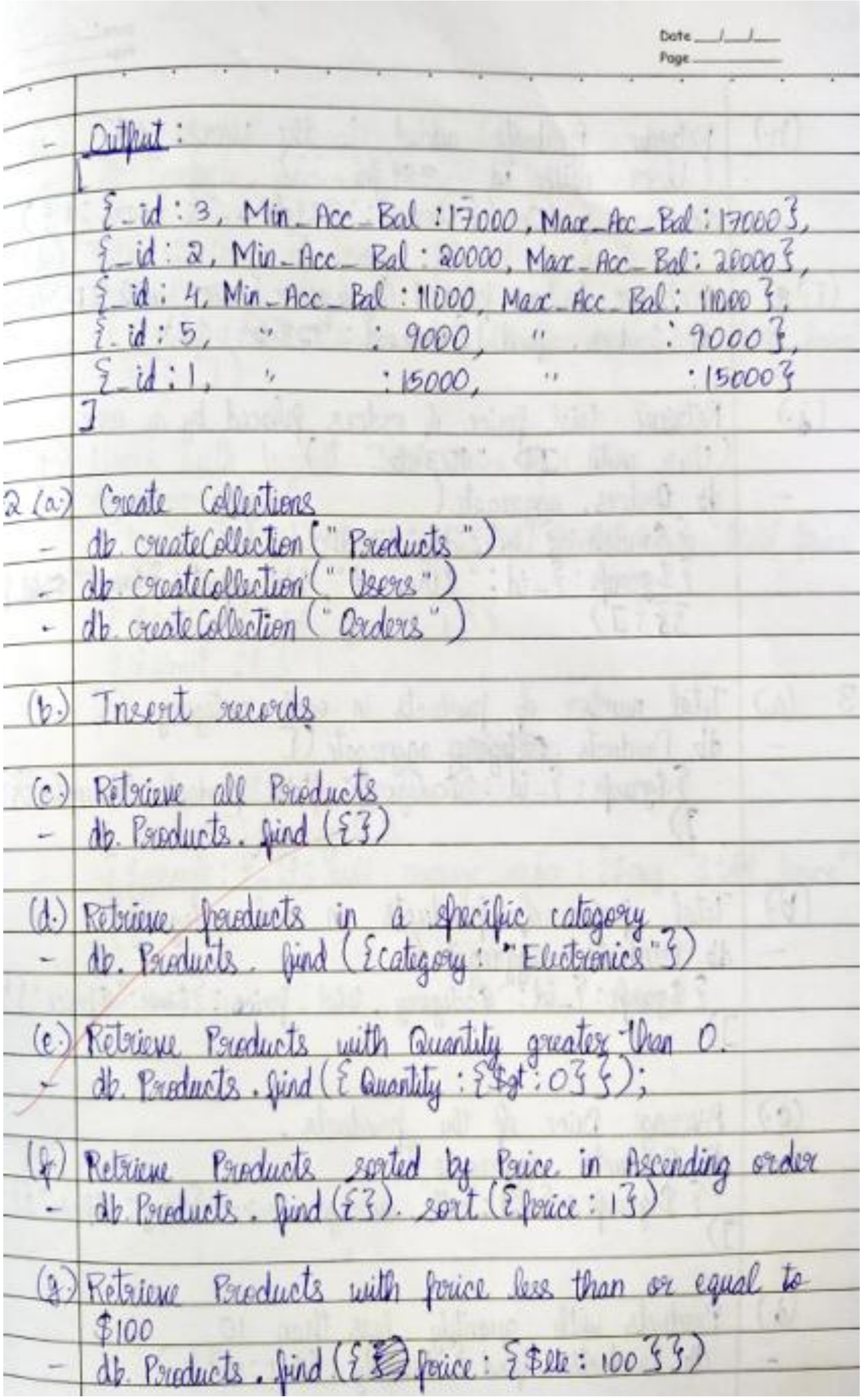
```
- db. Customers . find({ Acc_bal : {$gt : 12000 }, Acc_type : 'Z'
    {Cust_id : 1, _id : 0 });
```

- Output :

```
[ {Cust_id : 1 }, { Cust_id : 3 } ]
```

(d.) Determine minimum and maximum account balance for each customer.

```
- db. Customers . aggregate ([
    {
      $group : {
         _id : "$Cust_id",
         Min_Acc_Bal : {$min : "$Acc_bal" },
         Max_Acc_Bal : { $max: "$Acc_bal" }
      }
    }
  ]);
```



Date __/__/__
Page ______

- Output :

```
[
 {_id : 3, Min_Acc_Bal : 17000, Max_Acc_Bal : 17000},
 {_id : 2, Min_Acc_Bal : 20000, Max_Acc_Bal : 20000},
 {_id : 4, Min_Acc_Bal : 11000, Max_Acc_Bal : 11000},
 {_id : 5,     "      : 9000,     "      : 9000},
 {_id : 1,     "      : 15000,    "      : 15000}
]
```

2 (a.) Create Collections

- db.createCollection("Products")
- db.createCollection("Users")
- db.createCollection("Orders")

(b.) Insert records

(c.) Retrieve all Products

- db.Products.find({})

(d.) Retrieve products in a specific category

- db.Products.find({category: "Electronics"})

(e.) Retrieve Products with Quantity greater than 0.

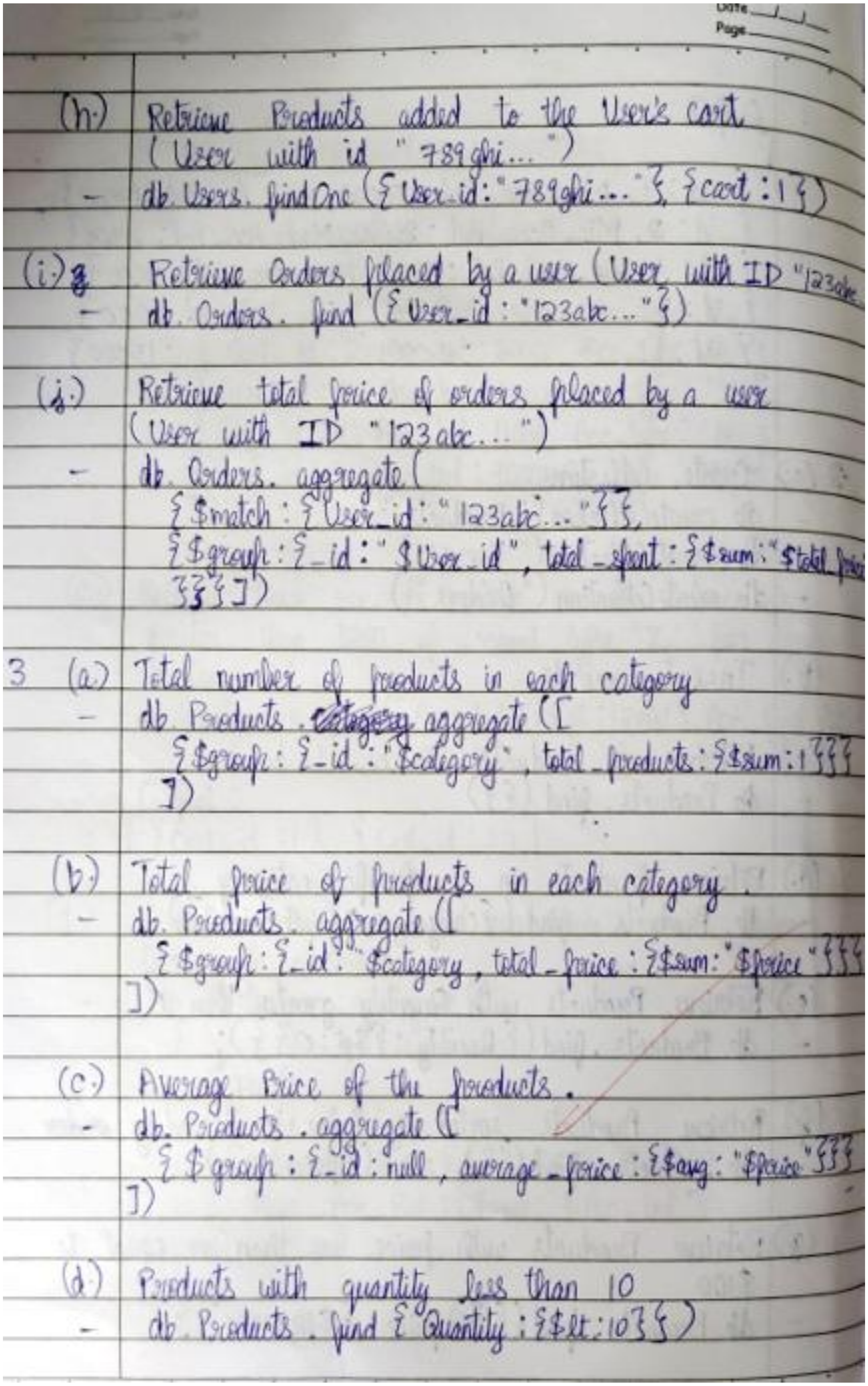
- db.Products.find({Quantity : {$gt: 0}});

(f.) Retrieve Products sorted by Price in Ascending order

- db.Products.find({}).sort({price : 1})

(g.) Retrieve Products with price less than or equal to $100

- db.Products.find({ price : {$lte : 100}})



Date __/__/__
Page ____

(h.) Retrieve Products added to the User's cart (User with id "789ghi...")

```
- db.Users.findOne({User_id:"789ghi..."}, {cart:1})
```

(i.) Retrieve Orders placed by a user (User with ID "123abc

```
- db.Orders.find({User_id:"123abc..."})
```

(j.) Retrieve total price of orders placed by a user (User with ID "123abc...")

```
- db.Orders.aggregate(
    {$match:{User_id:"123abc..."}},
    {$group:{_id:"$User_id", total_spent:{$sum:"$total_pri
    }}}])
```

3 (a.) Total number of products in each category

```
- db.Products.~~category~~ aggregate([
    {$group:{_id:"$category", total_products:{$sum:1}}}
  ])
```

(b.) Total price of products in each category.

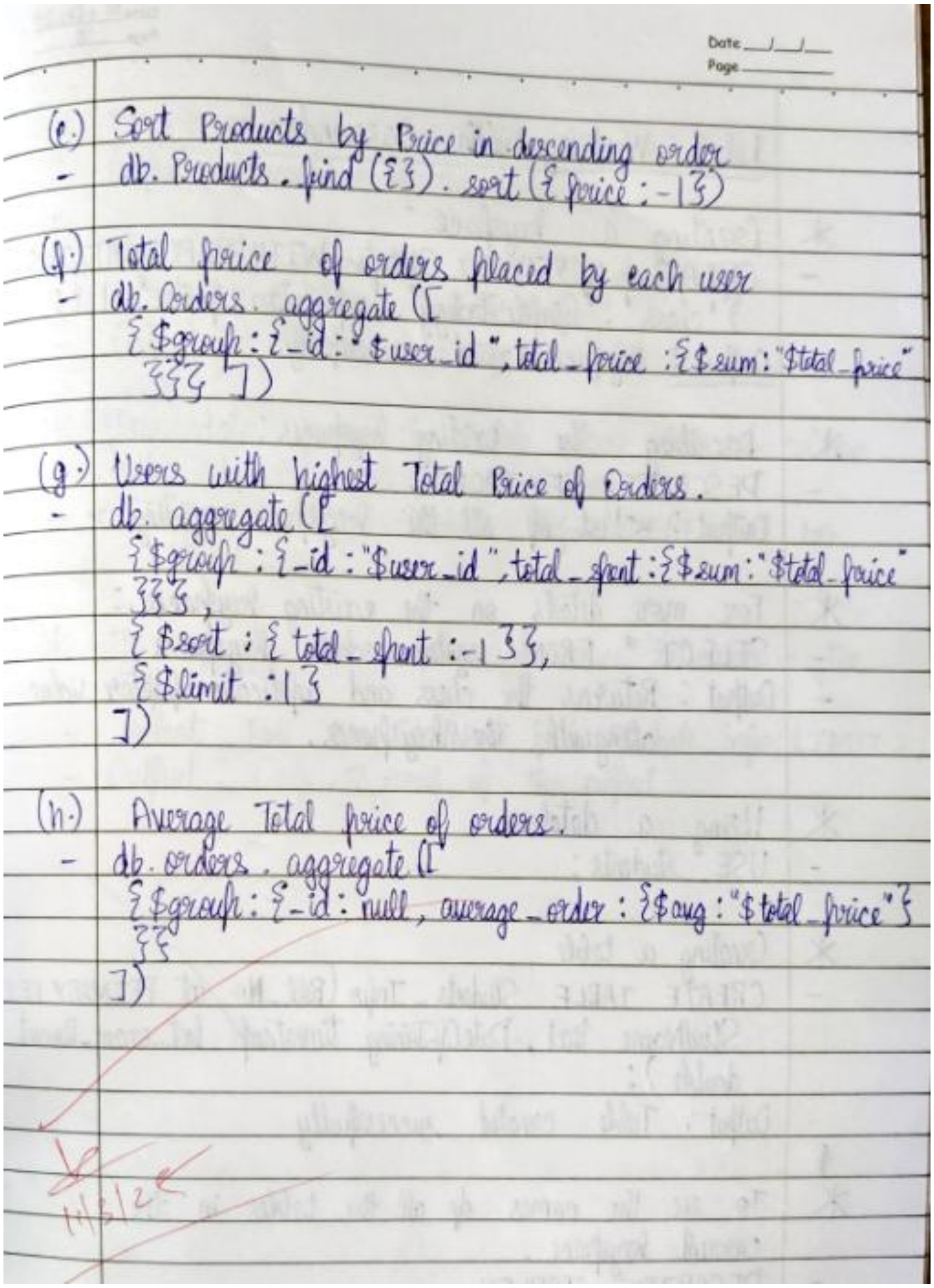
```
- db.Products.aggregate([
    {$group:{_id:"$category, total_price:{$sum:"$price"}}}
  ])
```

(c.) Average Price of the products.

```
- db.Products.aggregate([
    {$group:{_id:null, average_price:{$avg:"$price"}}}
  ])
```

(d.) Products with quantity less than 10

```
- db.Products.find{Quantity:{$lt:10}})
```



Date __/__/__
Page ____

(e.) Sort Products by Price in descending order

```
- db. Products . find ({}). sort ({ price : -1})
```

(f.) Total price of orders placed by each user

```
- db. Orders . aggregate ([
   { $group : {_id : "$user_id", total_price : {$sum: "$total_price"
   }}} ])
```

(g.) Users with highest Total Price of Orders.

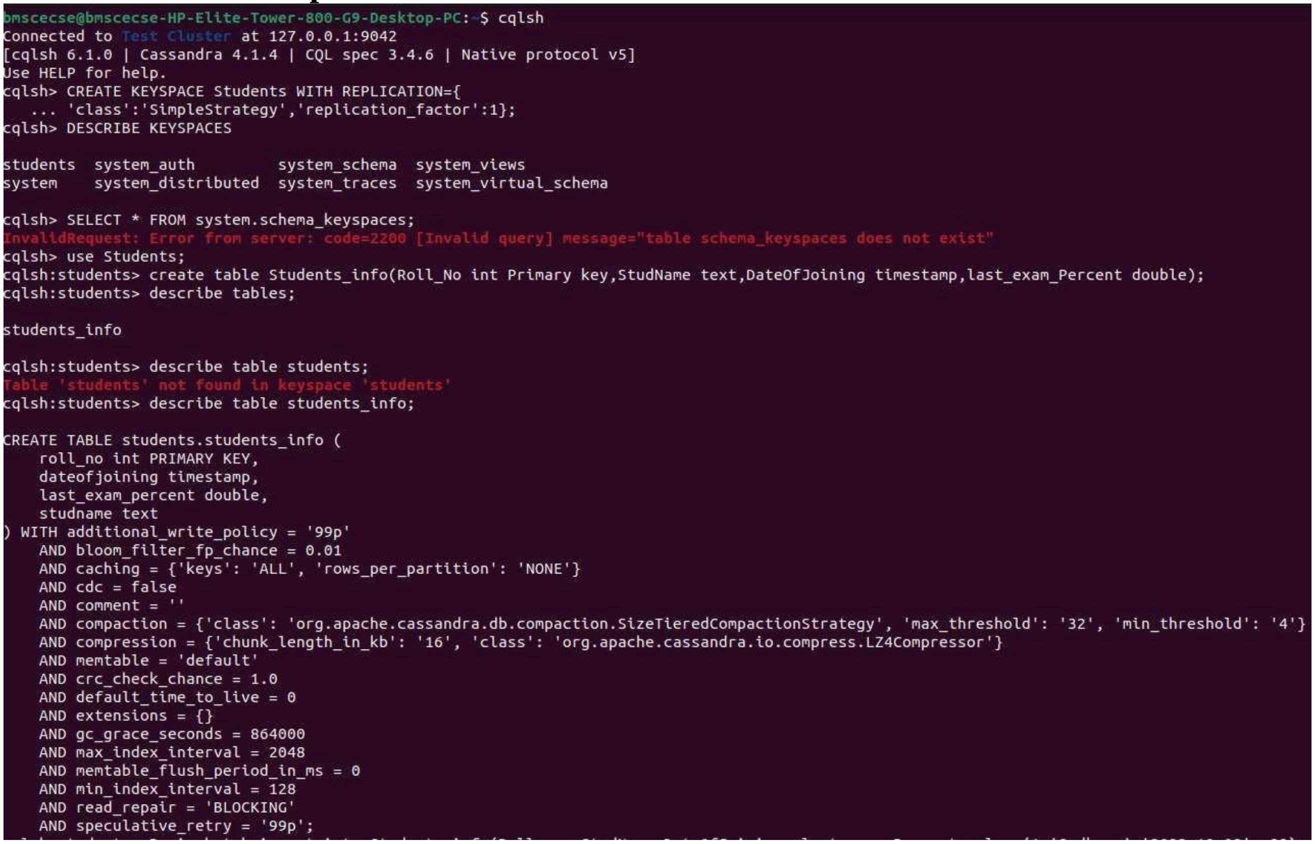
```
- db. aggregate ([
   {$group : {_id : "$user_id", total_spent : {$sum : "$total_price"
   }}},
   { $sort : { total_spent : -1 }},
   {$limit : 1}
   ])
```

(h.) Average Total price of orders.

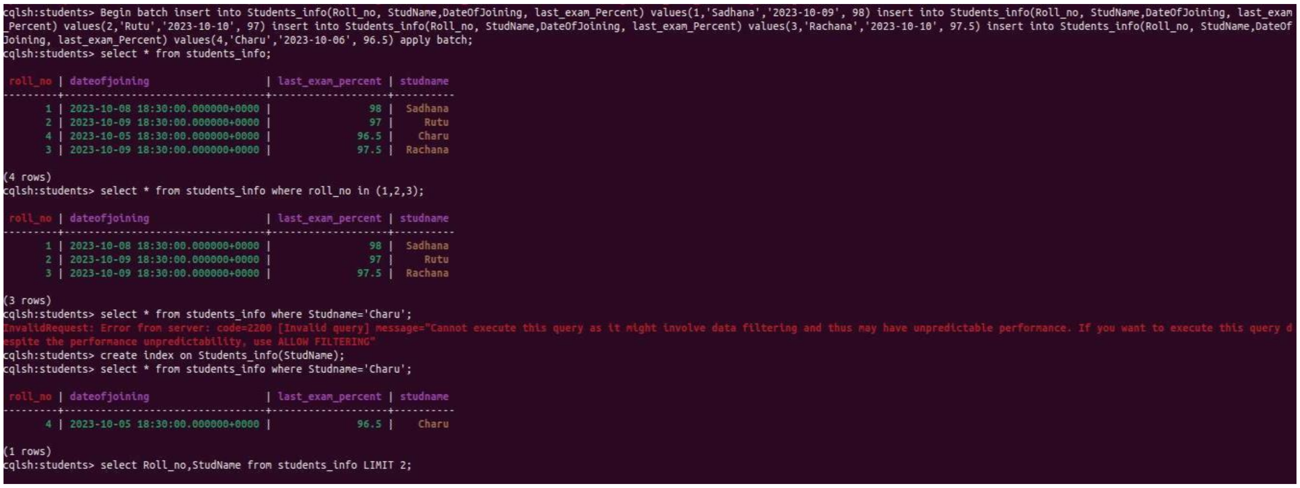
```
- db. orders . aggregate ([
   {$group : {_id : null, average_order : {$avg : "$total_price"}
   }}
   ])
```

11/3/24

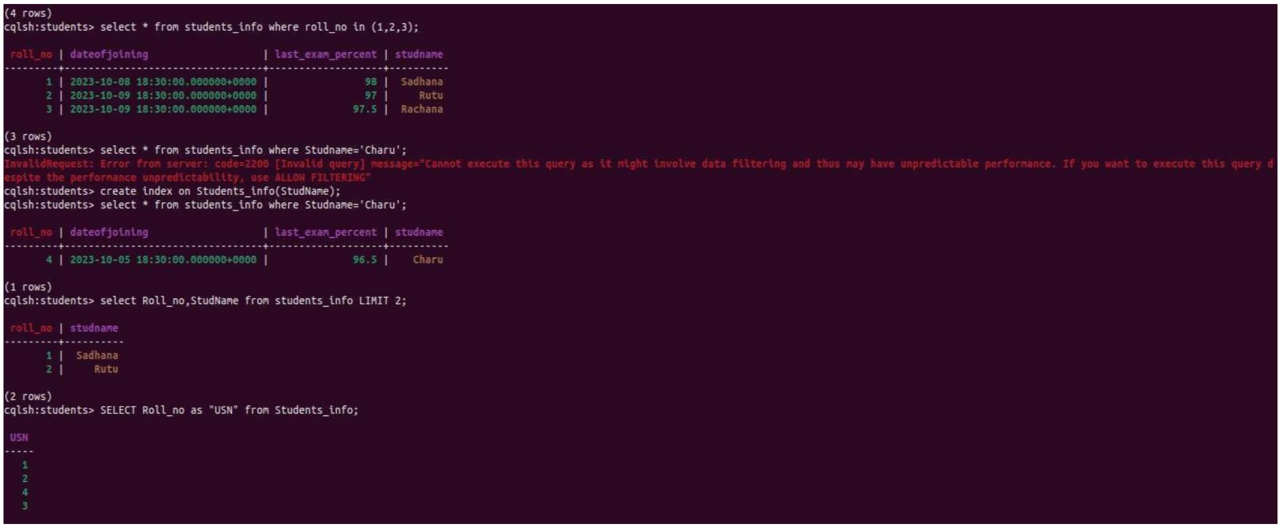
1.2.2 Code with Output:



```
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;
```

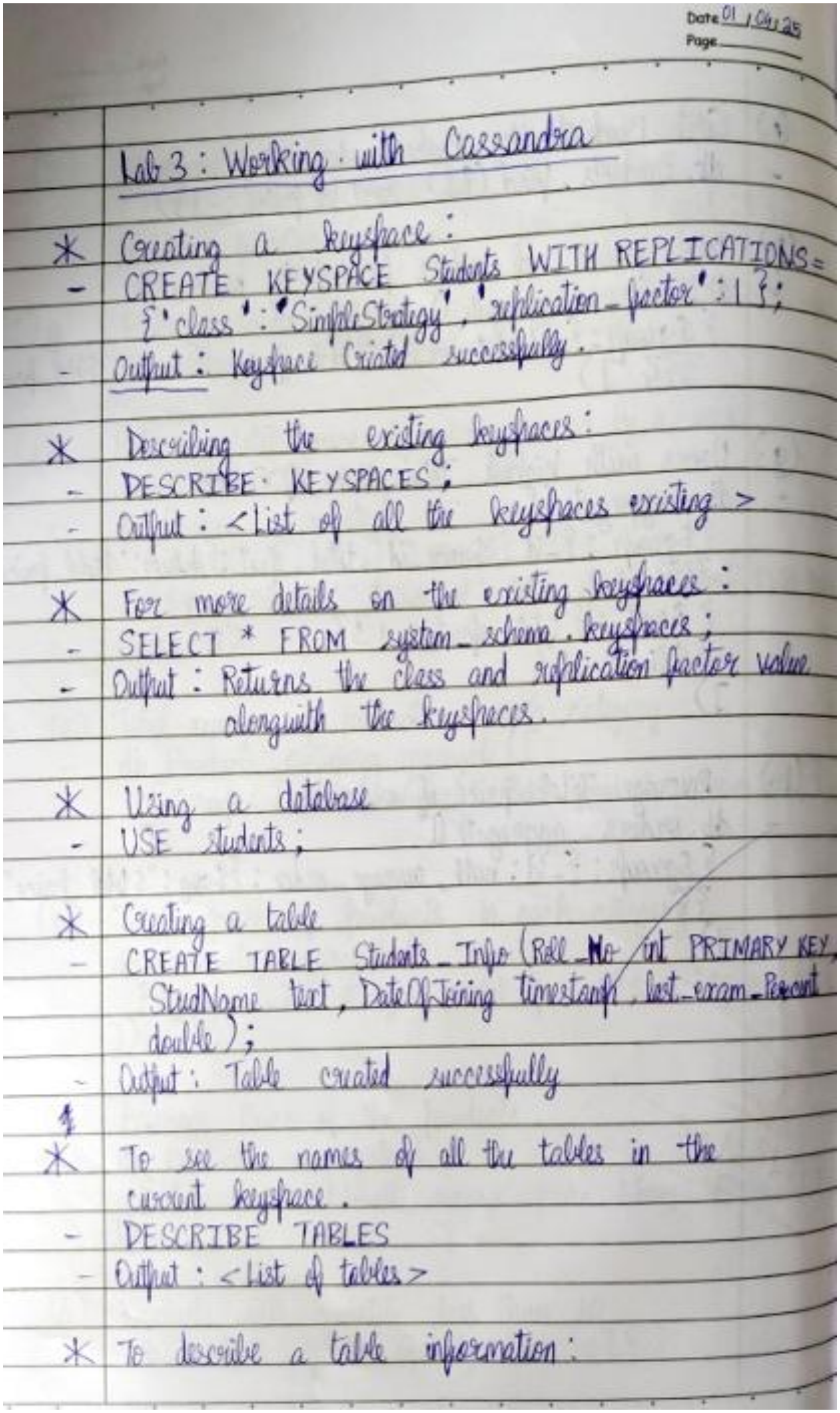


```
(4 rows)
cqlsh:students> select * from students_info where roll_no in (1,2,3);

 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
```

1.3 Experiment - 3

1.3.1 Question:

MongoDB - CRUD Demonstration.



Date 01 / 04 / 25
Page

Lab 3 : Working with Cassandra

* Creating a keyspace :
- CREATE KEYSPACE Students WITH REPLICATIONS = {'class':'SimpleStrategy', 'replication_factor': 1 };
- Output : Keyspace Criated successfully.

* Describing the existing keyspaces :
- DESCRIBE KEYSPACES ;
- Output : <list of all the keyspaces existing >

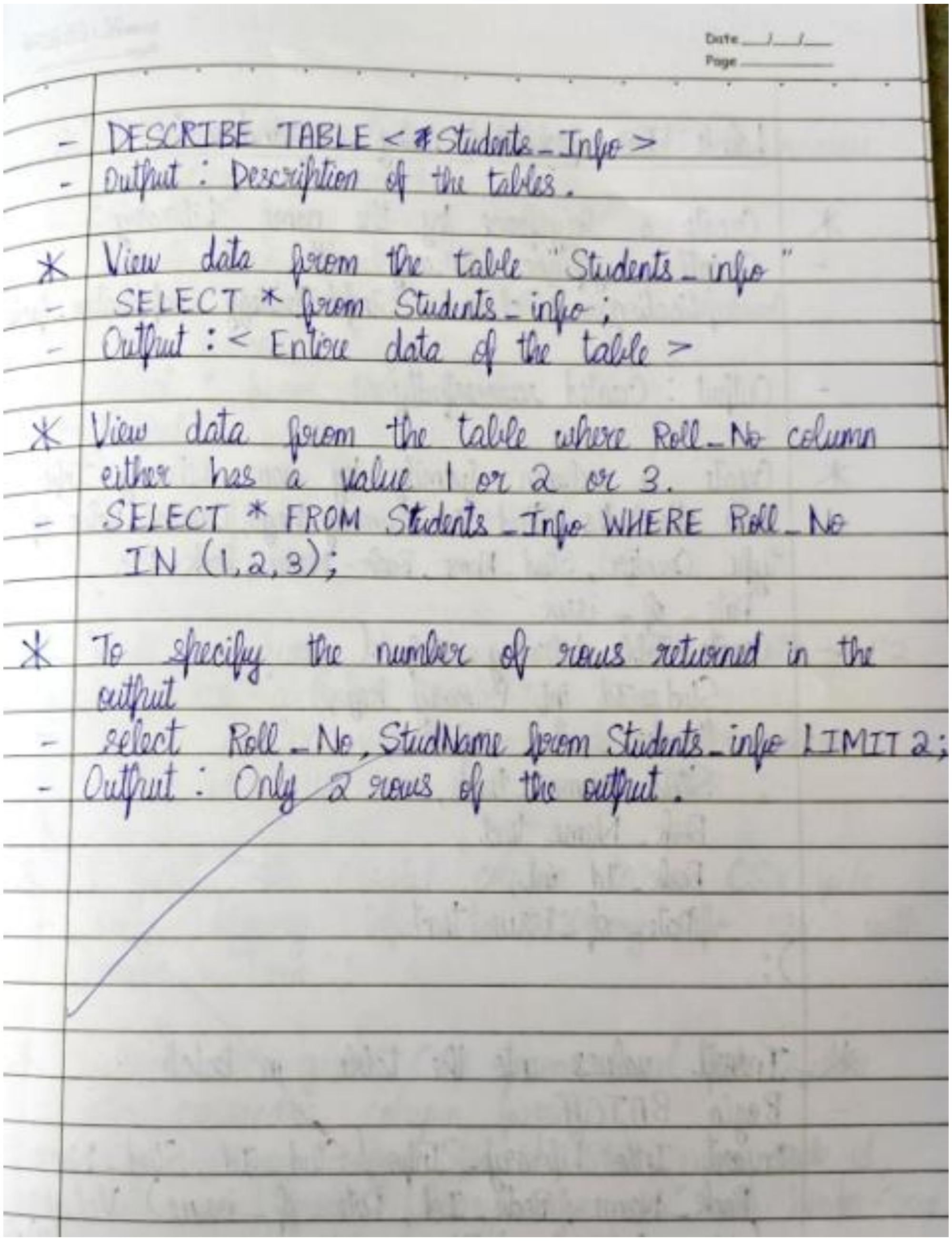
* For more details on the existing keyspaces :
- SELECT * FROM system_schema . keyspaces ;
- Output : Returns the class and replication factor value alongwith the keyspaces.

* Using a database
- USE students ;

* Creating a table
- CREATE TABLE Students_Info (Roll_No int PRIMARY KEY, StudName text, DateOfJoining timestamp, last_exam_Percent double);
- Output : Table created successfully

* To see the names of all the tables in the current keyspace.
- DESCRIBE TABLES
- Output : <List of tables >

* To describe a table information :



Date __/__/__
Page ____

- DESCRIBE TABLE < ~~*~~Students_Info >
- Output : Description of the tables.

* View data from the table "Students_info"
- SELECT * from Students_info;
- Output : < Entire data of the table >

* View data from the table where Roll_No column either has a value 1 or 2 or 3.
- SELECT * FROM Students_Info WHERE Roll_No IN (1,2,3);

* To specify the number of rows returned in the output
- select Roll_No, StudName from Students_info LIMIT 2;
- Output : Only 2 rows of the output.

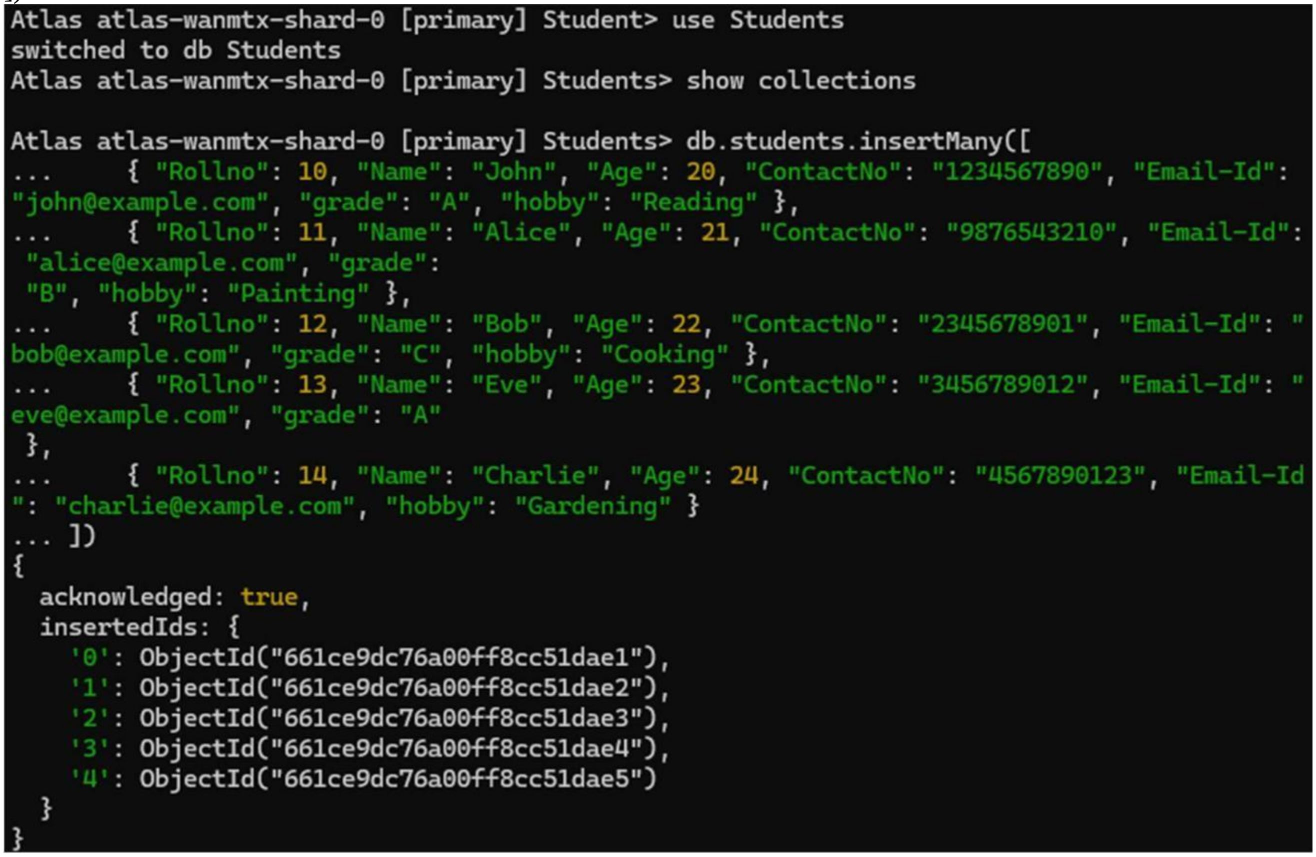
1.3.2 Code with Output:

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

use Students

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

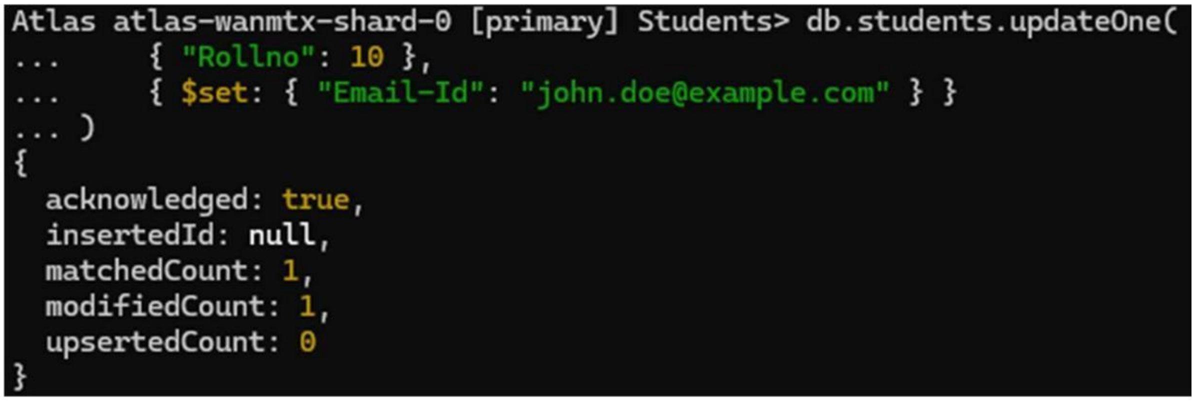
```
db.students.insertMany([
   { "Rollno": 10, "Name": "John", "Age": 20, "ContactNo": "1234567890", "Email-Id":
"john@example.com", "grade": "A", "hobby": "Reading" },
   { "Rollno": 11, "Name": "Alice", "Age": 21, "ContactNo": "9876543210", "Email-Id":
"alice@example.com", "grade": "B", "hobby": "Painting" },
   { "Rollno": 12, "Name": "Bob", "Age": 22, "ContactNo": "2345678901", "Email-Id":
"bob@example.com", "grade": "C", "hobby": "Cooking" },
   { "Rollno": 13, "Name": "Eve", "Age": 23, "ContactNo": "3456789012", "Email-Id":
"eve@example.com", "grade": "A" },
   { "Rollno": 14, "Name": "Charlie", "Age": 24, "ContactNo": "4567890123", "Email-Id":
"charlie@example.com", "hobby": "Gardening" }
])
```



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

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

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

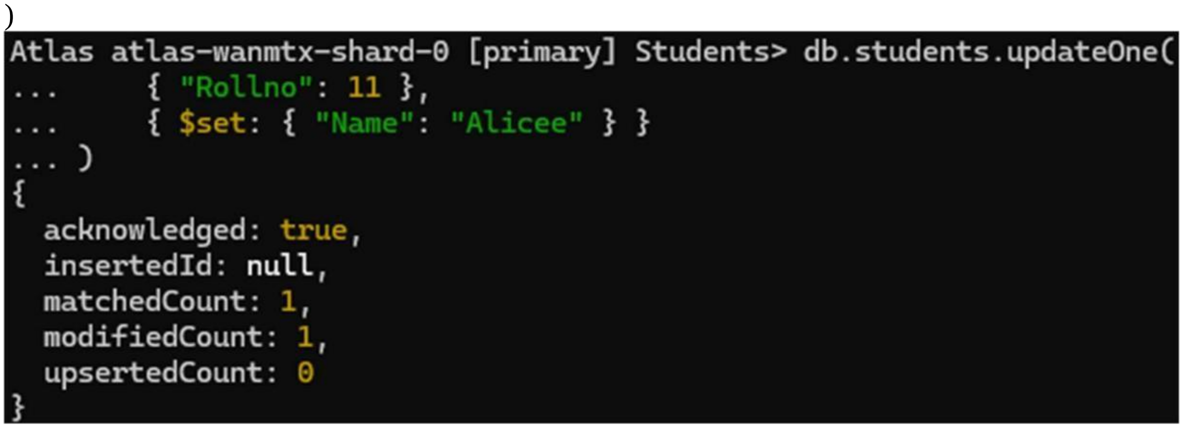
```
db.students.updateOne(
   { "Rollno": 10 },
   { $set: { "Email-Id": "john.doe@example.com" } }
)
```



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

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

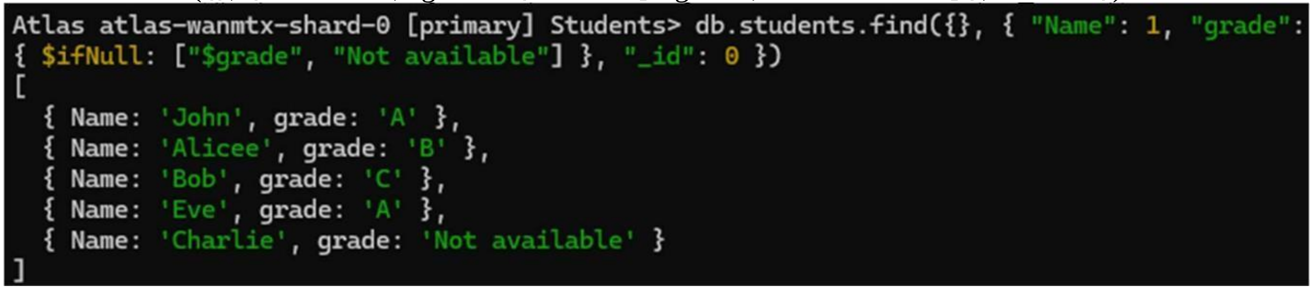
```
db.students.updateOne(
   { "Rollno": 11 },
   { $set: { "Name": "Alicee" } }
)
```



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

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

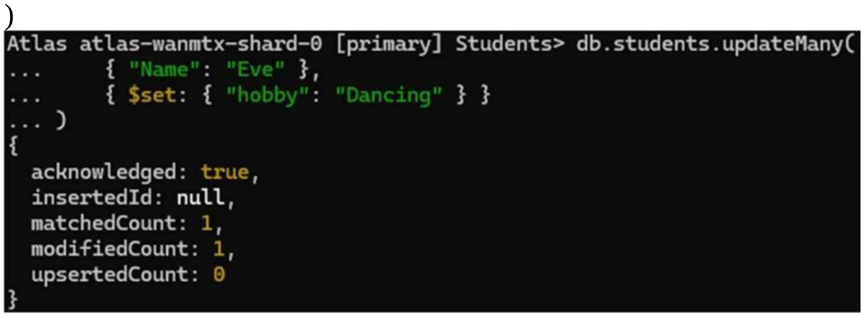
```
db.students.find({}, { "Name": 1, "grade": { $ifNull: ["$grade", "Not available"] }, "_id": 0 })
```



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

6. Update to add hobbies

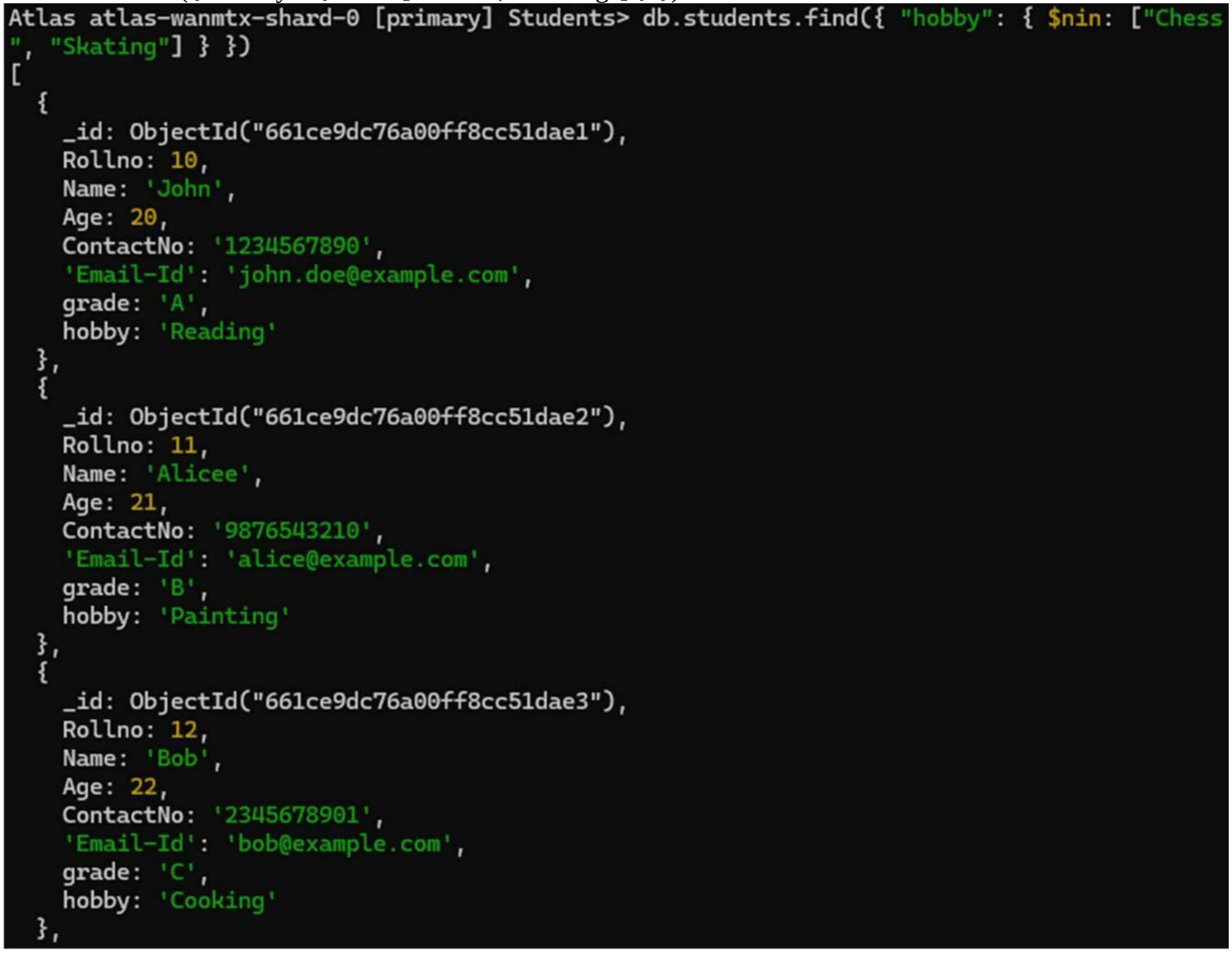
```
db.students.updateMany(
   { "Name": "Eve" },
   { $set: { "hobby": "Dancing" } }
)
```



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

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

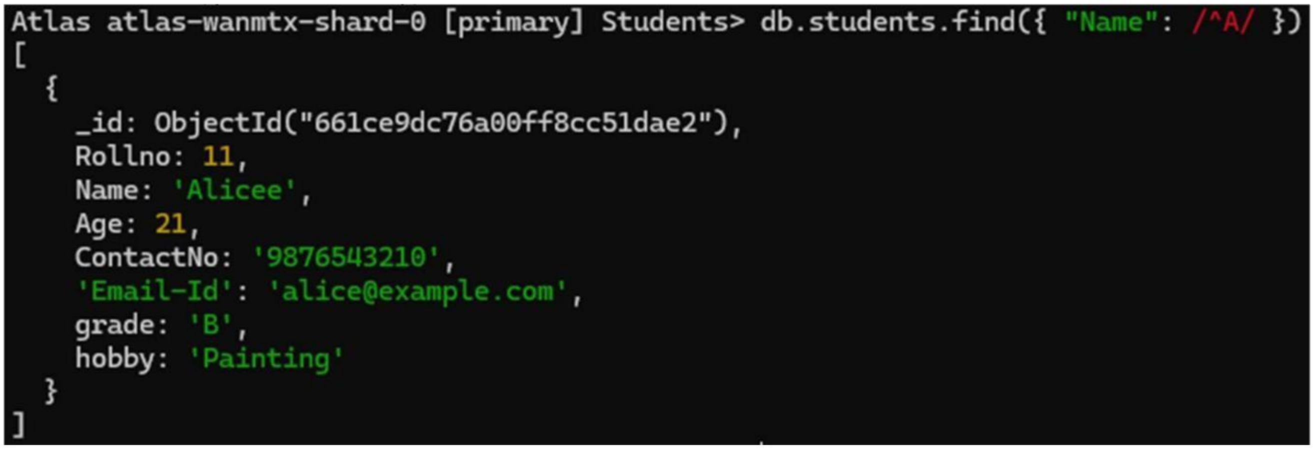
db.students.find({ "hobby": { $nin: ["Chess", "Skating"] } })



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

8. Find documents whose name begins with A

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

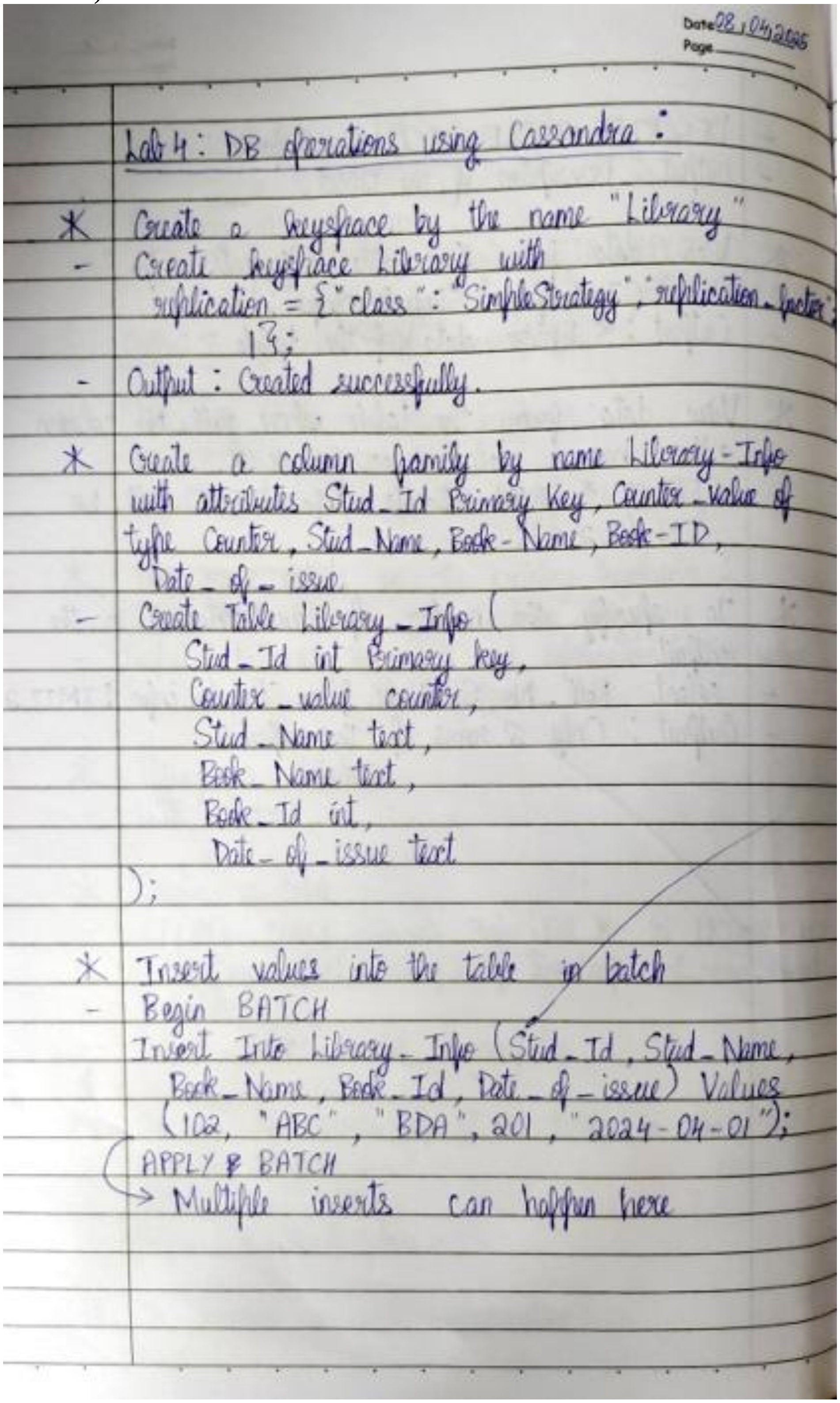


```
Atlas atlas-wanmtx-shard-0 [primary] Students> db.students.find({ "Name": /^A/ })
[
  {
    _id: ObjectId("661ce9dc76a00ff8cc51dae2"),
    Rollno: 11,
    Name: 'Alicee',
    Age: 21,
    ContactNo: '9876543210',
    'Email-Id': 'alice@example.com',
    grade: 'B',
    hobby: 'Painting'
  }
]
```

Experiment - 4

1.3.3 Question:

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



Date 08/04/2025
Page

Lab 4 : DB operations using Cassandra :

* Create a keyspace by the name "Library"

- Create keyspace Library with
  replication = {"class": "SimpleStrategy", "replication_factor":
  1};

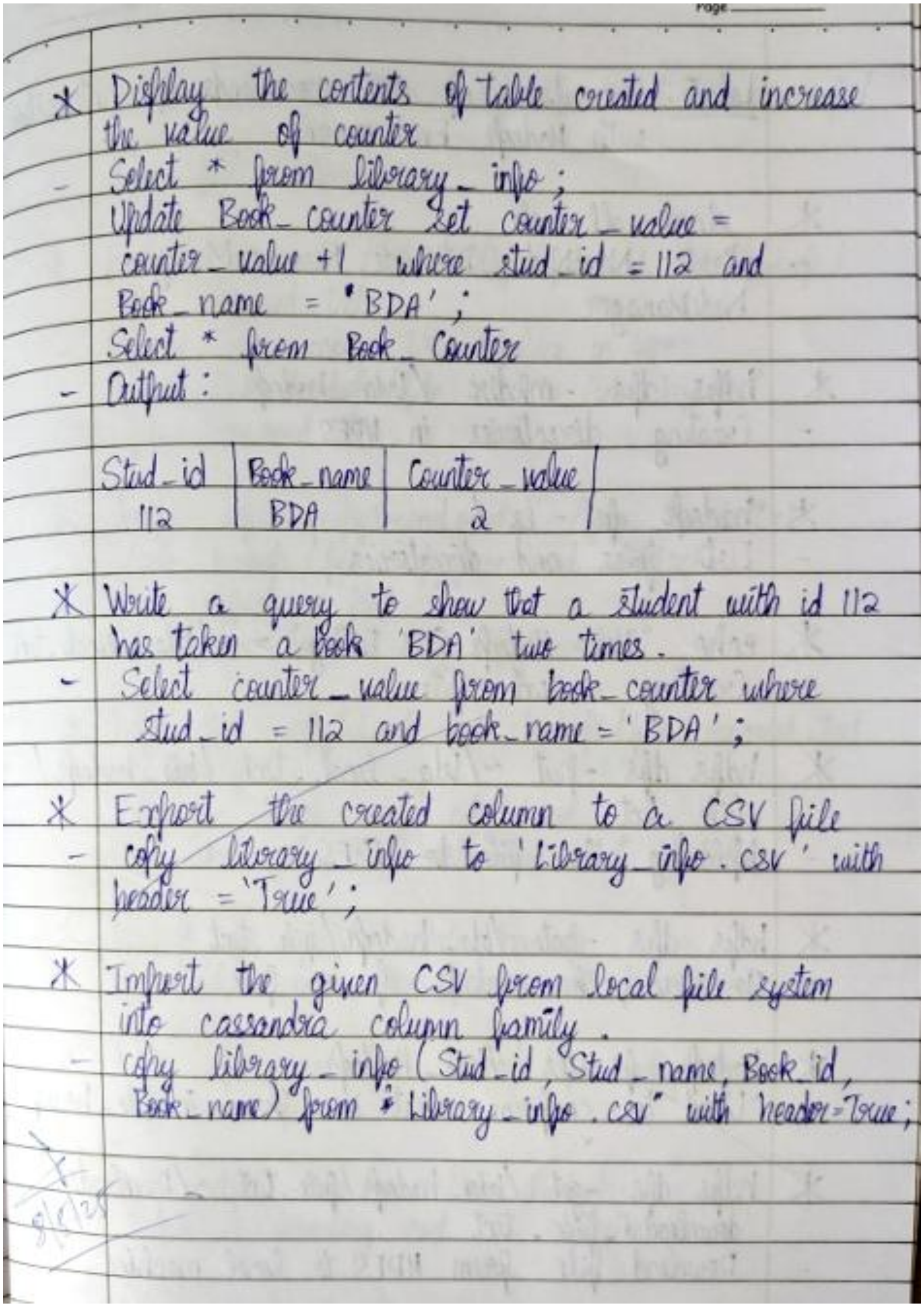
- Output : Created successfully.

* 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

- Create Table Library_Info (
      Stud_Id int Primary key,
      Counter_value counter,
      Stud_Name text,
      Book_Name text,
      Book_Id int,
      Date_of_issue text
  );

* Insert values into the table in batch

- Begin BATCH
  Insert Into Library_Info (Stud_Id, Stud_Name,
    Book_Name, Book_Id, Date_of_issue) Values
    (102, "ABC", "BDA", 201, "2024-04-01");
  APPLY ~~B~~ BATCH
  → Multiple inserts can happen here



* Display the contents of table created and increase the value of counter

```
- Select * from library_info;
  Update Book_counter Set counter_value =
  counter_value +1 where stud_id = 112 and
  Book_name = 'BDA';
  Select * from Book_Counter
```

- Output:

Stud_id	Book_name	Counter_value
112	BDA	2

* Write a query to show that a student with id 112 has taken a book 'BDA' two times.

```
- Select counter_value from book_counter where
  stud_id = 112 and book_name = 'BDA';
```

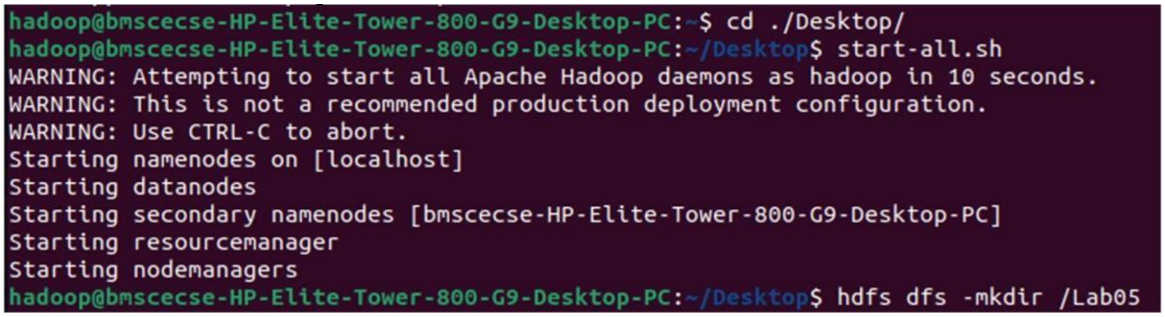
* Export the created column to a CSV file

```
- copy library_info to 'Library_info.csv' with
  header = 'True';
```

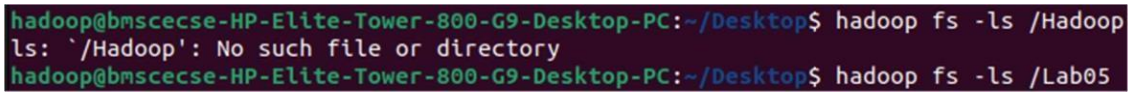
* Import the given CSV from local file system into cassandra column family.

```
- copy library_info (Stud_id, Stud_name, Book_id,
  Book_name) from "Library_info.csv" with header=True;
```

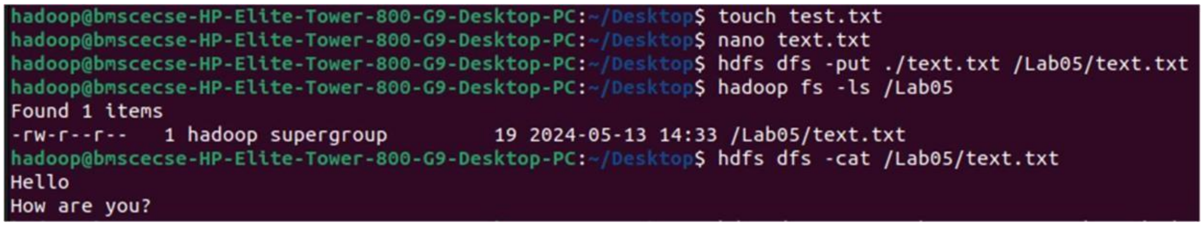
1.3.4 Code with Output:



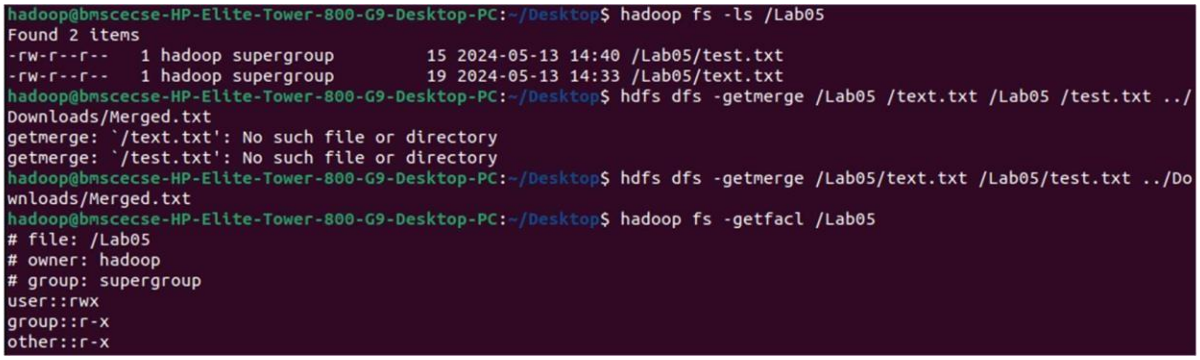
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cd ./Desktop/
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: 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
```



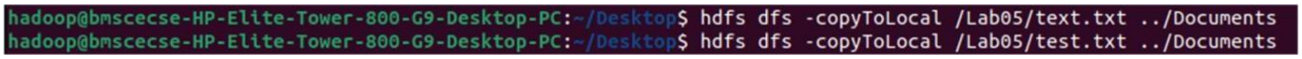
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Hadoop
ls: `/Hadoop': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -ls /Lab05
```



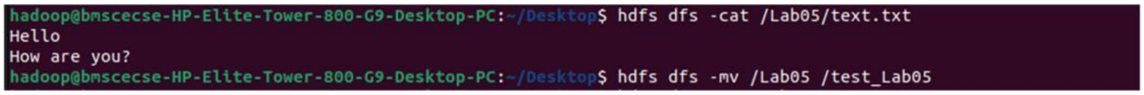
```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ touch test.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ nano text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -put ./text.txt /Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$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



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

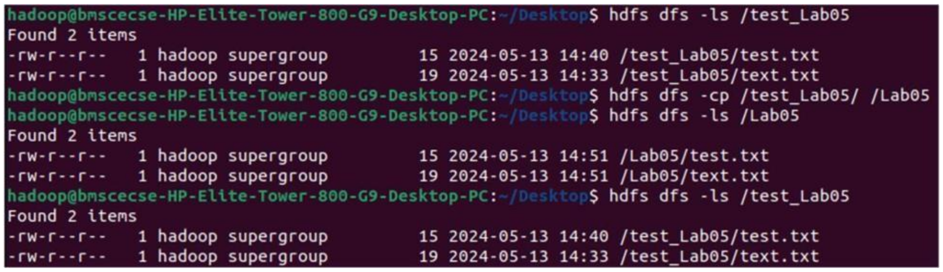


```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab05/text.txt ../Documents
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -copyToLocal /Lab05/test.txt ../Documents
```



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

`

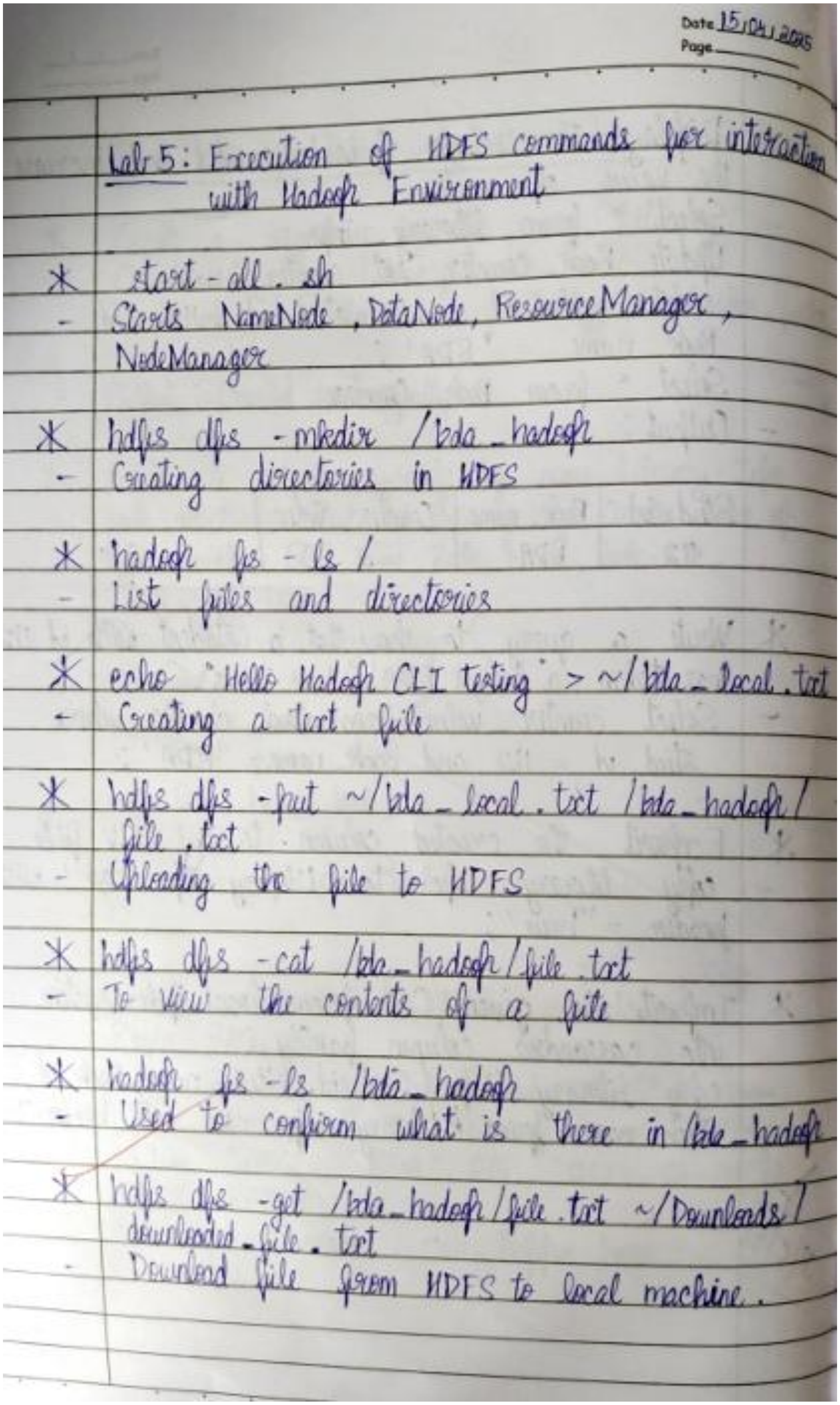


```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -ls /test_Lab05
Found 2 items
-rw-r--r--   1 hadoop supergroup         15 2024-05-13 14:40 /test_Lab05/test.txt
-rw-r--r--   1 hadoop supergroup         19 2024-05-13 14:33 /test_Lab05/text.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$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

Experiment - 6

1.3.5 Question:

Implement WordCount Program on Hadoop framework.



Date 15/04/2025
Page

Lab 5: Execution of HDFS commands for interaction with Hadoop Environment

* start-all.sh
- Starts NameNode, DataNode, ResourceManager, NodeManager

* hdfs dfs -mkdir /bda_hadoop
- Creating directories in HDFS

* hadoop fs -ls /
- List files and directories

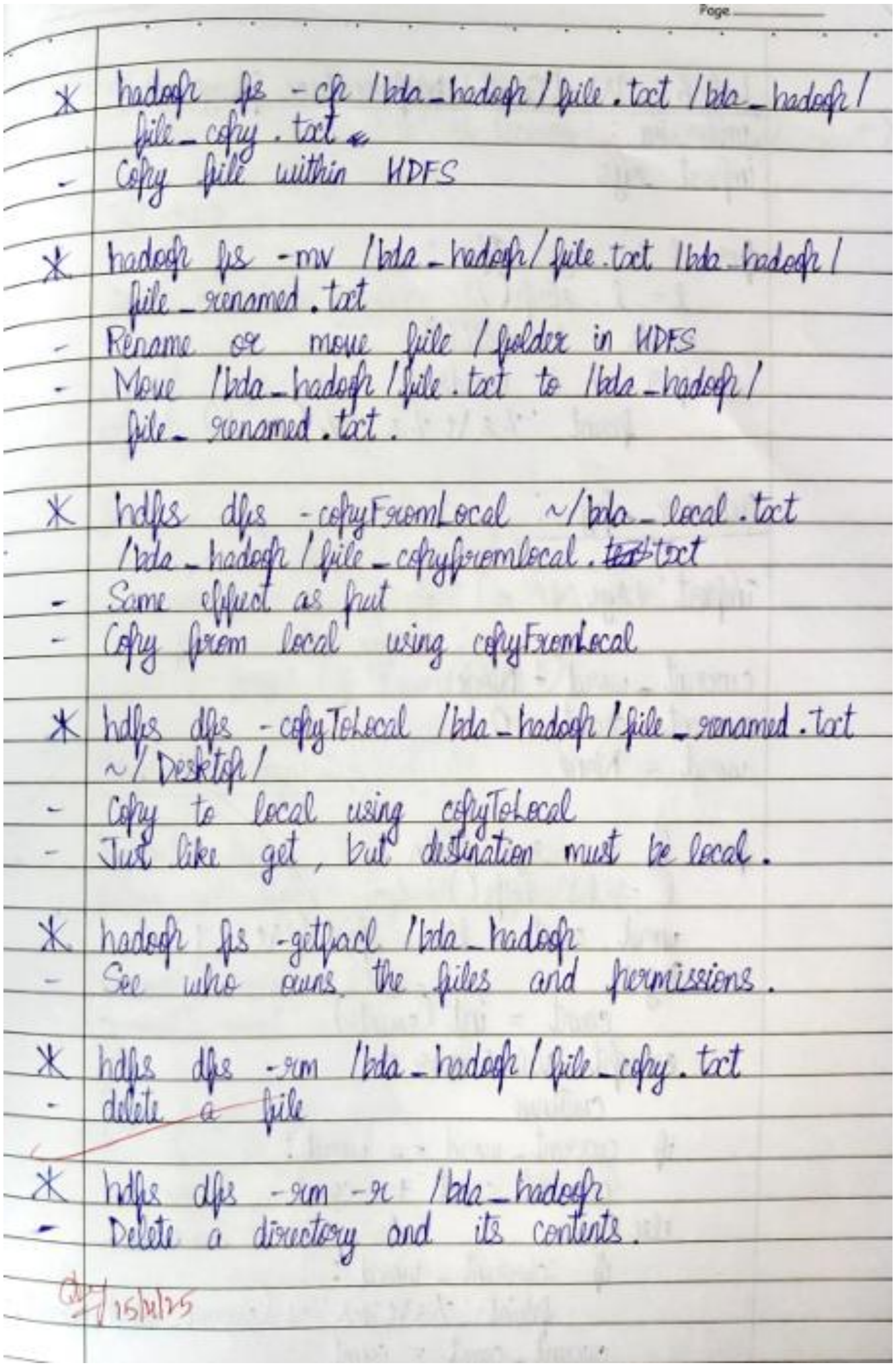
* echo "Hello Hadoop CLI testing" > ~/bda_local.txt
- Creating a text file

* hdfs dfs -put ~/bda_local.txt /bda_hadoop/file.txt
- Uploading the file to HDFS

* hdfs dfs -cat /bda_hadoop/file.txt
- To view the contents of a file

* hadoop fs -ls /bda_hadoop
- Used to confirm what is there in /bda_hadoop

* hdfs dfs -get /bda_hadoop/file.txt ~/Downloads/downloaded_file.txt
- Download file from HDFS to local machine.



Page

* hadoop fs -cp /bda_hadoop/file.txt /bda_hadoop/file_copy.txt
- Copy file within HDFS

* hadoop fs -mv /bda_hadoop/file.txt /bda_hadoop/file_renamed.txt
- Rename or move file / folder in HDFS
- Move /bda_hadoop/file.txt to /bda_hadoop/file_renamed.txt.

* hdfs dfs -copyFromLocal ~/bda_local.txt /bda_hadoop/file_copyfromlocal.~~txt~~txt
- Same effect as put
- Copy from local using copyFromLocal

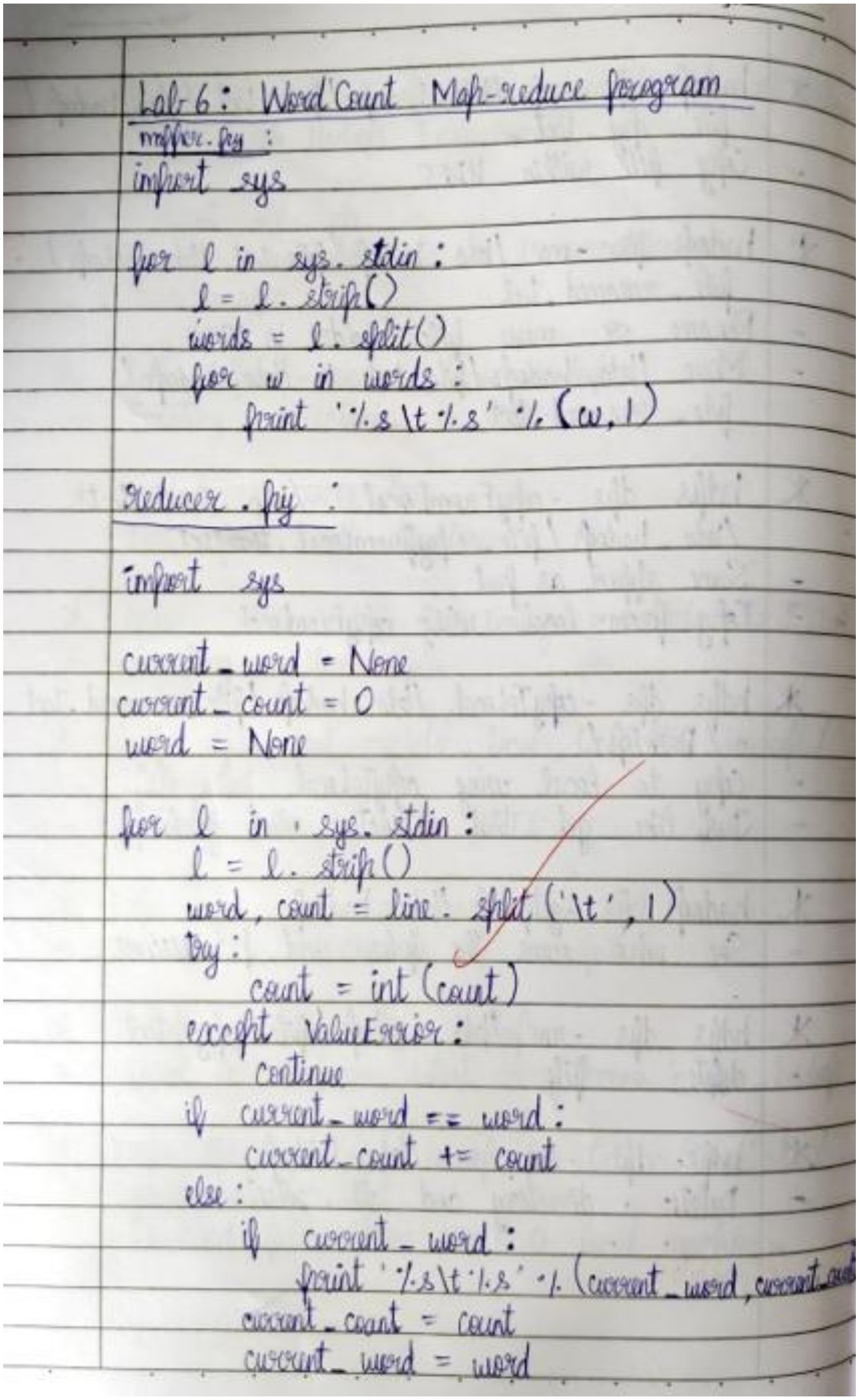
* hdfs dfs -copyToLocal /bda_hadoop/file_renamed.txt ~/Desktop/
- Copy to local using copyToLocal
- Just like get, but destination must be local.

* hadoop fs -getfacl /bda_hadoop
- See who owns the files and permissions.

* hdfs dfs -rm /bda_hadoop/file_copy.txt
- delete a file

* hdfs dfs -rm -r /bda_hadoop
- Delete a directory and its contents.

15/11/25



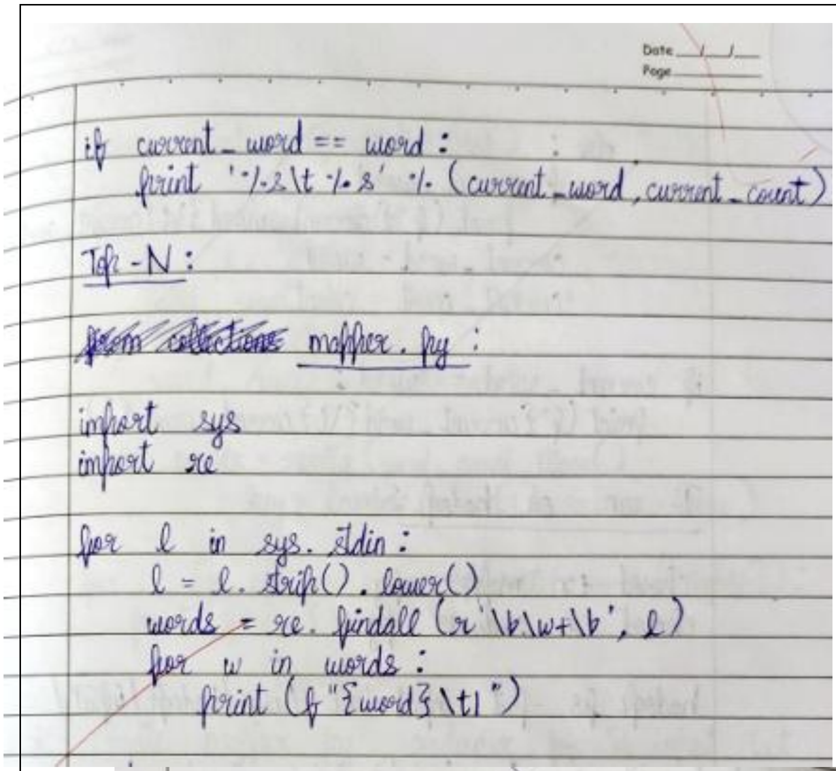
Lab 6 : Word Count Map-reduce program

mapper.py :

```
import sys

for l in sys.stdin :
    l = l.strip()
    words = l.split()
    for w in words :
        print '%s\t%s' % (w, 1)
```

reducer.py :

```
import sys

current_word = None
current_count = 0
word = None

for l in sys.stdin :
    l = l.strip()
    word, count = line.split('\t', 1)
    try :
        count = int(count)
    except ValueError :
        continue
    if current_word == word :
        current_count += count
    else :
        if current_word :
            print '%s\t%s' % (current_word, current_cou
        current_count = count
        current_word = word
```

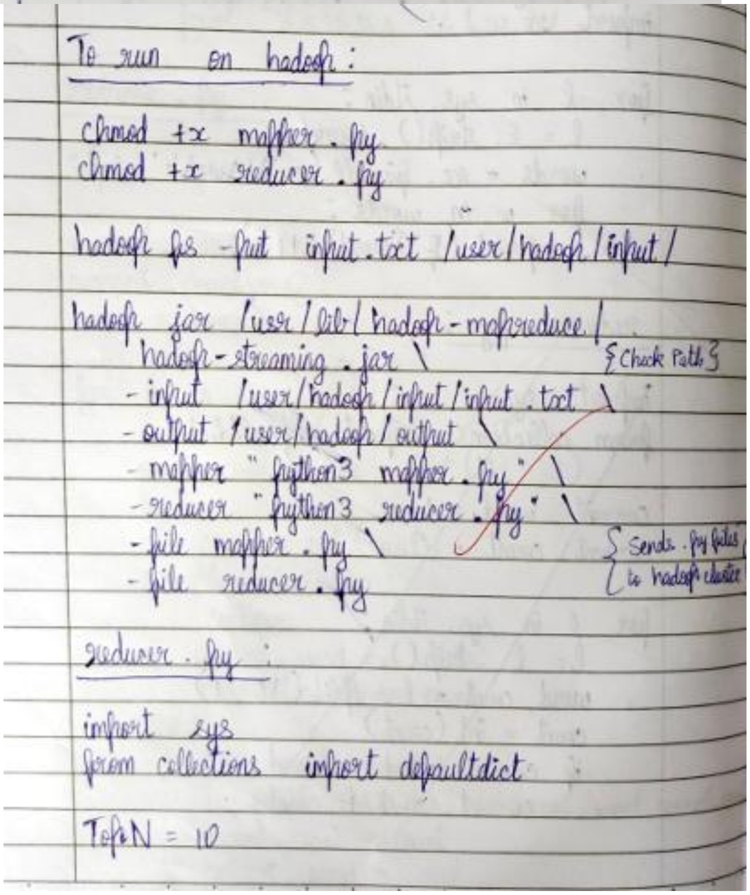


```
if current_word == word :
    print '%s\t %s' % (current_word, current_count)
```

Top-N :

~~from collections~~ mapper.py :

```
import sys
import re

for l in sys.stdin :
    l = l.strip().lower()
    words = re.findall(r'\b\w+\b', l)
    for w in words :
        print (f"{word}\t1")
```

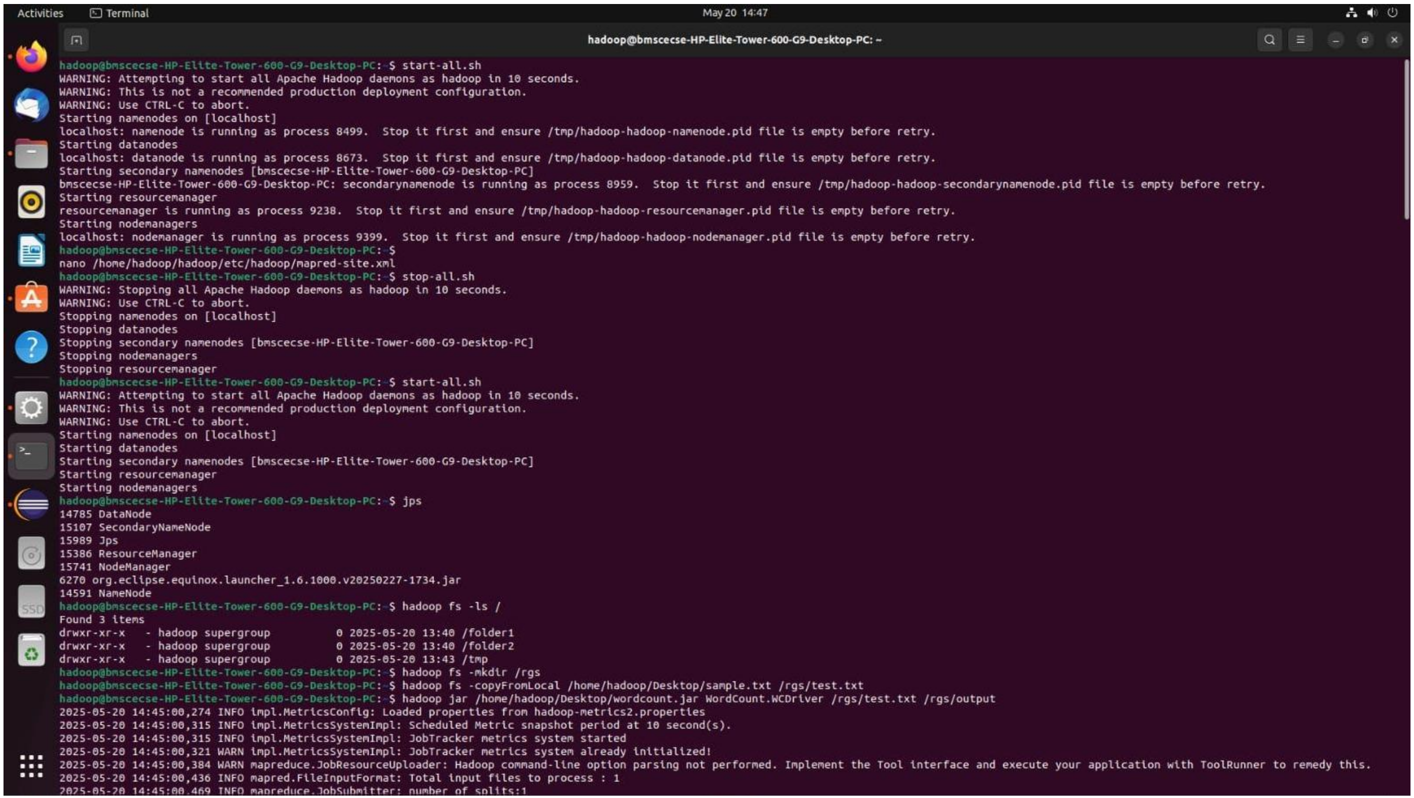


To run on hadoop :

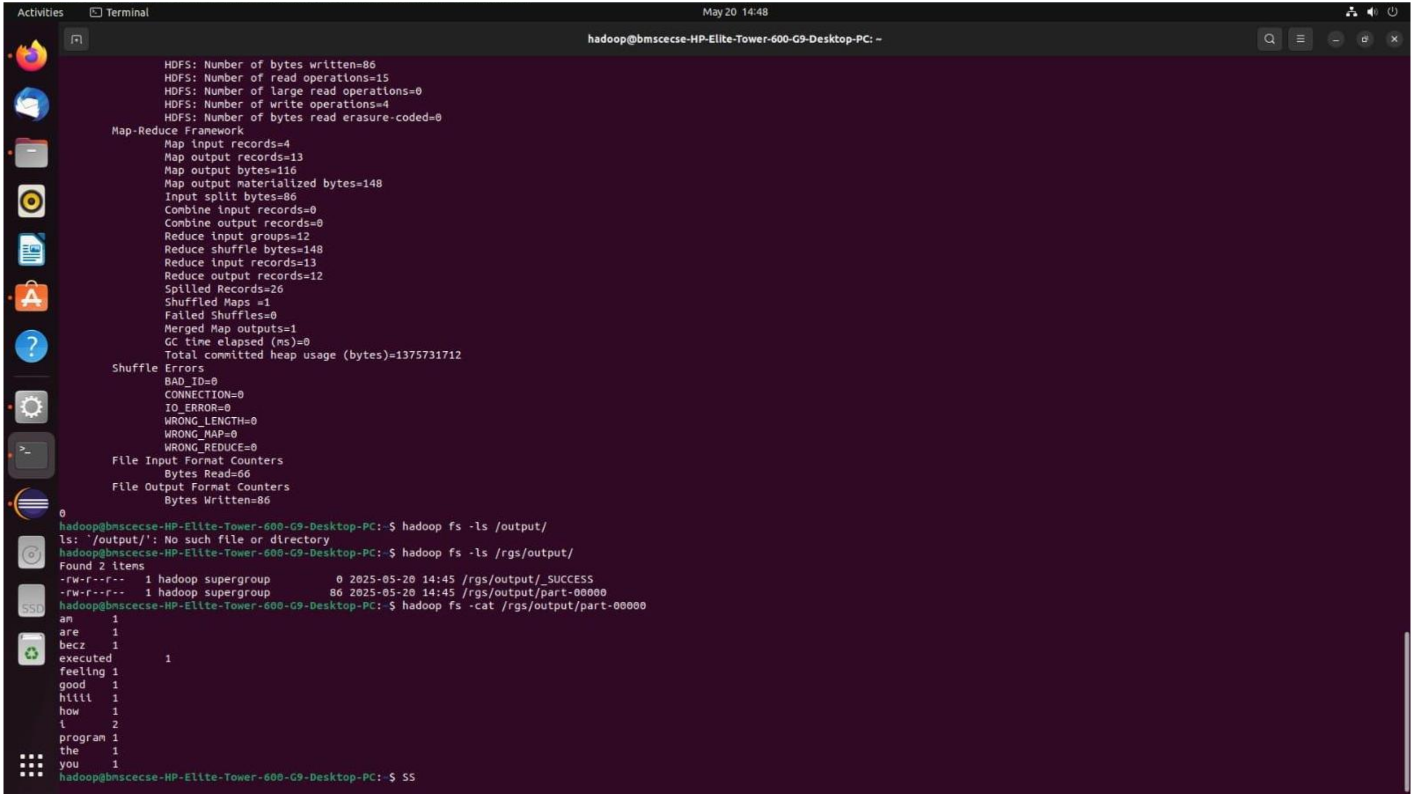
```
chmod +x mapper.py
chmod +x reducer.py

hadoop fs -put input.txt /user/hadoop/input/

hadoop jar /usr/lib/hadoop-mapreduce/
    hadoop-streaming.jar \                    {Check Path}
    -input /user/hadoop/input/input.txt \
    -output /user/hadoop/output \
    -mapper "python3 mapper.py" \
    -reducer "python3 reducer.py" \
    -file mapper.py \                          { Sends .py files
    -file reducer.py                           { to hadoop cluster
```

reducer.py :

```
import sys
from collections import defaultdict

TopN = 10
```



```
hadoop@bmscecse-HP-Elite-Tower-600-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-600-G9-Desktop-PC]
bmscecse-HP-Elite-Tower-600-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-600-G9-Desktop-PC:~$
nano /home/hadoop/hadoop/etc/hadoop/mapred-site.xml
hadoop@bmscecse-HP-Elite-Tower-600-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-600-G9-Desktop-PC]
Stopping nodemanagers
Stopping resourcemanager
hadoop@bmscecse-HP-Elite-Tower-600-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-600-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ jps
14785 DataNode
15107 SecondaryNameNode
15989 Jps
15386 ResourceManager
15741 NodeManager
6270 org.eclipse.equinox.launcher_1.6.1000.v20250227-1734.jar
14591 NameNode
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2025-05-20 13:40 /folder1
drwxr-xr-x   - hadoop supergroup          0 2025-05-20 13:40 /folder2
drwxr-xr-x   - hadoop supergroup          0 2025-05-20 13:43 /tmp
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -mkdir /rgs
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -copyFromLocal /home/hadoop/Desktop/sample.txt /rgs/test.txt
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop jar /home/hadoop/Desktop/wordcount.jar WordCount.WCDriver /rgs/test.txt /rgs/output
2025-05-20 14:45:00,274 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-20 14:45:00,315 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-05-20 14:45:00,315 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-20 14:45:00,321 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2025-05-20 14:45:00,384 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2025-05-20 14:45:00,436 INFO mapred.FileInputFormat: Total input files to process : 1
2025-05-20 14:45:00,469 INFO mapreduce.JobSubmitter: number of splits:1
```



```
                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=116
                Map output materialized bytes=148
                Input split bytes=86
                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-600-G9-Desktop-PC:~$ hadoop fs -ls /output/
ls: `/output/': No such file or directory
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -ls /rgs/output/
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2025-05-20 14:45 /rgs/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup         86 2025-05-20 14:45 /rgs/output/part-00000
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ hadoop fs -cat /rgs/output/part-00000
am      1
are     1
becz    1
executed        1
feeling 1
good    1
hiiii   1
how     1
i       2
program 1
the     1
you     1
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ SS
```

1.3.6 Code with Output:

Mapper Code:

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

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

Reducer Code:

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

Driver Code: WCDriver Java Class file.

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

1.4 Experiment - 7

1.4.1 Question:

From the following link extract the weather data:

Create a Map Reduce program to:

c) Find average temperature for each year from NCDC data set.

d) Find the mean max temperature for every month.

1.4.2 Code with Output:

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

AverageDriver:

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

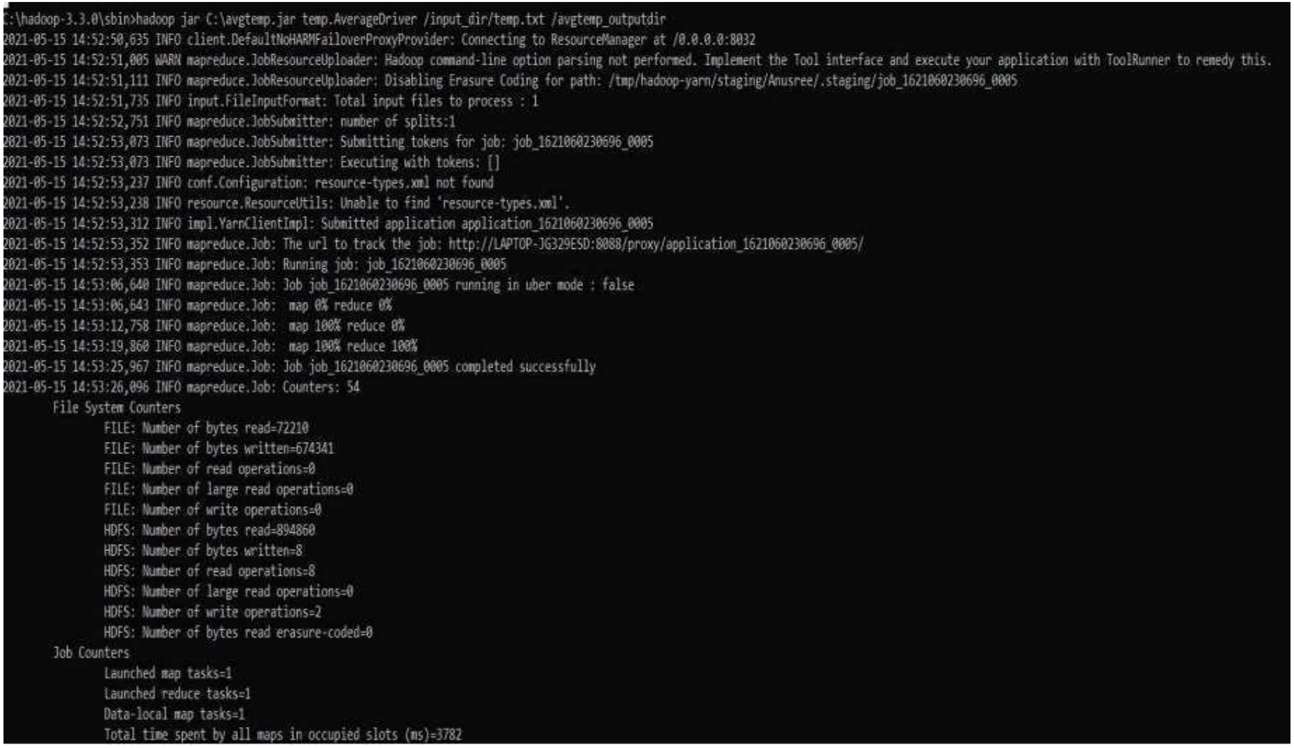
AverageMapper:

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

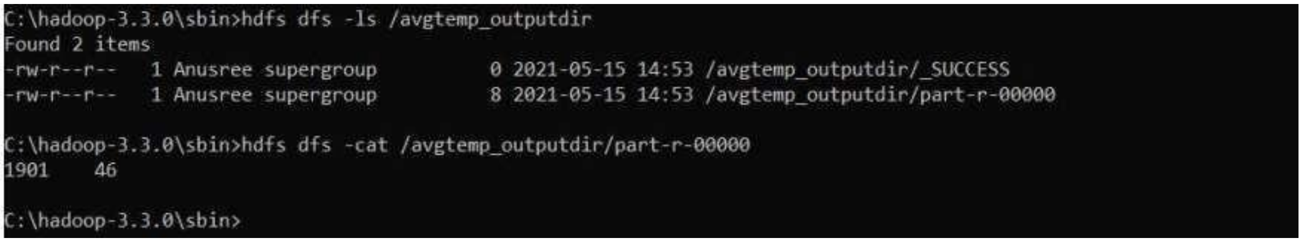
```
temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(year), new IntWritable(temperature));
}
}
```

AverageReducer:

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



```
:\hadoop-3.3.0\sbin>hadoop jar C:\avgtemp.jar temp.AverageDriver /input_dir/temp.txt /avgtemp_outputdir
2021-05-15 14:52:50,635 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-15 14:52:51,005 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-15 14:52:51,111 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621060230696_0005
2021-05-15 14:52:51,735 INFO input.FileInputFormat: Total input files to process : 1
2021-05-15 14:52:52,751 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621060230696_0005
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-15 14:52:53,237 INFO conf.Configuration: resource-types.xml not found
2021-05-15 14:52:53,238 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-15 14:52:53,312 INFO impl.YarnClientImpl: Submitted application application_1621060230696_0005
2021-05-15 14:52:53,352 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621060230696_0005/
2021-05-15 14:52:53,353 INFO mapreduce.Job: Running job: job_1621060230696_0005
2021-05-15 14:53:06,640 INFO mapreduce.Job: Job job_1621060230696_0005 running in uber mode : false
2021-05-15 14:53:06,643 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-15 14:53:12,758 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-15 14:53:19,860 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-15 14:53:25,967 INFO mapreduce.Job: Job job_1621060230696_0005 completed successfully
2021-05-15 14:53:26,096 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=72210
                FILE: Number of bytes written=674341
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=894860
                HDFS: Number of bytes written=8
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3782
```



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

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

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

b) find the mean max temperature for every month

MeanMaxDriver.class

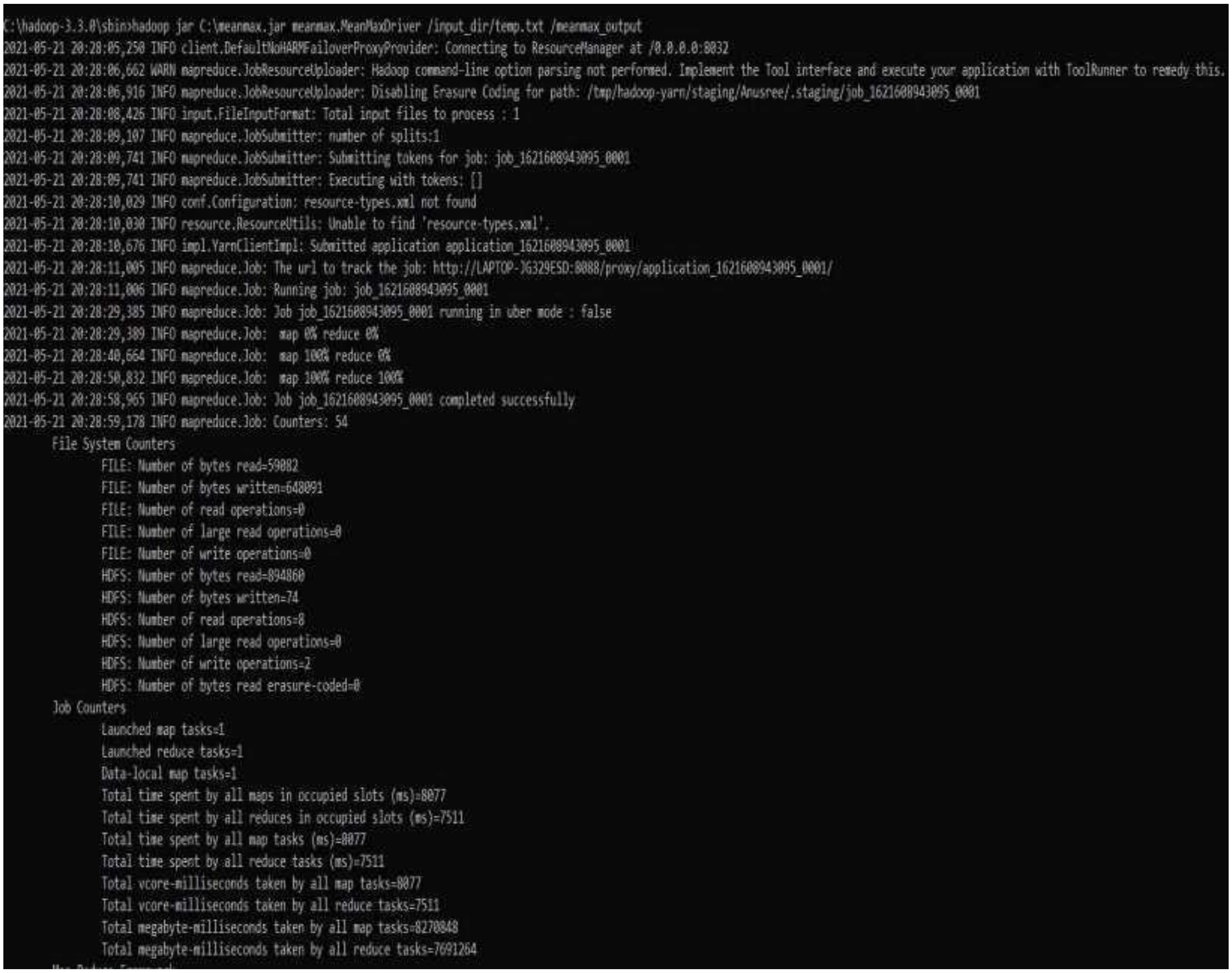
```
package meanmax;
import  org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public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

MeanMaxMapper.class

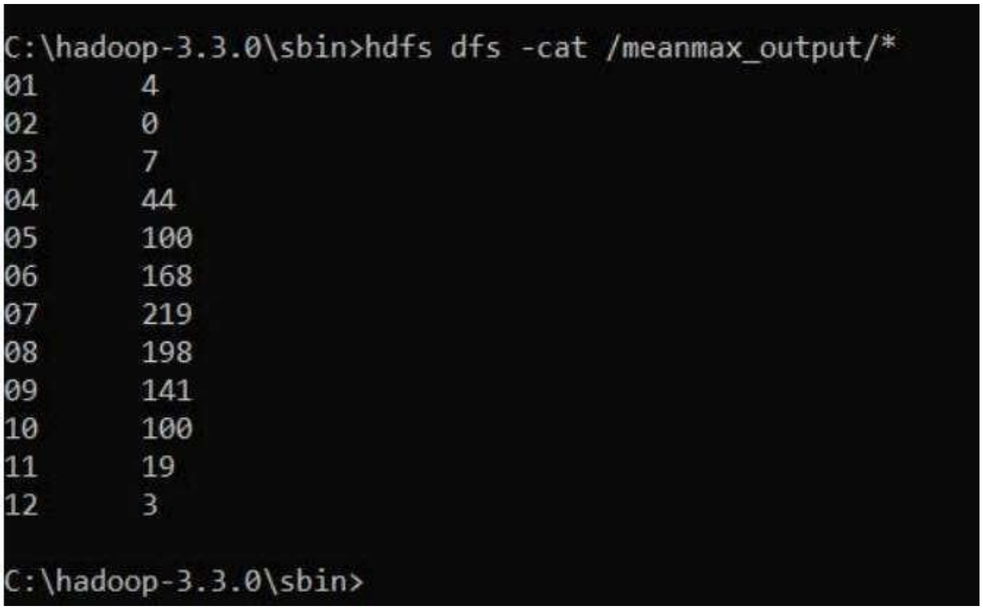
```
package meanmax;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class MeanMaxMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text,
IntWritable>.Context context) throws IOException, InterruptedException {
int temperature;
String line = value.toString();
String month = line.substring(19, 21);
if (line.charAt(87) == '+') {
temperature = Integer.parseInt(line.substring(88, 92));
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999 && quality.matches("[01459]"))
context.write(new Text(month), new IntWritable(temperature));
}
}
```

MeanMaxReducer.class

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



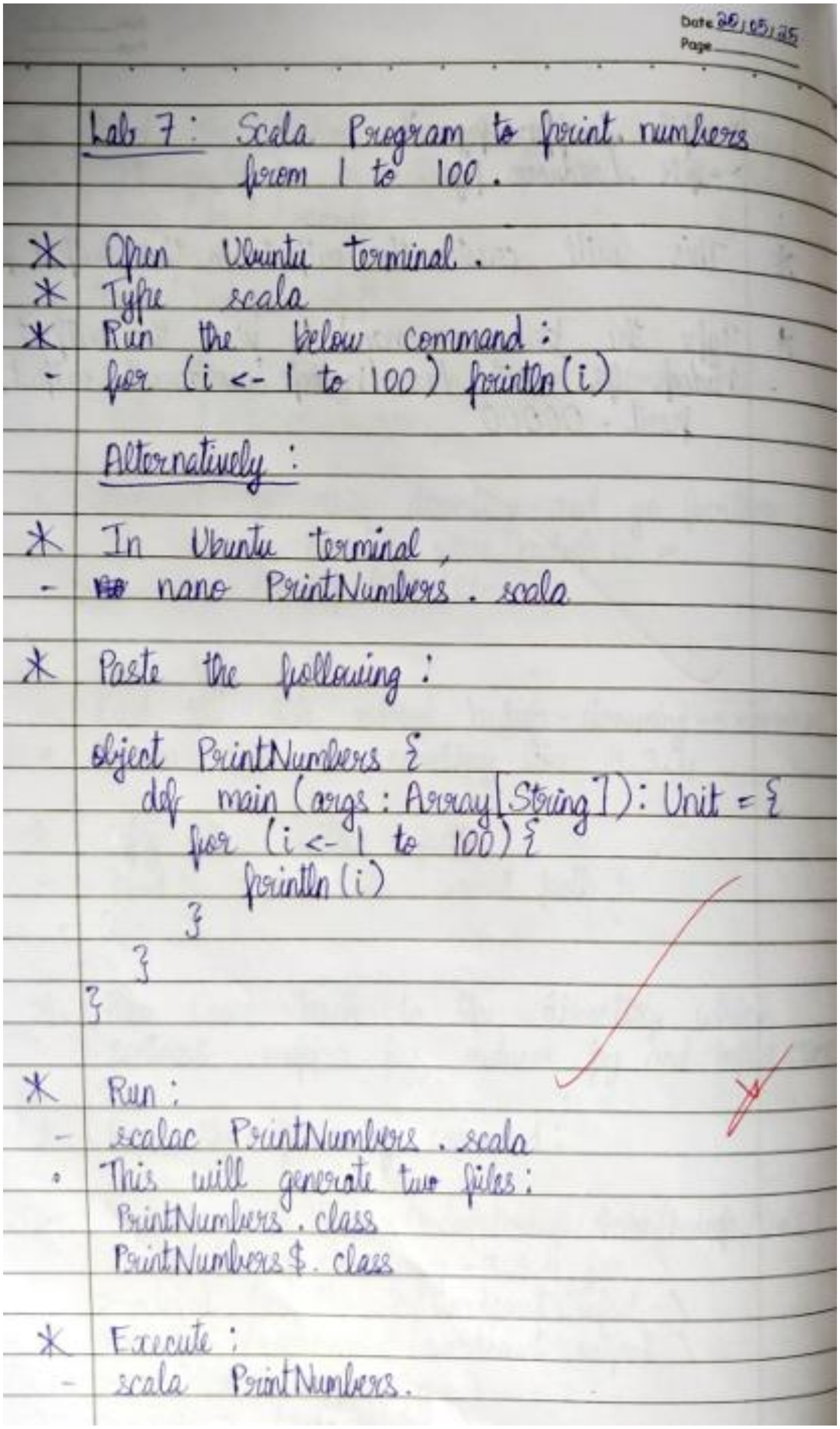
```
C:\hadoop-3.3.0\sbin>hadoop jar C:\meanmax.jar meanmax.MeanMaxDriver /input_dir/temp.txt /meanmax_output
2021-05-21 20:28:05,250 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-21 20:28:06,662 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-21 20:28:06,916 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621608943095_0001
2021-05-21 20:28:08,426 INFO input.FileInputFormat: Total input files to process : 1
2021-05-21 20:28:09,107 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621608943095_0001
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-21 20:28:10,029 INFO conf.Configuration: resource-types.xml not found
2021-05-21 20:28:10,030 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-21 20:28:10,676 INFO impl.YarnClientImpl: Submitted application application_1621608943095_0001
2021-05-21 20:28:11,005 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621608943095_0001/
2021-05-21 20:28:11,006 INFO mapreduce.Job: Running job: job_1621608943095_0001
2021-05-21 20:28:29,385 INFO mapreduce.Job: Job job_1621608943095_0001 running in uber mode : false
2021-05-21 20:28:29,389 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-21 20:28:40,664 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-21 20:28:50,832 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-21 20:28:58,965 INFO mapreduce.Job: Job job_1621608943095_0001 completed successfully
2021-05-21 20:28:59,178 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=59082
                FILE: Number of bytes written=648091
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=894860
                HDFS: Number of bytes written=74
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=8077
                Total time spent by all reduces in occupied slots (ms)=7511
                Total time spent by all map tasks (ms)=8077
                Total time spent by all reduce tasks (ms)=7511
                Total vcore-milliseconds taken by all map tasks=8077
                Total vcore-milliseconds taken by all reduce tasks=7511
                Total megabyte-milliseconds taken by all map tasks=8270848
                Total megabyte-milliseconds taken by all reduce tasks=7691264
```



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

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

Experiment – 8

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



Date 20/05/25
Page

Lab 7: Scala Program to print numbers from 1 to 100.

* Open Ubuntu terminal.
* Type scala
* Run the below command:

```
- for (i <- 1 to 100) println(i)
```

Alternatively:

* In Ubuntu terminal,

```
- nano PrintNumbers.scala
```

* Paste the following:

```
object PrintNumbers {
    def main (args : Array[String]): Unit = {
        for (i <- 1 to 100) {
            println (i)
        }
    }
}
```

* Run:

```
- scalac PrintNumbers.scala
```

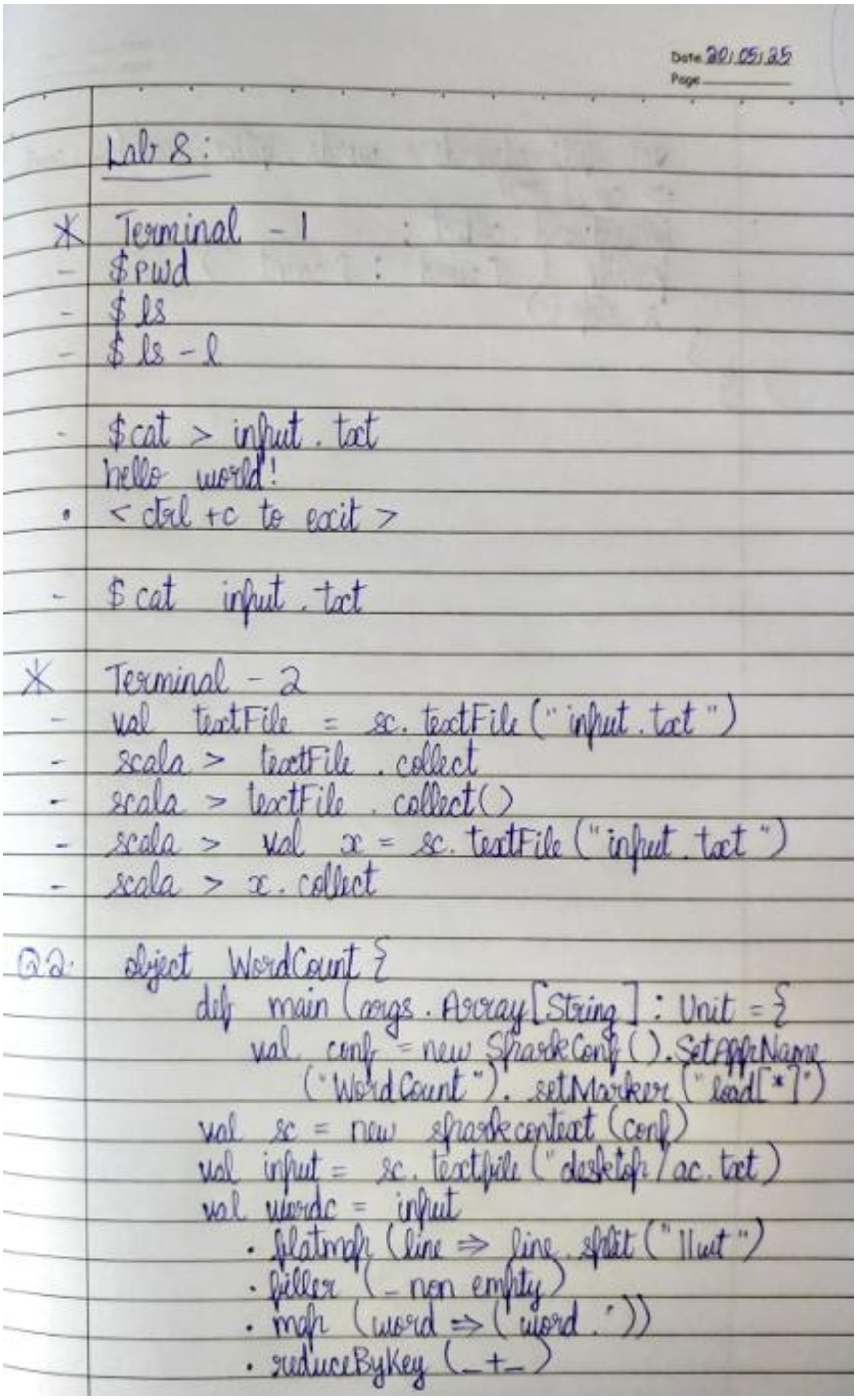
. This will generate two files:
PrintNumbers.class
PrintNumbers$.class

* Execute:

```
- scala PrintNumbers.
```

Experiment – 9

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



Date 20.05.25
Page

Lab 8:

* Terminal - 1

```
- $pwd
- $ls
- $ls -l

- $cat > input.txt
  hello world!
• < ctrl +c to exit >

- $cat input.txt
```

* Terminal - 2

```
- val textFile = sc.textFile("input.txt")
- scala > textFile.collect
- scala > textFile.collect()
- scala > val x = sc.textFile("input.txt")
- scala > x.collect
```

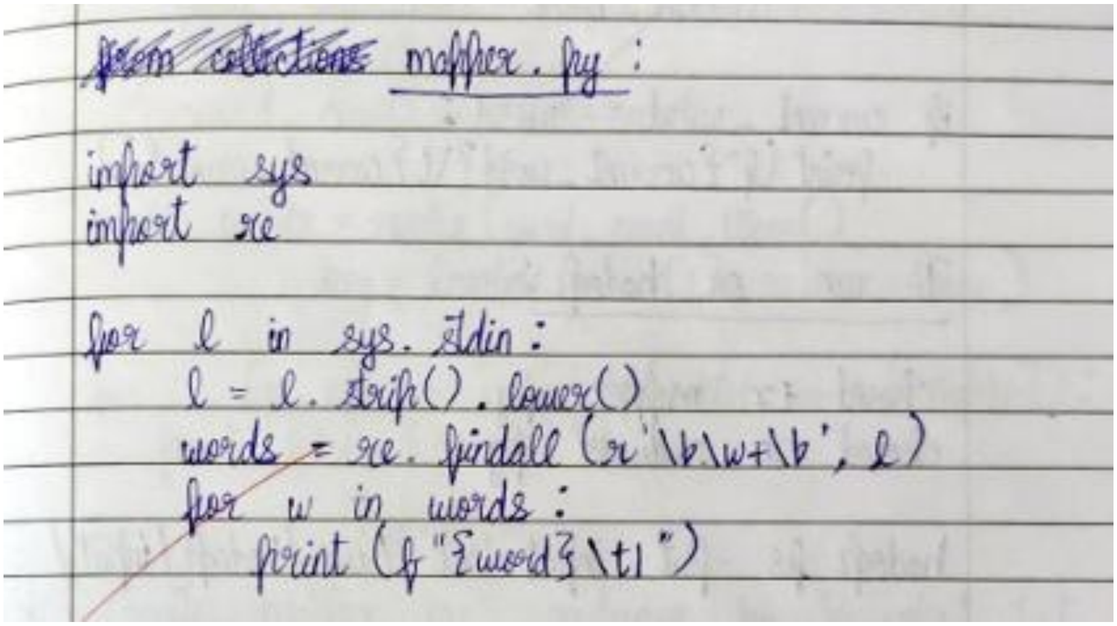
Q2:

```
object WordCount {
      def main (args . Array[String] : Unit = {
          val conf = new SparkConf().SetAppName
              ("WordCount"). setMarker ("load[*]")
      val sc = new sparkcontext (conf)
      val input = sc. textfile ("desktop/ac.txt)
      val wordc = input
          . flatmap (line => line. split ("11ut")
          . filler (_ non empty)
          . map (word => (word . '))
          . reduceByKey (_+_)
```

Experiment - 9

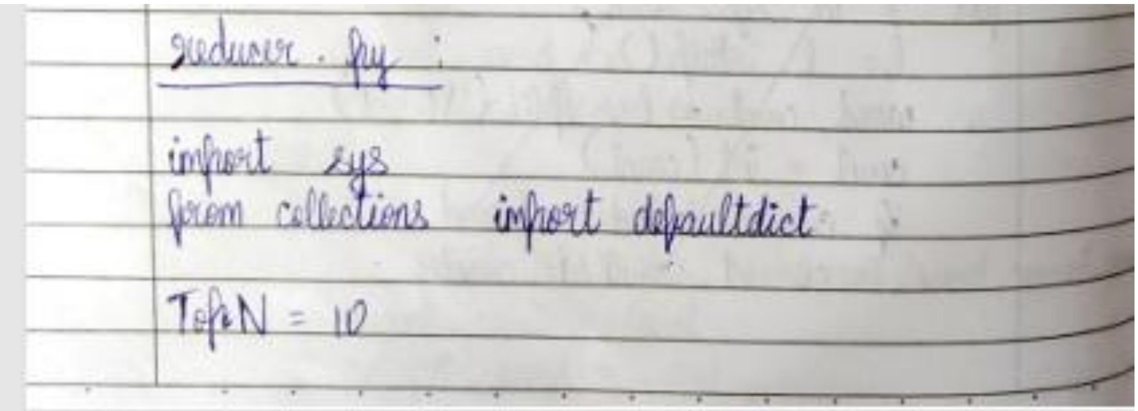
1.4.3 Question:

For a given Text file, Create a Map Reduce program to sort the content in an alphabetic order listing only top 10 maximum occurrences of words.



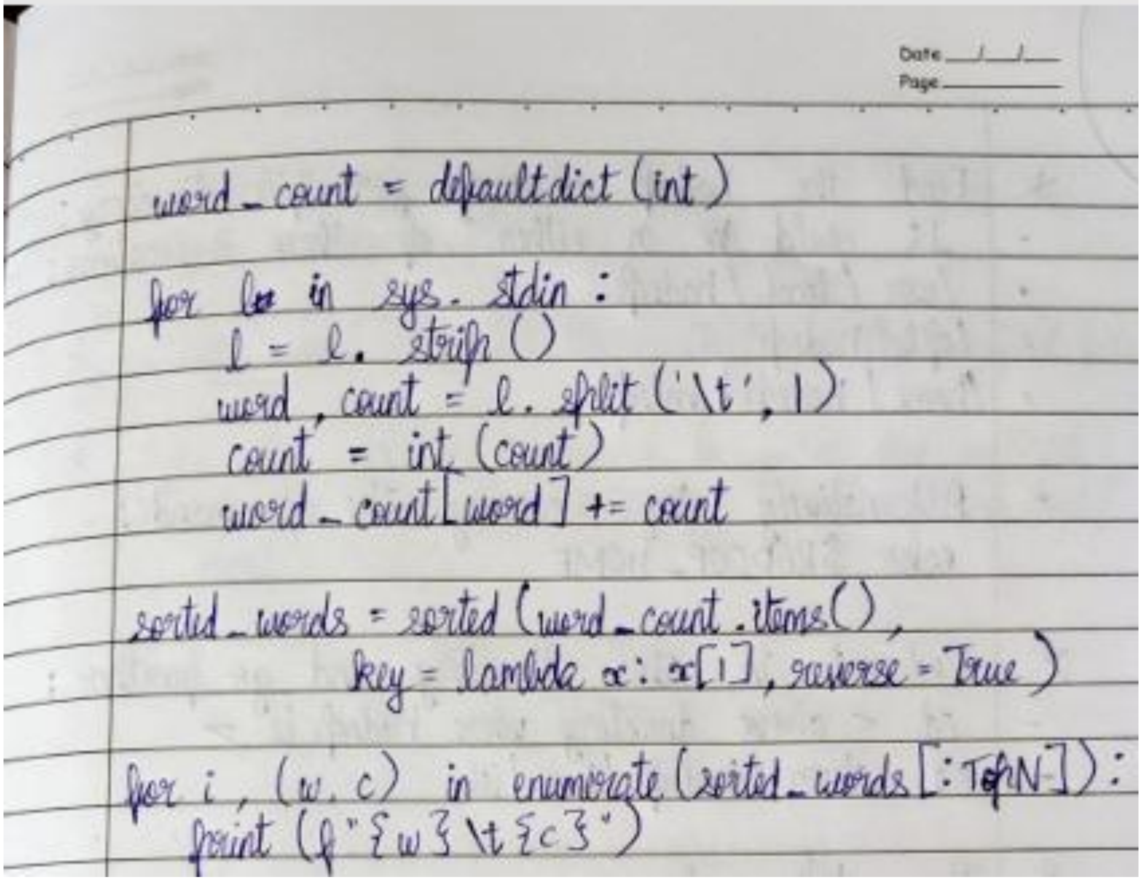
~~from collections~~ mapper.py :

```
import sys
import re

for l in sys. stdin :
    l = l. strip() . lower()
    words = re. findall (r'\b\w+\b', l)
    for w in words :
        print (f"{word}\t1")
```



reducer . py :

```
import sys
from collections import defaultdict

TopN = 10
```



Date ___/___/___
Page ______

```
word_count = defaultdict(int)

for l in sys.stdin:
    l = l.strip()
    word, count = l.split('\t', 1)
    count = int(count)
    word_count[word] += count

sorted_words = sorted(word_count.items(),
            key = lambda x: x[1], reverse = True)

for i, (w, c) in enumerate(sorted_words[:TopN]):
    print(f"{w}\t{c}")
```

1.4.4 Code with Output:

Driver-TopN.class

```
package samples.topn;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class TopN {
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
String[] otherArgs = (new GenericOptionsParser(conf, args)).getRemainingArgs();
if (otherArgs.length != 2) {
System.err.println("Usage: TopN <in> <out>");
System.exit(2);
}
Job job = Job.getInstance(conf);
job.setJobName("Top N");
job.setJarByClass(TopN.class);
job.setMapperClass(TopNMapper.class);
job.setReducerClass(TopNReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
private static final IntWritable one = new IntWritable(1);
private Text word = new Text();
private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\""]";
public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
StringTokenizer itr = new StringTokenizer(cleanLine);
while (itr.hasMoreTokens()) {
this.word.set(itr.nextToken().trim());
context.write(this.word, one);
}
}
}
}
```

TopNCombiner.class

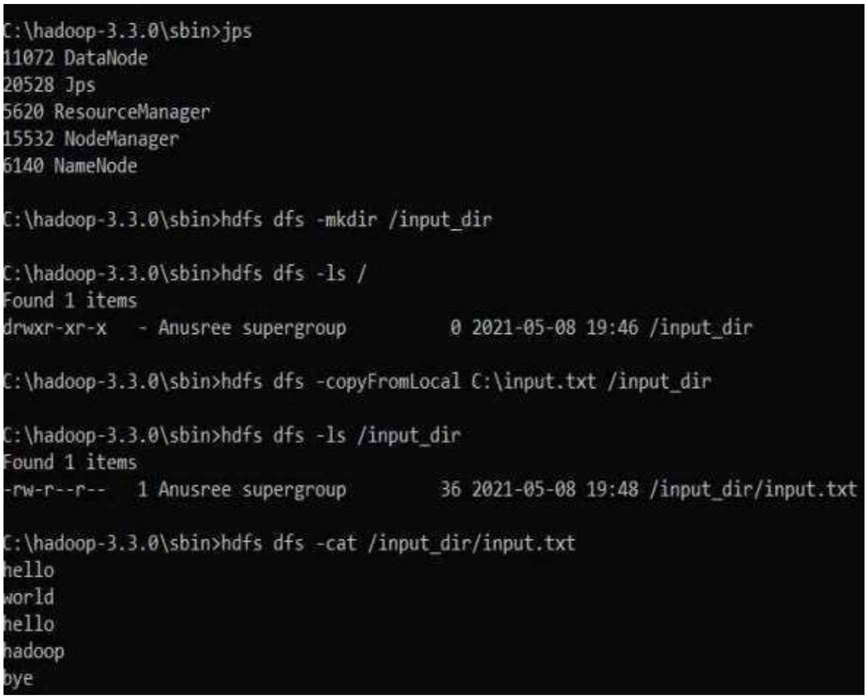
```
package samples.topn;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class TopNCombiner extends Reducer<Text, IntWritable, Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values)
sum += val.get();
context.write(key, new IntWritable(sum));
}
}
```

TopNMapper.class

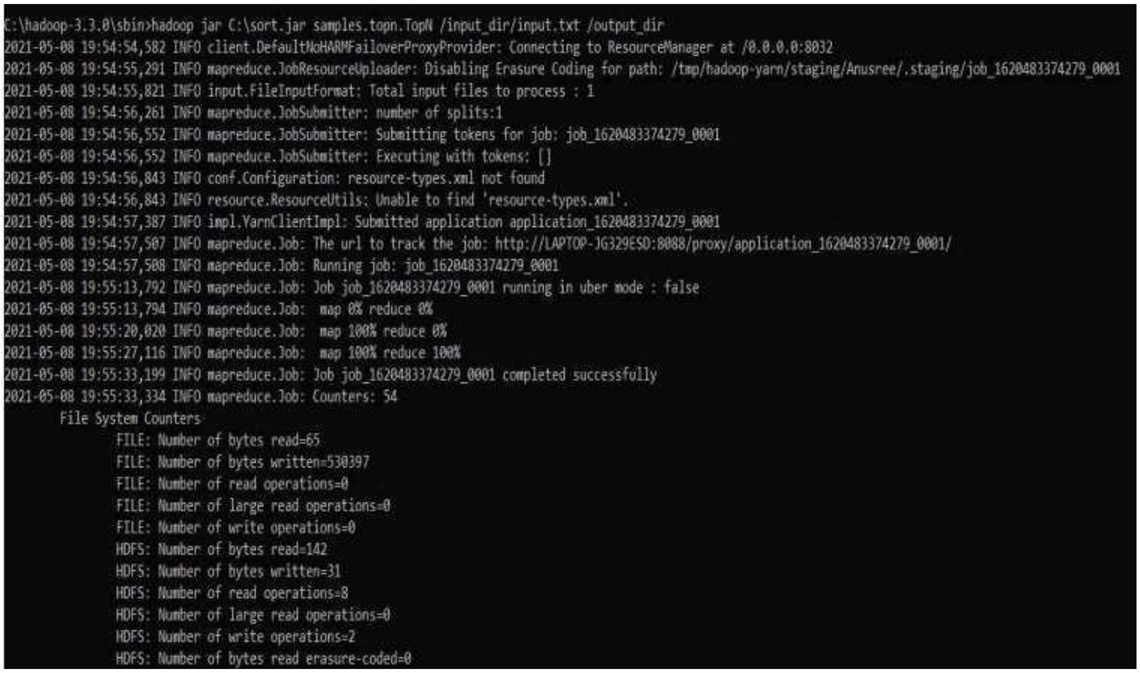
```
package samples.topn;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
private static final IntWritable one = new IntWritable(1);
private Text word = new Text();
private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";
public void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context
context) throws IOException, InterruptedException {
String cleanLine = value.toString().toLowerCase().replaceAll(this.tokens, " ");
StringTokenizer itr = new StringTokenizer(cleanLine);
while (itr.hasMoreTokens()) {
this.word.set(itr.nextToken().trim());
context.write(this.word, one);
}
}
}
```

TopNReducer.class

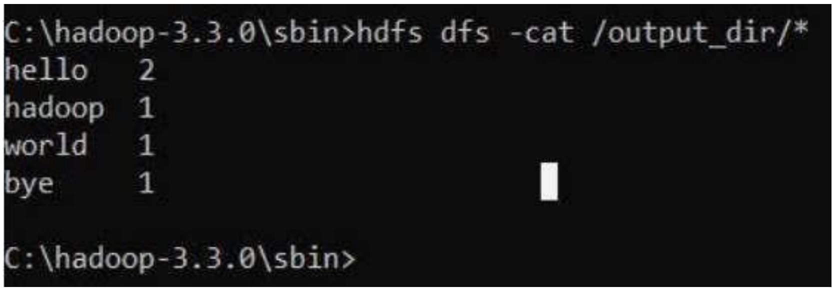
```
package samples.topn;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;
public class TopNReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
private Map<Text, IntWritable> countMap = new HashMap<>();
public void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable,
Text, IntWritable>.Context context) throws IOException, InterruptedException {
int sum = 0;
for (IntWritable val : values)
sum += val.get();
this.countMap.put(new Text(key), new IntWritable(sum));
}
protected void cleanup(Reducer<Text, IntWritable, Text, IntWritable>.Context context)
throws IOException, InterruptedException {
Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(this.countMap);
int counter = 0;
for (Text key : sortedMap.keySet()) {
if (counter++ == 20)
break;
context.write(key, sortedMap.get(key));
}
}
}
```



```
C:\hadoop-3.3.0\sbin>jps
11072 DataNode
20528 Jps
5620 ResourceManager
15532 NodeManager
6140 NameNode

C:\hadoop-3.3.0\sbin>hdfs dfs -mkdir /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - Anusree supergroup          0 2021-05-08 19:46 /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -copyFromLocal C:\input.txt /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /input_dir
Found 1 items
-rw-r--r--   1 Anusree supergroup         36 2021-05-08 19:48 /input_dir/input.txt

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/input.txt
hello
world
hello
hadoop
bye
```



```
C:\hadoop-3.3.0\sbin>hadoop jar C:\sort.jar samples.topn.TopN /input_dir/input.txt /output_dir
2021-05-08 19:54:54,582 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-08 19:54:55,291 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1620483374279_0001
2021-05-08 19:54:55,821 INFO input.FileInputFormat: Total input files to process : 1
2021-05-08 19:54:56,261 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1620483374279_0001
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-08 19:54:56,843 INFO conf.Configuration: resource-types.xml not found
2021-05-08 19:54:56,843 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-08 19:54:57,387 INFO impl.YarnClientImpl: Submitted application application_1620483374279_0001
2021-05-08 19:54:57,507 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1620483374279_0001/
2021-05-08 19:54:57,508 INFO mapreduce.Job: Running job: job_1620483374279_0001
2021-05-08 19:55:13,792 INFO mapreduce.Job: Job job_1620483374279_0001 running in uber mode : false
2021-05-08 19:55:13,794 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-08 19:55:20,020 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-08 19:55:27,116 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-08 19:55:33,199 INFO mapreduce.Job: Job job_1620483374279_0001 completed successfully
2021-05-08 19:55:33,334 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=65
                FILE: Number of bytes written=530397
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=142
                HDFS: Number of bytes written=31
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
```



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

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