

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

Big Data Analytics

Submitted by

Monish P (1BM22CS163)

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 **Monish P (1BM22CS163)**, 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,
Assistant professor,
Department of CSE
BMSCE, Bengaluru

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

Index Sheet

Sl. No.	Experiment Title	Page No.
1	MongoDB- CRUD Operations Demonstration (Practice and Self Study)	1
2	Perform the following DB operations using Cassandra.	4
3	Perform the following DB operations using Cassandra	7
4	Execution of HDFS Commands for interaction with Hadoop Environment.	9
5	Implement Wordcount program on Hadoop framework	11
6	Create a MapReduce program to find average temperature for each year from data set. find the mean max temperature for every month	16
7	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.	20
8	Write a Scala program to print numbers from 1 to 100 using for loop.	22
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.	24

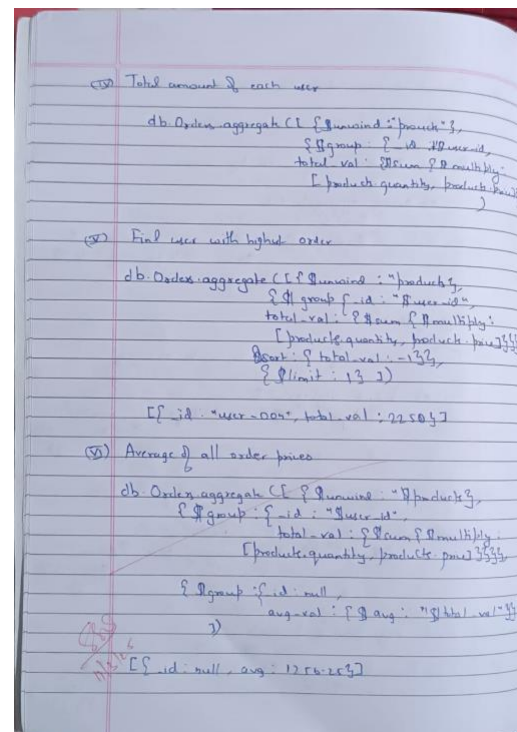
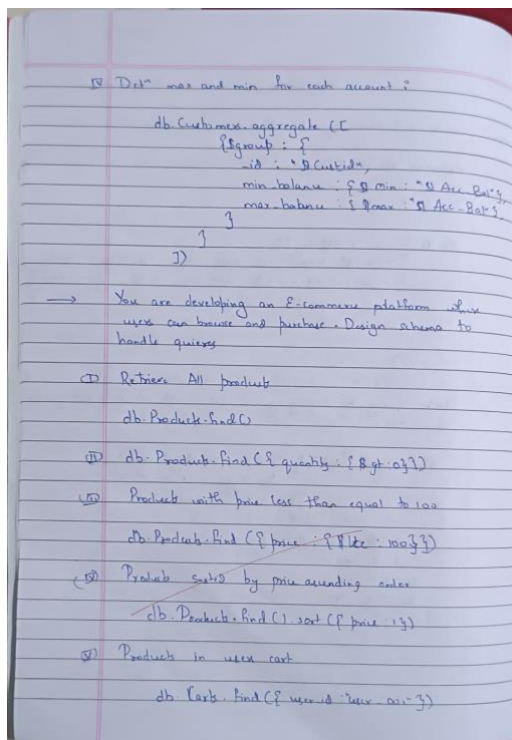
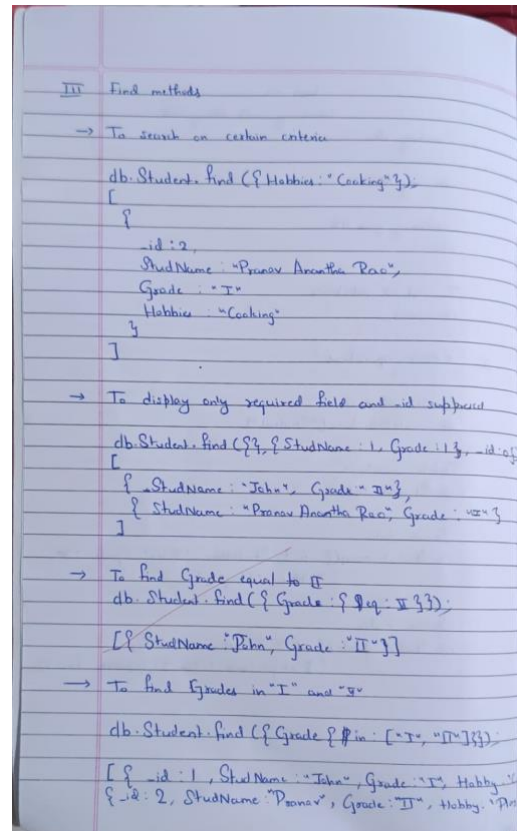
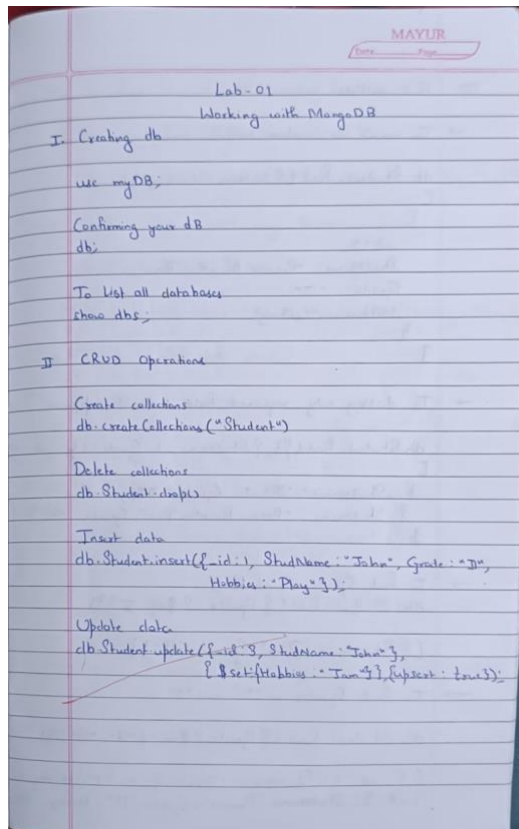
Course Outcome

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

MongoDB- CRUD Operations Demonstration (Practice and Self Study)



Code Outputs:

```
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")
  }
}
```

```
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: 'Alice',
    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'
  },
]
```

```
...
}
...
});
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67f9b0712f8b888f2fa0222'),
    '1': ObjectId('67f9b0712f8b888f2fa0222'),
    '2': ObjectId('67f9b0712f8b888f2fa0222'),
    '3': ObjectId('67f9b0712f8b888f2fa0222'),
    '4': ObjectId('67f9b0712f8b888f2fa0222')
  }
}
lab2_2> db.Products.find()
{
  _id: ObjectId('67f9b02f8b888f2fa021f'),
  product_id: 'P001',
  name: 'Smartphone',
  category: 'Electronics',
  price: 699,
  quantity: 50
},
{
  _id: ObjectId('67f9b02f8b888f2fa021e'),
  product_id: 'P002',
  name: 'Laptop',
  category: 'Electronics',
  price: 999,
  quantity: 30
},
{
  _id: ObjectId('67f9b02f8b888f2fa021d'),
  product_id: 'P003',
  name: 'Headphones',
  category: 'Electronics',
  price: 150,
  quantity: 100
},
{
  _id: ObjectId('67f9b02f8b888f2fa021c'),
  product_id: 'P004',
  name: 'Sneakers',
  category: 'Clothing',
  price: 85,
  quantity: 200
},
{
  _id: ObjectId('67f9b02f8b888f2fa021b'),
  product_id: 'P005',
  quantity: 150
},
}
```

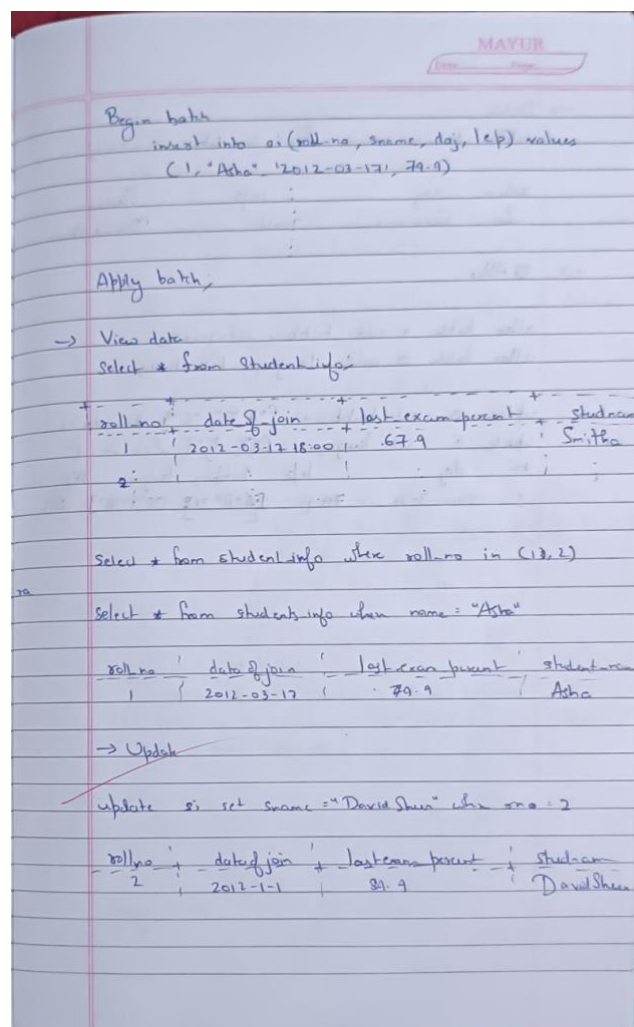
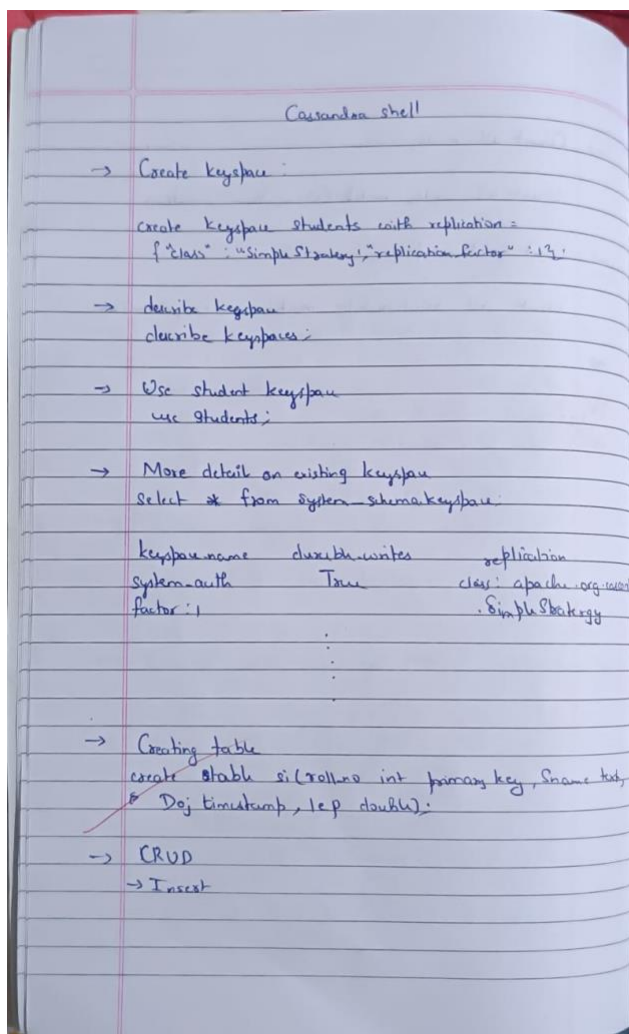
```
lab2_2> db.Carts.insertMany([
  {
    user_id: '123abc',
    cart_items: [
      { product_id: 'P002', quantity: 1 },
      { product_id: 'P003', quantity: 2 }
    ]
  },
  {
    user_id: '789ghi',
    cart_items: [
      { product_id: 'P002', quantity: 1 },
      { product_id: 'P003', quantity: 1 }
    ]
  },
  {
    user_id: '456def',
    cart_items: [
      { product_id: 'P003', quantity: 2 }
    ]
  },
  {
    user_id: '123abc',
    cart_items: [
      { product_id: 'P003', quantity: 1 },
      { product_id: 'P004', quantity: 1 }
    ]
  },
  {
    user_id: '789ghi',
    cart_items: [
      { product_id: 'P002', quantity: 1 },
      { product_id: 'P005', quantity: 3 }
    ]
  }
]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67f9b02f8b888f2fa0220'),
    '1': ObjectId('67f9b02f8b888f2fa0220'),
    '2': ObjectId('67f9b02f8b888f2fa0220'),
    '3': ObjectId('67f9b02f8b888f2fa0220'),
    '4': ObjectId('67f9b02f8b888f2fa0220')
  }
}
lab2_2> db.Carts.find({user_id: '123abc'})
{
  _id: ObjectId('67f9b02f8b888f2fa0220'),
  user_id: '123abc',
  cart_items: [
    { product_id: 'P002', quantity: 1 },
    { product_id: 'P003', quantity: 2 }
  ]
},
{
  _id: ObjectId('67f9b02f8b888f2fa0220'),
  user_id: '123abc',
  cart_items: [
    { product_id: 'P003', quantity: 1 },
    { product_id: 'P004', quantity: 1 }
  ]
}
```

```
price: 999,
quantity: 30
}
lab2_2> db.Products.find({price: {$lte: 100}})
{
  _id: ObjectId('67f9b02f8b888f2fa021c'),
  product_id: 'P004',
  name: 'Sneakers',
  category: 'Clothing',
  price: 85,
  quantity: 200
},
{
  _id: ObjectId('67f9b02f8b888f2fa021b'),
  product_id: 'P005',
  name: 'Sneakers',
  category: 'Clothing',
  price: 85,
  quantity: 150
}
lab2_2> db.Carts.find({user_id: '123abc'})
lab2_2> db.Carts.insertMany([
  {
    user_id: '123abc',
    cart_items: [
      { product_id: 'P001', quantity: 1 },
      { product_id: 'P002', quantity: 2 }
    ]
  },
  {
    user_id: '789ghi',
    cart_items: [
      { product_id: 'P001', quantity: 1 },
      { product_id: 'P002', quantity: 1 }
    ]
  },
  {
    user_id: '456def',
    cart_items: [
      { product_id: 'P003', quantity: 2 }
    ]
  },
  {
    user_id: '123abc',
    cart_items: [
      { product_id: 'P001', quantity: 1 },
      { product_id: 'P001', quantity: 1 }
    ]
  }
]);
```


Experiment – 2

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.




```

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 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> update employee_info set emp_name = 'Priyanka GH' where emp_id = '120';
InvalidRequest: Error from server: code=2206 [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=2206 [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)

```

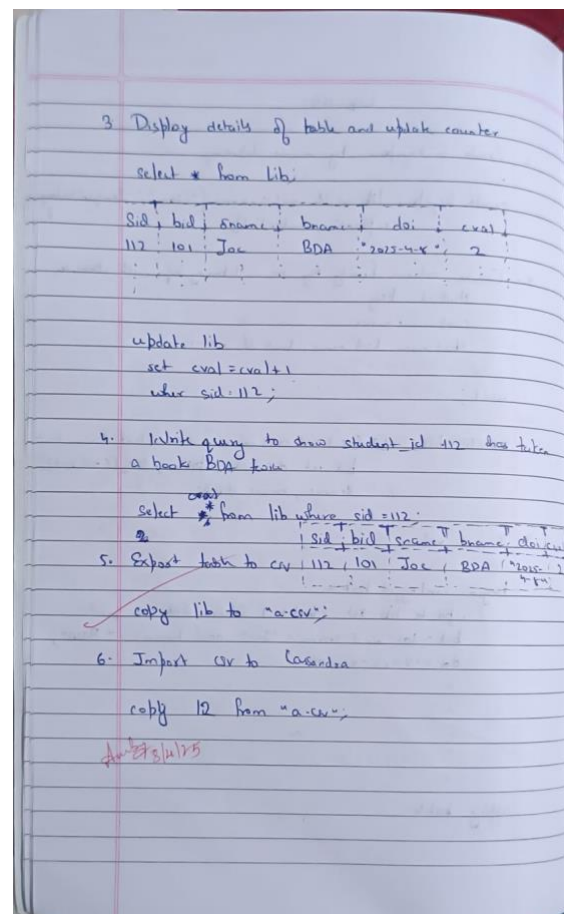
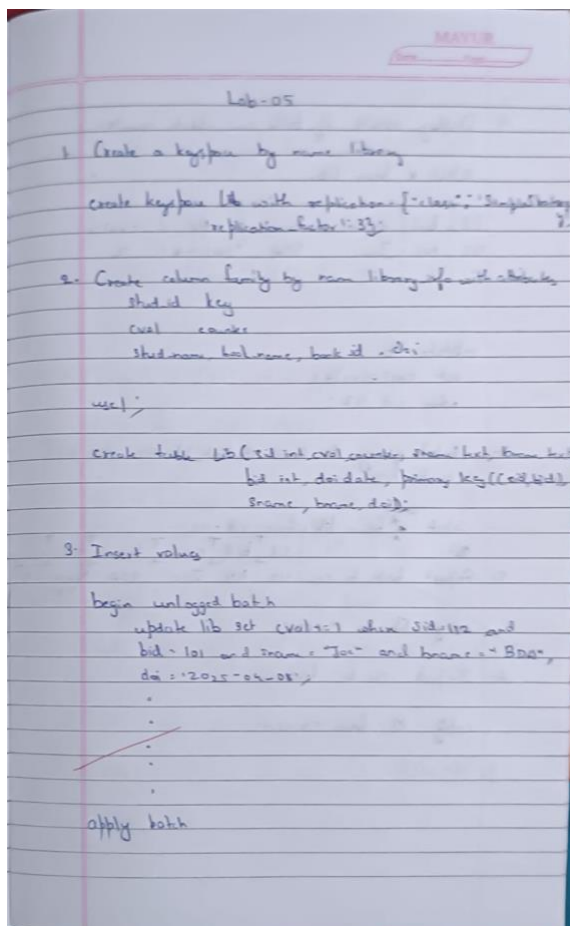
cqlsh:employee> update employee_info set bonus = 11000 where emp_id = 121;
cqlsh:employee> select * from employee_info using ttl 15 where emp_id = 123;
SyntaxException: line 1:128 mismatched Token 'using', expecting EOF token = from employee_info [using] ttl...
cqlsh:employee> select * from employee_info 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;

```

Experiment – 3

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.



Codes Output:

```
bmsccscse@bmsccscse-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 dclocal_read_consistency = 'ONE'
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,DateOfJoining, last_exam_Percent) values(4,'Charu','2023-10-05', 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
3 | 2023-10-05 18:30:00.000000+0000 | 96.5 | Charu
4 | 2023-10-09 18:30:00.000000+0000 | 97.5 | Rachana
```

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

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

```
(3 rows)
cqlsh:students> select * from students_info where StudName='Charu';
```

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
```

```
cqlsh:students> create index on Students_info(StudName);
```

```
cqlsh:students> select * from students_info where StudName='Charu';
```

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

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

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

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

```
(3 rows)
cqlsh:students> select * from students_info where StudName='Charu';
```

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
```

```
cqlsh:students> create index on Students_info(StudName);
```

```
cqlsh:students> select * from students_info where StudName='Charu';
```

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

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

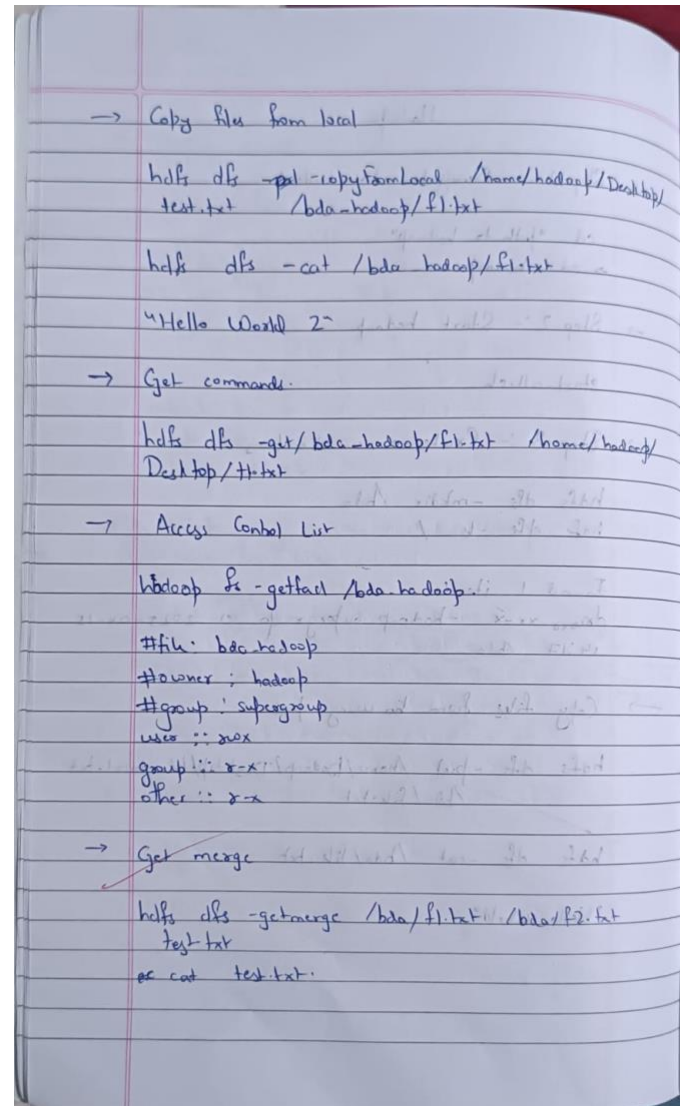
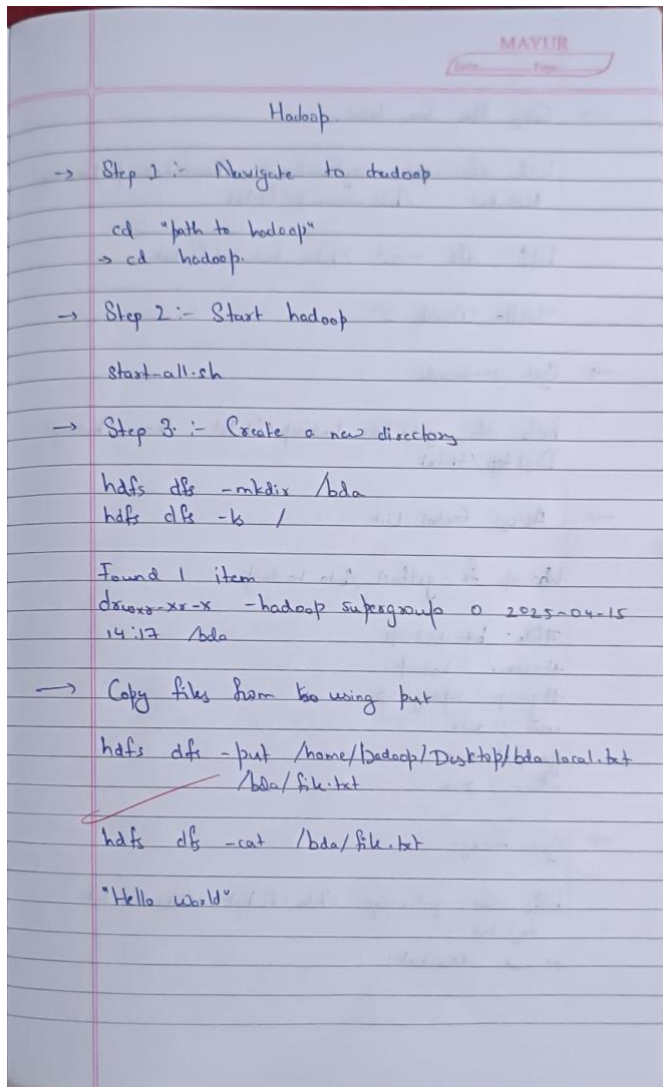
```
roll_no | studname
-----|-----
1 | Sadhana
2 | Rutu
```

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

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


Experiment-4

Execution of HDFS Commands for interaction with Hadoop Environment.



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 ../Downloads/Merged.txt
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hadoop fs -getfacl /Lab05
# file: /Lab05
# owner: hadoop
# group: supergroup
user::rwx
group::r-x
other::r-x

hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~/Desktop$ hdfs dfs -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-5

Implement Wordcount program on Hadoop framework

Lab-07

Driver.java

```
import java.io.IOException;
import java.io.Iterator;
import java.org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.util.*;

public class WCDriver extends Configuration 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]));
        FileInputFormat.setInputPaths(conf, new Path(args[1]));
        conf.setMapperClass(WCDriver.class);
        conf.setReducerClass(WCReducer.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        return 0;
    }

    public static void main(String args[]) throws Exception {
        int exitCode = ToolRunner.run(new WCDriver(), args);
    }
}
```

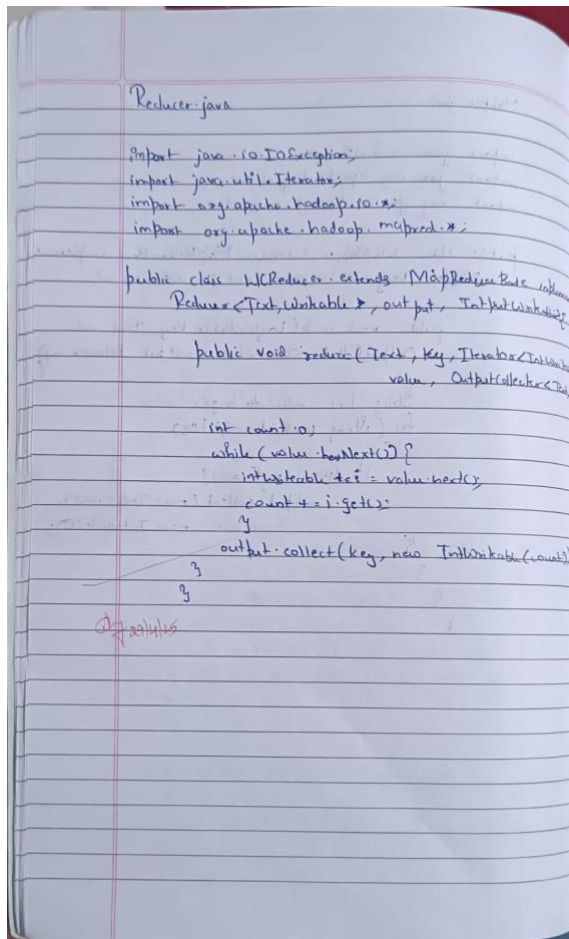
Mapper.java

MAYUR

```
import java.io.IOException;
import java.org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.Mapper;

public class WCMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

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

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@msccscs-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 8671. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [msccscs-HP-Elite-Tower-800-G9-Desktop-PC]
msccscs-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 resource manager
resource manager is running as process 9210. Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodenaggers
localhost: nodenanager is running as process 9399. Stop it first and ensure /tmp/hadoop-hadoop-nodenanager.pid file is empty before retry.
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $
nano /home/hadoop/hadoop/etc/hadoop/mapred-site.xml
hadoop@msccscs-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 [msccscs-HP-Elite-Tower-800-G9-Desktop-PC]
Stopping nodenaggers
Stopping resource manager
hadoop@msccscs-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 [msccscs-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resource manager
Starting nodenaggers
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ jps
14785 DataNode
15087 SecondaryNameNode
15089 Jps
15386 ResourceManager
15743 NodeManager
6276 org.ecliipse.equinix.launcher_1.6.1000.v28258227-1734.jar
14591 NameNode
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -ls /
Found 3 items
drwxr-xr-x - hadoop supergroup 0 2025-05-26 13:40 /folder1
drwxr-xr-x - hadoop supergroup 0 2025-05-26 13:40 /folder2
drwxr-xr-x - hadoop supergroup 0 2025-05-26 13:43 /tmp
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -mkdir /rgs
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -copyFromLocal /home/hadoop/desktop/sample.txt /rgs/test.txt
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop jar /home/hadoop/desktop/wordcount.jar WordCount.UCDriver /rgs/test.txt /rgs/output
2025-05-26 14:45:00,274 INFO Impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-05-26 14:45:00,315 INFO Impl.MetricsSystemImpl: Scheduled metric snapshot period at 10 second(s).
2025-05-26 14:45:00,333 INFO Impl.MetricsSystemImpl: JobTracker metrics system started
2025-05-26 14:45:00,321 WARN Impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2025-05-26 14:45:00,388 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the tool interface and execute your application with ToolRunner to remedy this.
2025-05-26 14:45:00,436 INFO mapred.FileInputFormat: Total input files to process : 1
2025-05-26 14:45:00,469 INFO mapreduce.JobSubmitter: number of splits:1
May 20 14:48
hadoop@msccscs-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=8
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=8
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
MRIOO_LENC=0
MRIOO_LEN=0
MRIOO_MAP=0
MRIOO_REDC=0
File Input Format Counters
Bytes Read=86
File Output Format Counters
Bytes Written=86
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -ls /output/
ls: /output/: No such file or directory
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -ls /rgs/output/
Found 2 items
-rw-r--r-- 1 hadoop supergroup 0 2025-05-26 14:41 /rgs/output/_SUCCESS
-rw-r--r-- 1 hadoop supergroup 88 2025-05-26 14:41 /rgs/output/part-00000
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ hadoop fs -cat /rgs/output/part-00000
an 1
are 1
bex 1
executed 1
fueling 1
good 1
hills 1
how 1
l 2
program 1
the 1
you 1
hadoop@msccscs-HP-Elite-Tower-800-G9-Desktop-PC: $ ss
```

Experiment-6

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.

MAYUR

Lab-08

Mapper.py

```
import sys

for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    parts = line.split()
    if len(parts) != 2:
        continue
    year, temp_str = parts
    try:
        temperature = float(temp_str)
        print(f"{year}\t{temperature}")
    except ValueError:
        continue
```

Reducer.py

```
import sys

curr = None
temp_sum = 0.0
temp_count = 0
t_min = None
t_max = None

for l in sys.stdin:
    l = l.strip()
    parts = l.split()
    y, temp = parts
    try:
        temp = float(temp)
    except ValueError:
        continue

    if curr != year:
        if curr is not None:
            avg = temp_sum / count
            print(curr + "\t" + str(t_min) + "\t" + str(t_max) + "\t" + str(avg))
            curr = year
            sum = temp
            temp_count = 1
            min = temp
            max = temp
        else:
            temp_sum += temp
            temp_count += 1
            temp_min = min(temp_min, temp)
            temp_max = max(temp_max, temp)
```

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.")
```

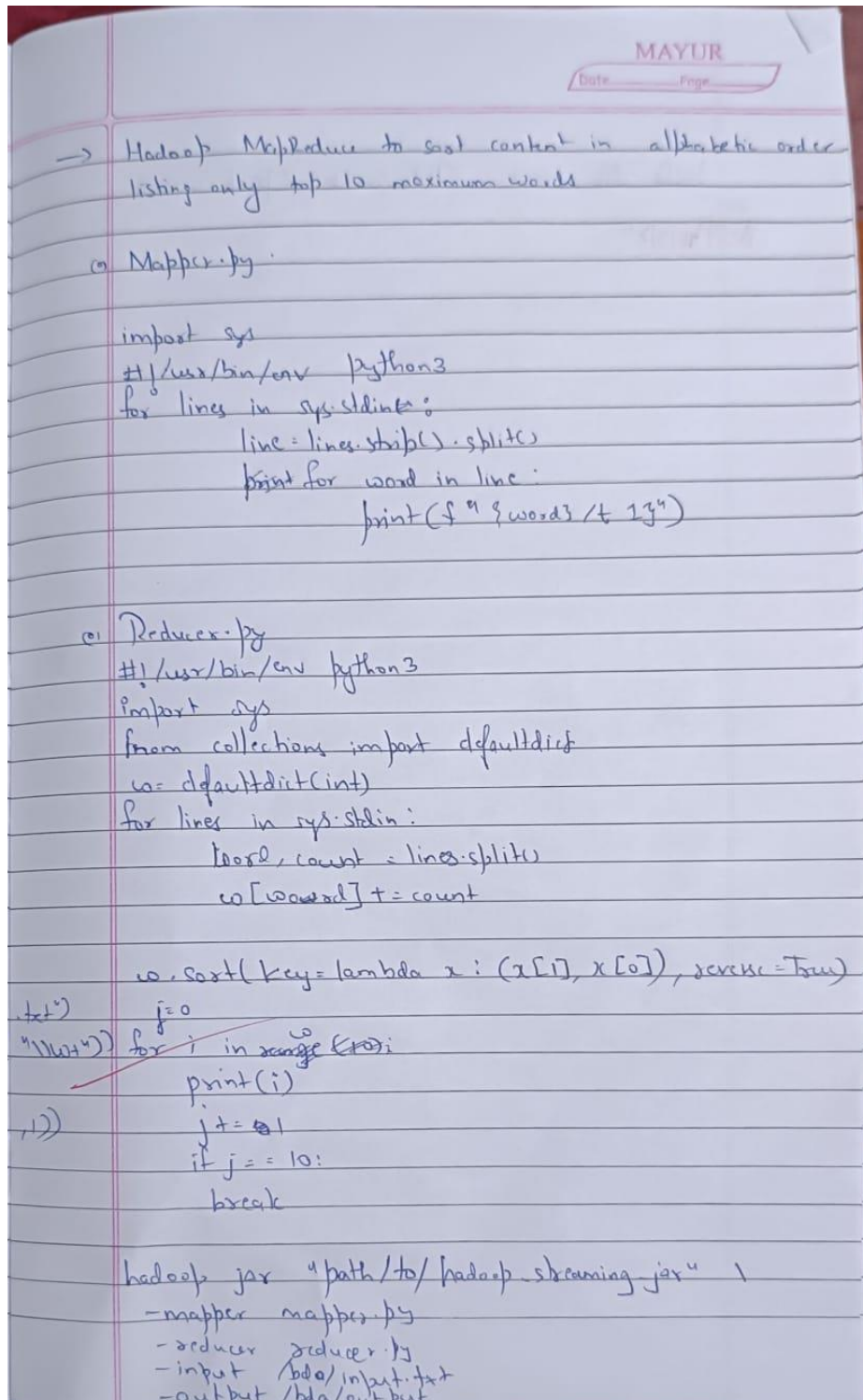

Codes Output:

```
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
prajwal@PrajwalDevice:~$
```

```
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)=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
prajwal@PrajwalDevice:~$
```

Experiment-7

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.



Mapper:

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

import sys
import re

for line in sys.stdin:
    words = re.findall(r'\w+', line.lower()) # normalize case
    for word in words:
        print(f'{word}\t1')
```

Reducer:

```
#!/usr/bin/env python3
import sys
from collections import defaultdict

N = 10 # change this to desired Top-N

word_counts = defaultdict(int)

# Aggregate word counts
for line in sys.stdin:
    word, count = line.strip().split("\t")
    word_counts[word] += int(count)

# Sort by frequency desc, then word asc
top_n = sorted(word_counts.items(), key=lambda x: (-x[1], x[0]))[:N]

# Output Top-N
for word, count in top_n:
    print(f'{word}\t{count}')
```

Codes Output:

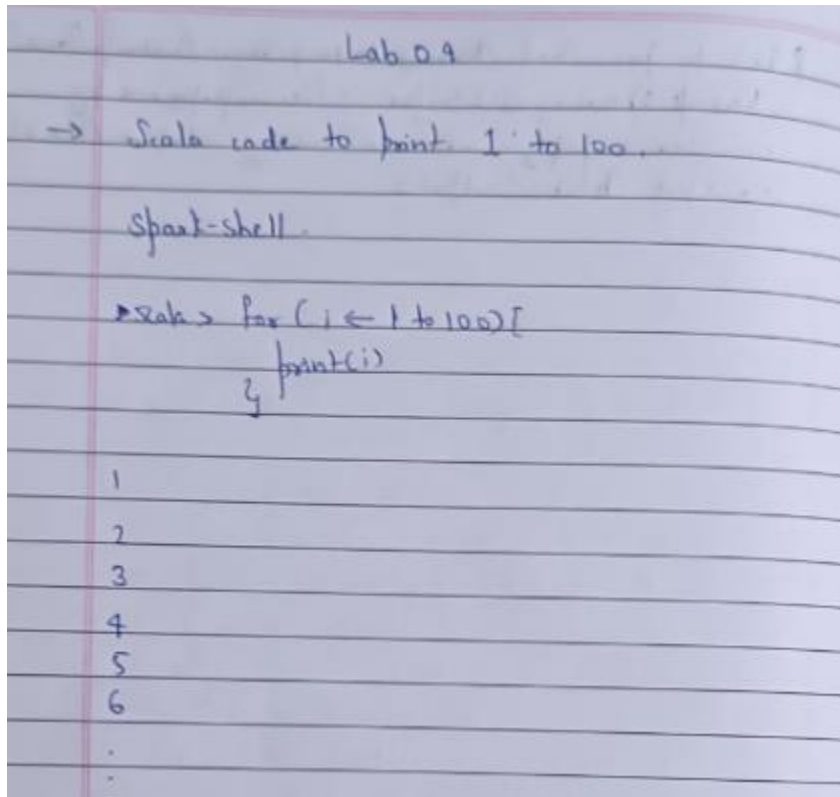
```
Combine Output Records=0
Reduce input groups=18
Reduce shuffle bytes=239
Reduce input records=25
Reduce output records=10
Spilled Records=50
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=15
Total committed heap usage (bytes)=421527552

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=137
File Output Format Counters
Bytes Written=72
2025-05-24 17:25:13,559 INFO streaming.StreamJob: Output directory: /bda/out3
prajwal@PrajwalDevice:~$ hdfs dfs -cat /bda/out3/part-00000
the      3
foxes    2
hares    2
jumps    2
quick    2
than     2
are      1
blue     1
brown    1
dog       1
```

Experiment-8

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



Scala Code:

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

0

1

2

.

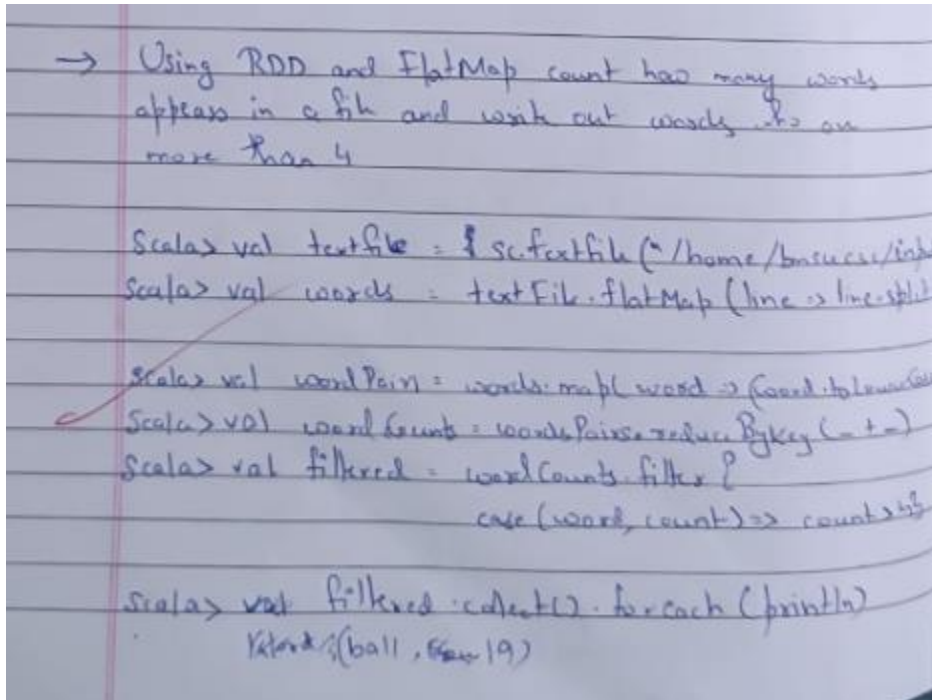
.

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  
34
```

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.



→ Using RDD and FlatMap count how many words appears in a file and write out words who are more than 4

```
Scala> val textFile = sc.textFile("/home/bnsuresh/11.txt")
Scala> val words = textFile.flatMap(line => line.split(" "))
Scala> val wordPairs = words.map(word => (word.toLowerCase, 1))
Scala> val wordCounts = wordPairs.reduceByKey(_+_).collect()
Scala> val filtered = wordCounts.filter {
  case (word, count) => count > 4
}
Scala> val filtered.collect().foreach(println)
ball 19
```

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.textFile("11.txt")
25/05/24 17:42:00 WARN GarbageCollectionMetrics: To enable non-built-in garbage collector(s) List(G1 Concurrent GC), users should configure it(them) to spark.
spark.eventLog.gcMetrics.oldGenerationGarbageCollectors
val file=sc.textFile("11.txt")
file: org.apache.spark.rdd.RDD[String] = 11.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> val words=file.flatMap(line=>line.split(" "))
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(_+_).collect()
wordc: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23

scala> val fil2=wordc.filter{case(word,count)=>count>4}
fil2: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at filter at <console>:23

scala> fil2.collect().foreach(println)
ball 19
```

