VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS

Submitted by

MANIKANTHA GADA (1BM20CS194)

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-2023 to July-2023

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 **MANIKANTHA GADA (1BM20CS194),** 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 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics - (20CS6PEBDA)** work prescribed for the said degree.

Dr. Shyamala G
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index Sheet

Course Outcome

CO1	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO2	Analyze the Big Data and obtain insight using data analytics mechanisms.
CO3	Design and implement Big data applications by applying NoSQL, Hadoop or Spark

Program no: **01** Program Title: **Cassandra Operations**

Aim: Perform the following DB operations using Cassandra -

1. Create a keyspace by name Employee

2. Create a column family by name

- Employee-Info with attributes
- Emp_Id Primary Key, Emp_Name,
- Designation, Date_of_Joining, Salary, Dept_Name

3. Insert the values into the table in batch

4. Update Employee name and Department of Emp-Id 121

5. Sort the details of Employee records based on salary

6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

7. Update the altered table to add project names.

8.Create a TTL of 15 seconds to display the values of Employees.

Commands:

1. Create a keyspace by name Employee

```
cqlsh> create keyspace Employee with replication = {
... 'class' : 'SimpleStrategy',
... 'replication_factor' : 1
... };
cqlsh> use Employee;
```

2. Create a column family by name
Employee-Info with attributes
Emp_Id Primary Key, Emp_Name,
Designation, Date_of_Joining, Salary, Dept_Name

```
cqlsh:employee> create table Employee_info(
... Emp_id int,
... Emp_name text,
```

```
... Designation text,
... DOJ timestamp,
... salary double,
... Dept_name text,
... primary key(Emp_id,salary)
... );
```

5

3. Insert the values into the table in batch

```
cqlsh:employee> begin batch
... insert into
Employee_info(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) values
(111,'John','Assistant professor','2022-05-11',75000,'CSE')
... insert into
Employee_info(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) values
(121,'Amber','Assistant professor','2022-05-11',85000,'CSE')
... insert into
Employee_info(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) values
(131,'Mary','Associate professor','2022-05-11',95000,'ECE')
... insert into
Employee_info(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) values
(141,'Jane','Associate professor','2022-05-11',105000,'ISE')
... insert into
Employee_info(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) values
(151,'Yelena','Associate professor','2022-05-11',95000,'ISE')
... apply batch;
cqlsh:employee> select * from Employee_info;

emp_id | salary | dept_name | designation | doj |
emp_name
--------+----------+-----------+---------------------+---------------------------------+----------
111 | 75000 | CSE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | John
151 | 95000 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Yelena
```

6

```
121 | 85000 | CSE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | Amber
141 | 1.05e+05 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Jane
131 | 95000 | ECE | Associate professor | 2022-05-10
```

18:30:00.000000+0000 | Mary
4. Update Employee name and Department of Emp-Id 121
cqlsh:employee> update Employee_info set Emp_name = 'Josh', Dept_name =
'ECE' where Emp_id = 121 and salary = 85000;
cqlsh:employee> select * from Employee_info;

emp_id | salary | dept_name | designation | doj |
emp_name
--------+----------+-----------+---------------------+---------------------------------+----------
111 | 75000 | CSE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | John
151 | 95000 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Yelena
121 | 85000 | ECE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | Josh
141 | 1.05e+05 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Jane
131 | 95000 | ECE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Mary

(5 rows)

7

5. Sort the details of Employee records based on salary
cqlsh:employee> select * from Employee_info where Emp_id
in(111,121,131,141,151) order by salary desc;

emp_id | salary | dept_name | designation | doj |
emp_name
--------+----------+-----------+---------------------+---------------------------------+----------
141 | 1.05e+05 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Jane
131 | 95000 | ECE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Mary
151 | 95000 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Yelena
121 | 85000 | ECE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | Josh
111 | 75000 | CSE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | John

(5 rows)
6. Alter the schema of the table Employee_Info to add a column Projects which

stores a set of Projects done by the corresponding Employee.
cqlsh:employee> update Employee_info set project = project+{'AI','Data warehouse'} where Emp_id = 111 and salary = 75000;
cqlsh:employee> update Employee_info set project = project+{'IOT','Data warehouse'} where Emp_id = 121 and salary = 85000;

8

cqlsh:employee> update Employee_info set project = project+{'IOT','AI'} where Emp_id = 131 and salary = 95000;
cqlsh:employee> update Employee_info set project = project+{'IOT','machine learning'} where Emp_id = 141 and salary = 95000;
cqlsh:employee> update Employee_info set project = project+{'IOT','data science'} where Emp_id = 141 and salary = 105000;
cqlsh:employee> select * from Employee_info;

```
emp_id | salary | dept_name | designation | doj |
emp_name | project
--------+----------+-----------+---------------------+---------------------------------+----------+------
-----------------------
111 | 75000 | CSE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | John | {'AI', 'Data warehouse'}
151 | 95000 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Yelena | null
121 | 85000 | ECE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | Josh | {'Data warehouse', 'IOT'}
141 | 95000 | null | null | null | null | {'IOT',
'machine learning'}
141 | 1.05e+05 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Jane | {'IOT', 'data science'}
131 | 95000 | ECE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Mary | {'AI', 'IOT'}

(6 rows)
```

9

7. Update the altered table to add project names.
cqlsh:employee> update Employee_info set project = project+{'IOT','AI'} where Emp_id = 151 and salary = 95000;
cqlsh:employee> select * from Employee_info;

```
emp_id | salary | dept_name | designation | doj |
emp_name | project
```

--------+----------+-----------+--------------------+--------------------------------+----------+------
----
111 | 75000 | CSE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | John | {'AI', 'Data warehouse'}
151 | 95000 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Yelena | {'AI', 'IOT'}
121 | 85000 | ECE | Assistant professor | 2022-05-10
18:30:00.000000+0000 | Josh | {'Data warehouse', 'IOT'}
141 | 95000 | null | null | null | null | {'IOT',
'machine learning'}
141 | 1.05e+05 | ISE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Jane | {'IOT', 'data science'}
131 | 95000 | ECE | Associate professor | 2022-05-10
18:30:00.000000+0000 | Mary | {'AI', 'IOT'}

(6 rows)

10

8.Create a TTL of 15 seconds to display the values of Employees.
cqlsh:employee> insert into
Employee_info(Emp_id,Emp_name,Designation,DOJ,salary,Dept_name) values
(161,'Ryan','Associate professor','2022-05-11',95000,'ISE') using ttl 60;
cqlsh:employee> select ttl(Emp_name) from Employee_info where Emp_id = 161
and salary = 95000;

ttl(emp_name)

53

(1 rows)

11

Program no: **02** Program Title: **More Cassandra Operations**

Aim: Perform the following DB operations using Cassandra -

1.Create a keyspace by name Library

2. 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

3. Insert the values into the table in batch

4. Display the details of the table created and increase the value of the counter

5. Write a query to show that a student with id 112 has taken a book “BDA” 2 times.

6. Export the created column to a csv file

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

Commands:

1.Create a keyspace by name Library

```
cqlsh> create keyspace library with replication = {
... 'class' : 'SimpleStrategy',
... 'replication_factor' : 1
... };
cqlsh> use library
... ;
```

2. 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

```
cqlsh:library> create table library_info (
... stud_id int,
... stud_name text,
... book_id int,
... book_name text,
... date_of_issue timestamp,
... counter_value counter,
```

```
... primary key ((stud_id,book_id),stud_name,book_name,date_of_issue)
... );
```

12

3. Insert the values into the table in batch

```
cqlsh:library> update library_info
... set counter_value = counter_value+1
... where stud_id = 111 and stud_name = 'Raj' and book_id = 100 and
book_name = 'ADA' and date_of_issue = '2022-04-05';
cqlsh:library> update library_info
... set counter_value = counter_value+1
... where stud_id = 112 and stud_name = 'Ram' and book_id = 200 and
book_name = 'DSA' and date_of_issue = '2022-04-06';
cqlsh:library> update library_info
... set counter_value = counter_value+1
... where stud_id = 113 and stud_name = 'sohan' and book_id = 300 and
book_name = 'JAVA' and date_of_issue = '2022-04-07';
cqlsh:library> update library_info
... set counter_value = counter_value+1
... where stud_id = 114 and stud_name = 'rohan' and book_id = 400 and
book_name = 'UNIX' and date_of_issue = '2022-04-07';
```

4. Display the details of the table created and increase the value of the counter

```
cqlsh:library> select * from library_info;

stud_id | book_id | stud_name | book_name | date_of_issue |
counter_value
---------+---------+-----------+-----------+--------------------------------+---------------
114 | 400 | rohan | UNIX | 2022-04-06 18:30:00.000000+0000 |
1
111 | 100 | Raj | ADA | 2022-04-04 18:30:00.000000+0000 | 1
```

13

```
112 | 200 | Ram | DSA | 2022-04-05 18:30:00.000000+0000 | 1
113 | 300 | sohan | JAVA | 2022-04-06 18:30:00.000000+0000 |
1
cqlsh:library> update library_info
... set counter_value = counter_value+1
... where stud_id = 114 and stud_name = 'rohan' and book_id = 400 and
book_name = 'UNIX' and date_of_issue = '2022-04-07';
```

5. Write a query to show that a student with id 114 has taken a book “UNIX” 2 times.

```
cqlsh:library> select stud_id from library_info where book_name = 'UNIX' and
counter_value = 2 allow filtering;

stud_id
---------
114

(1 rows)
```

6. Export the created column to a csv file

```
cqlsh:library> copy
library_info(stud_id,stud_name,book_id,book_name,date_of_issue,counter_value
) to 'd:\library_info.csv';
Using 15 child processes

Starting copy of library.library_info with columns [stud_id, stud_name, book_id,
book_name, date_of_issue, counter_value].
Processed: 4 rows; Rate: 1 rows/s; Avg. rate: 1 rows/s

14

4 rows exported to 1 files in 5.025 seconds.
```

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

```
cqlsh:library> truncate library_info;
cqlsh:library> select * from library_info;

stud_id | book_id | stud_name | book_name | date_of_issue | counter_value
---------+---------+-----------+-----------+---------------+---------------

(0 rows)

cqlsh:library> truncate library_info;
cqlsh:library> select * from library_info;

stud_id | book_id | stud_name | book_name | date_of_issue | counter_value
---------+---------+-----------+-----------+---------------+---------------

(0 rows)
cqlsh:library> copy
library_info(stud_id,book_id,stud_name,book_name,date_of_issue,counter_value
) from 'd:\library_info.csv' with header = true;
Using 15 child processes

Starting copy of library.library_info with columns [stud_id, book_id, stud_name,
book_name, date_of_issue, counter_value].
```

Process ImportProcess-256: 1 rows/s; Avg. rate: 1 rows/s

15

```
cqlsh:library> select * from library_info;

stud_id | book_id | stud_name | book_name | date_of_issue |
counter_value
---------+---------+-----------+-----------+---------------------------------+---------------
111 | 100 | ram | ada | 2022-05-04 18:30:00.000000+0000 | 1
112 | 200 | raj | dsa | 2022-05-05 18:30:00.000000+0000 | 2
113 | 300 | shyam | ada | 2022-05-06 18:30:00.000000+0000 |
1
```

Program no: **03** Program Title: **MongoDB - Crud Demonstration**

Aim: Demonstrate the crud operations in MongoDB

Code & Output: bmsce@bmsce-Precision-T1700:~$ mongo.sh
mongo.sh: command not found
bmsce@bmsce-Precision-T1700:~$ mongosh

Command 'mongosh' not found, did you mean:

  command 'mongos' from deb mongodb-server-core
(1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu2)

Try: sudo apt install <deb name>

bmsce@bmsce-Precision-T1700:~$ mongodbsh
mongodbsh: command not found
bmsce@bmsce-Precision-T1700:~$ mongodb

Command 'mongodb' not found, did you mean:

  command 'mongod' from deb mongodb-server-core
(1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu2)

Try: sudo apt install <deb name>

bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" :
UUID("39c28cce-395e-4cfc-aeca-97b72f2806c5") }
MongoDB server version: 3.6.8
Server has startup warnings:
2023-04-01T09:16:24.545+0530 I STORAGE [initandlisten]
2023-04-01T09:16:24.545+0530 I STORAGE [initandlisten] ** WARNING:
Using the XFS filesystem is strongly recommended with the WiredTiger
storage engine
2023-04-01T09:16:24.545+0530 I STORAGE [initandlisten] **
See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-04-01T09:16:31.820+0530 I CONTROL [initandlisten]
2023-04-01T09:16:31.820+0530 I CONTROL [initandlisten] ** WARNING:

```
Access control is not enabled for the database.
2023-04-01T09:16:31.821+0530 I CONTROL  [initandlisten] **
Read and write access to data and configuration is unrestricted.
2023-04-01T09:16:31.821+0530 I CONTROL  [initandlisten]
> show db
2023-04-01T09:20:58.208+0530 E QUERY    [thread1] Error: don't know
how to show [db] :
shellHelper.show@src/mongo/shell/utils.js:997:11
shellHelper@src/mongo/shell/utils.js:750:15
@(shellhelp2):1:1
> show dbs
Student   0.000GB
admin     0.000GB
config    0.000GB
dm        0.000GB
faculty   0.000GB
labtest   0.000GB
local     0.000GB
myDB      0.000GB
playlist  0.000GB
sample    0.000GB
studDB    0.000GB
studb     0.000GB
students  0.000GB
t1        0.000GB
test      0.000GB
> db
test
> use erenyeager
switched to db erenyeager
> db
erenyeager
> use mikasa
switched to db mikasa
> db
mikasa
> show dbs
Student   0.000GB
admin     0.000GB
config    0.000GB
dm        0.000GB
faculty   0.000GB
labtest   0.000GB
local     0.000GB
myDB      0.000GB
playlist  0.000GB
```

```
sample    0.000GB
studDB    0.000GB
studb     0.000GB
students  0.000GB
t1        0.000GB
test      0.000GB
> use aot
switched to db aot
> db
aot
> db.createCollection("surveycorps")
{ "ok" : 1 }
> db.createCollection("kingfritz")
{ "ok" : 1 }
> db.kingfritz.drop()
true
> db.surveycorps.find({});
{ "_id" : ObjectId("6427ac890d554b2816900352"), "id" : 1, "name" :
"Mikasa", "grade" : "10", "special" : "ackerman" }
{ "_id" : ObjectId("6427ace10d554b2816900353"), "id" : 2, "name" :
"eren", "grade" : "10", "special" : "attacktitan" }
> db.surveycorps.update({id:"2",grade:"10",special:"attactitan },{4set:{name:"ERENYEAGER"
} },{upsert:true} );
2023-04-01T09:43:00.954+0530 E QUERY    [thread1] SyntaxError: missing
} after property list @(shell):1:76
> db.surveycorps.update({id:"2",grade:"10",special:"attactitan },{$set:{name:"ERENYEAGER"
} },{upsert:true} );
2023-04-01T09:43:47.067+0530 E QUERY    [thread1] SyntaxError: missing
} after property list @(shell):1:76
> db.surveycorps.update({id:"2",grade:"10",special:"attactitan"
},{$set:{name:"ERENYEAGER" }},{upsert:true});
WriteResult({
"nMatched" : 0,
"nUpserted" : 1,
"nModified" : 0,
"_id" : ObjectId("6427afbb2a14840302172849")
})
> db.surveycorps.find({special:"attacktitan"});
{ "_id" : ObjectId("6427ace10d554b2816900353"), "id" : 2, "name" :
"eren", "grade" : "10", "special" : "attacktitan" }
> db.surveycorps.update({id:"2",grade:"10",special:"attacktitan"
},{$set:{name:"ERENYEAGER" }},{upsert:true});
WriteResult({
"nMatched" : 0,
"nUpserted" : 1,
"nModified" : 0,
```

```
"_id" : ObjectId("6427b0df2a14840302172850")
})
> db.surveycorps.find({special:"attacktitan"});
{ "_id" : ObjectId("6427ace10d554b2816900353"), "id" : 2, "name" :
"eren", "grade" : "10", "special" : "attacktitan" }
{ "_id" : ObjectId("6427b0df2a14840302172850"), "grade" : "10", "id" :
"2", "special" : "attacktitan", "name" : "ERENYEAGER" }
> db.surveycorps.insert({_id:1,name:"Mikasa",grade:"10",special } )
2023-04-01T09:50:58.484+0530 E QUERY    [thread1] ReferenceError:
special is not defined :
@(shell):1:55
> db.surveycorps.insert({_id:1,name:"Mikasa",grade:"10",special } );
2023-04-01T09:51:19.876+0530 E QUERY    [thread1] ReferenceError:
special is not defined :
@(shell):1:55
> db.surveycorps.insert({_id:1,name:"Mikasa",grade:"10",special:"ackerman" } );
WriteResult({ "nInserted" : 1 })
> db.surveycorps.insert({_id:1,name:"armin",grade:"10",special:"mind" } );
WriteResult({
"nInserted" : 0,
"writeError" : {
"code" : 11000,
"errmsg" : "E11000 duplicate key error collection: aot.surveycorps
index: _id_ dup key: { : 1.0 }"
}
})
> db.surveycorps.insert({_id:2,name:"armin",grade:"10",special:"mind" } );
WriteResult({ "nInserted" : 1 })
> db.surveycorps.update({_id:2,name:"armin",grade:"10" },{$set:{special:"colossaltitan"
}},{upsert:true});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.surveycorps.find({});
{ "_id" : ObjectId("6427ac890d554b2816900352"), "id" : 1, "name" :
"Mikasa", "grade" : "10", "special" : "ackerman" }
{ "_id" : ObjectId("6427ace10d554b2816900353"), "id" : 2, "name" :
"eren", "grade" : "10", "special" : "attacktitan" }
{ "_id" : ObjectId("6427afbb2a14840302172849"), "grade" : "10", "id" :
"2", "special" : "attactitan", "name" : "ERENYEAGER" }
{ "_id" : ObjectId("6427b0df2a14840302172850"), "grade" : "10", "id" :
"2", "special" : "attacktitan", "name" : "ERENYEAGER" }
{ "_id" : 1, "name" : "Mikasa", "grade" : "10", "special" : "ackerman" }
{ "_id" : 2, "name" : "armin", "grade" : "10", "special" : "colossaltitan" }
> db.surveycorps.find({},{name:1,grade:1,id:0,_id:0} );
Error: error: {
"ok" : 0,
"errmsg" : "Projection cannot have a mix of inclusion and exclusion.",
```

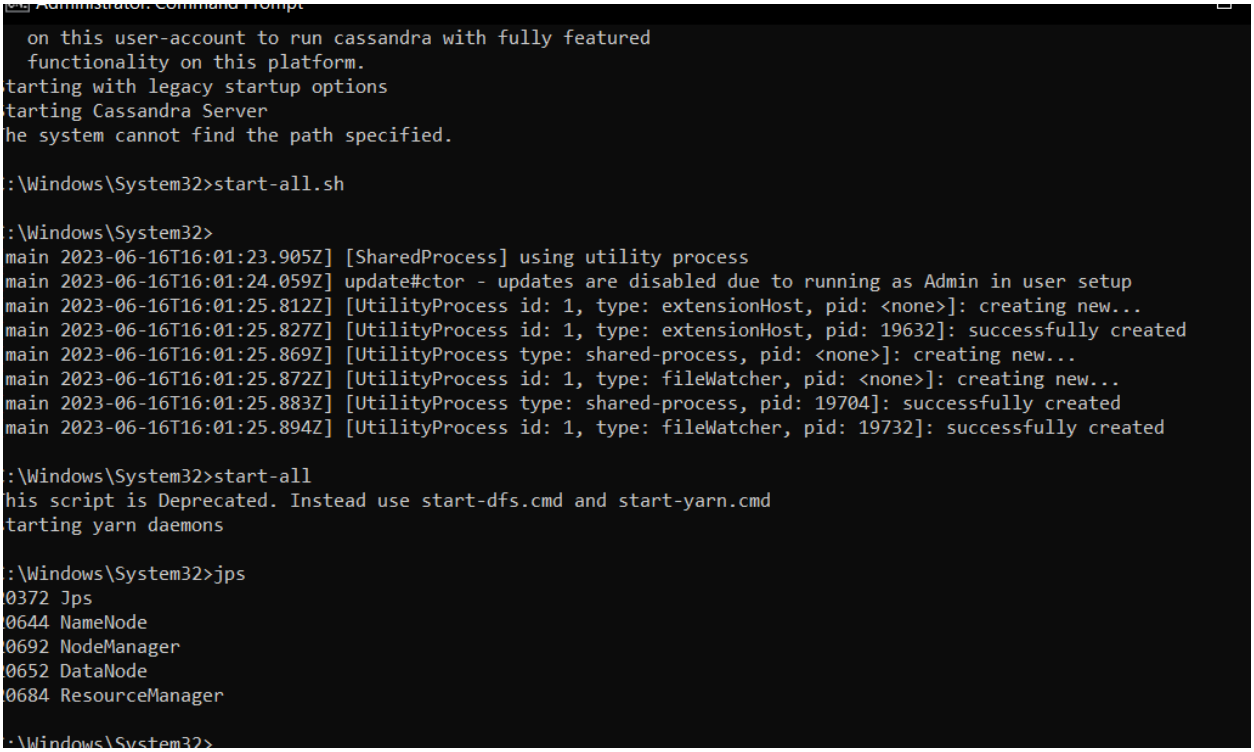
```
"code" : 2,
"codeName" : "BadValue"
}
> db.surveycorps.find({},{name:1,grade:1,_id:0} );
{ "name" : "Mikasa", "grade" : "10" }
{ "name" : "eren", "grade" : "10" }
{ "grade" : "10", "name" : "ERENYEAGER" }
{ "grade" : "10", "name" : "ERENYEAGER" }
{ "name" : "Mikasa", "grade" : "10" }
{ "name" : "armin", "grade" : "10" }
> db.surveycorps.find({},{name:1,grade:1,_id:0} ).pretty(0);
{ "name" : "Mikasa", "grade" : "10" }
{ "name" : "eren", "grade" : "10" }
{ "grade" : "10", "name" : "ERENYEAGER" }
{ "grade" : "10", "name" : "ERENYEAGER" }
{ "name" : "Mikasa", "grade" : "10" }
{ "name" : "armin", "grade" : "10" }
> db.surveycorps.find({},{name:1,grade:1,_id:0} ).pretty();
{ "name" : "Mikasa", "grade" : "10" }
{ "name" : "eren", "grade" : "10" }
{ "grade" : "10", "name" : "ERENYEAGER" }
{ "grade" : "10", "name" : "ERENYEAGER" }
{ "name" : "Mikasa", "grade" : "10" }
{ "name" : "armin", "grade" : "10" }
> db.surveycorps.insert({_id:3,name:"jean",grade:"5",special:"horseface" } );
WriteResult({ "nInserted" : 1 })
> db.surveycorps.insert({_id:4,name:"connie",grade:"5",special:"dumb" } );
WriteResult({ "nInserted" : 1 })
> db.surveycorps.find({grade:{eq:"5"}}).pretty();
>
> db.surveycorps.find({grade:{eq:"5"}}).pretty();
> db.surveycorps.find({grade:{$eq:"5"}}).pretty();
{ "_id" : 3, "name" : "jean", "grade" : "5", "special" : "horseface" }
{ "_id" : 4, "name" : "connie", "grade" : "5", "special" : "dumb" }
> db.surveycorps.find({special:{$in:["dumb","colossaltitan"]}}).pretty();
{
"_id" : 2,
"name" : "armin",
"grade" : "10",
"special" : "colossaltitan"
}
{ "_id" : 4, "name" : "connie", "grade" : "5", "special" : "dumb" }
> db.surveycorps.find(name:/^M/});
2023-04-01T10:06:27.783+0530 E QUERY    [thread1] SyntaxError: missing
) after argument list @(shell):1:24
> db.surveycorps.find({name:/^M/});
```

```
{ "_id" : ObjectId("6427ac890d554b2816900352"), "id" : 1, "name" :
"Mikasa", "grade" : "10", "special" : "ackerman" }
{ "_id" : 1, "name" : "Mikasa", "grade" : "10", "special" : "ackerman" }
> db.surveycorps.find({name:/e/});
{ "_id" : ObjectId("6427ace10d554b2816900353"), "id" : 2, "name" :
"eren", "grade" : "10", "special" : "attacktitan" }
{ "_id" : 3, "name" : "jean", "grade" : "5", "special" : "horseface" }
{ "_id" : 4, "name" : "connie", "grade" : "5", "special" : "dumb" }
> db.surveycorps.count();
8
> db.surveycorps.find().sort({name:-1}).pretty();
{ "_id" : 3, "name" : "jean", "grade" : "5", "special" : "horseface" }
{
"_id" : ObjectId("6427ace10d554b2816900353"),
"id" : 2,
"name" : "eren",
"grade" : "10",
"special" : "attacktitan"
}
{ "_id" : 4, "name" : "connie", "grade" : "5", "special" : "dumb" }
{
"_id" : 2,
"name" : "armin",
"grade" : "10",
"special" : "colossaltitan"
}
{
"_id" : ObjectId("6427ac890d554b2816900352"),
"id" : 1,
"name" : "Mikasa",
"grade" : "10",
"special" : "ackerman"
}
{ "_id" : 1, "name" : "Mikasa", "grade" : "10", "special" : "ackerman" }
{
"_id" : ObjectId("6427afbb2a14840302172849"),
"grade" : "10",
"id" : "2",
"special" : "attactitan",
"name" : "ERENYEAGER"
}
{
"_id" : ObjectId("6427b0df2a14840302172850"),
"grade" : "10",
"id" : "2",
"special" : "attacktitan",
```

```
"name" : "ERENYEAGER"
}
> mongoimport --db aot --collection surveycorps --type csv -headerline --file
/home/bmsce/Downloads/username.csv
2023-04-01T10:16:10.634+0530 E QUERY    [thread1] SyntaxError: missing
; before statement @(shell):1:14
> mongoimport --db aot --collection surveycorps --type csv -headerline --file
/home/bmsce/Downloads/username.csv;
2023-04-01T10:16:15.466+0530 E QUERY    [thread1] SyntaxError: missing
; before statement @(shell):1:14
> mongoimport --db aot --collection surveycorps --type csv -headerline --file
/home/bmsce/Downloads/username.csv;^C
bye
bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" :
UUID("7c88c987-7084-4a02-bd82-90091259985f") }
MongoDB server version: 3.6.8
Server has startup warnings:
2023-04-01T09:16:24.545+0530 I STORAGE  [initandlisten]
2023-04-01T09:16:24.545+0530 I STORAGE  [initandlisten] ** WARNING:
Using the XFS filesystem is strongly recommended with the WiredTiger
storage engine
2023-04-01T09:16:24.545+0530 I STORAGE  [initandlisten] **
See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-04-01T09:16:31.820+0530 I CONTROL  [initandlisten]
2023-04-01T09:16:31.820+0530 I CONTROL  [initandlisten] ** WARNING:
Access control is not enabled for the database.
2023-04-01T09:16:31.821+0530 I CONTROL  [initandlisten] **
Read and write access to data and configuration is unrestricted.
2023-04-01T09:16:31.821+0530 I CONTROL  [initandlisten]
> db
test
> use aot
switched to db aot
> db
aot
> db.surveycorps.save({name:"armin",grade:"10"});
WriteResult({ "nInserted" : 1 })
> db.surveycorps.update({_id:4},{$set:{location:"network"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.surveycorps.update({_id:4},{$unset:{location:"network"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.surveycorps.find( {)
```

Program no: **04** Program Title: **Screenshot of Hadoop Installation**

Screenshot:



```
  on this user-account to run cassandra with fully featured
  functionality on this platform.
tarting with legacy startup options
tarting Cassandra Server
he system cannot find the path specified.

:\Windows\System32>start-all.sh

:\Windows\System32>
main 2023-06-16T16:01:23.905Z] [SharedProcess] using utility process
main 2023-06-16T16:01:24.059Z] update#ctor - updates are disabled due to running as Admin in user setup
main 2023-06-16T16:01:25.812Z] [UtilityProcess id: 1, type: extensionHost, pid: <none>]: creating new...
main 2023-06-16T16:01:25.827Z] [UtilityProcess id: 1, type: extensionHost, pid: 19632]: successfully created
main 2023-06-16T16:01:25.869Z] [UtilityProcess type: shared-process, pid: <none>]: creating new...
main 2023-06-16T16:01:25.872Z] [UtilityProcess id: 1, type: fileWatcher, pid: <none>]: creating new...
main 2023-06-16T16:01:25.883Z] [UtilityProcess type: shared-process, pid: 19704]: successfully created
main 2023-06-16T16:01:25.894Z] [UtilityProcess id: 1, type: fileWatcher, pid: 19732]: successfully created

:\Windows\System32>start-all
his script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
tarting yarn daemons

:\Windows\System32>jps
0372 Jps
0644 NameNode
0692 NodeManager
0652 DataNode
0684 ResourceManager

:\Windows\System32>
```

Program no: **05** Program Title: **HDFS Commands**

Aim: Execution of HDFS Commands for interaction with Hadoop Environment.

Commands:

1. mkdir
Hadoop HDFS mkdir Command Usage
mkdir
Hadoop HDFS mkdir Command Example
hdfs dfs -mkdir /abc
Hadoop HDFS mkdir Command Description
This HDFS command takes path URI's as an argument and creates directories.

2. ls
Hadoop HDFS ls Command Usage
ls
Hadoop HDFS ls Command Example
hadoop fs -ls /
Hadoop HDFS ls Commnad Description
This Hadoop HDFS ls command displays a list of the contents of a directory specified by path provided by the user, showing the names, permissions, owner, size and modification date for each entry.

3. put
Hadoop HDFS put Command Usage
put
Hadoop HDFS put Command Example
hdfs dfs -put /home/hduser/Desktop/Welcome.txt /abc/WC.txt
Hadoop
HDFS put Command Description
This hadoop basic command copies the file or directory from the local file system to the destination within the DFS.
Display the contents of the file WC.txt
hdfs dfs -cat /abc/WC.txt

4. copyFromLocal
Hadoop HDFS copyFromLocal Command Usage
copyFromLocal
Hadoop HDFS copyFromLocal Command Example

hdfs dfs -put /home/hduser/Desktop/Welcome.txt /abc/WC.txt
Hadoop HDFS copyFromLocal Command Description
This hadoop shell command is similar to put command, but the source is restricted to a local file reference.
Display the contents of the file WC2.txt
hdfs dfs -cat /abc/WC2.txt

5. get
Hadoop HDFS get Command Usage
get [-crc]
i.Hadoop HDFS get Command Example
hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/WWC.txt

This HDFS fs command copies the file or directory in HDFS identified by the source to the local file system path identified by local destination.

ii.Hadoop HDFS get Command Example
hdfs dfs -getmerge /abc/WC.txt /abc/WC2.txt /home/hduser/Desktop/Merge.txt

This HDFS basic command retrieves all files that match to the source path entered by the user in HDFS, and creates a copy of them to one single, merged file in the local file system identified by local destination.

iii. Hadoop HDFS get Command Example
hadoop fs -getfacl /abc/

This Apache Hadoop command shows the Access Control Lists (ACLs) of files and directories.

6. copyToLocal
Hadoop HDFS copyToLocal Command Usage
copyToLocal
Hadoop HDFS copyToLocal Command Example
hdfs dfs -copyToLocal /abc/WC.txt /home/hduser/Desktop

Similar to get command, only the difference is that in this the destination is restricted to a local file reference.

7. cat
Hadoop HDFS cat Command Usage
cat
Hadoop HDFS cat Command Example

hdfs dfs -cat /abc/WC.txt

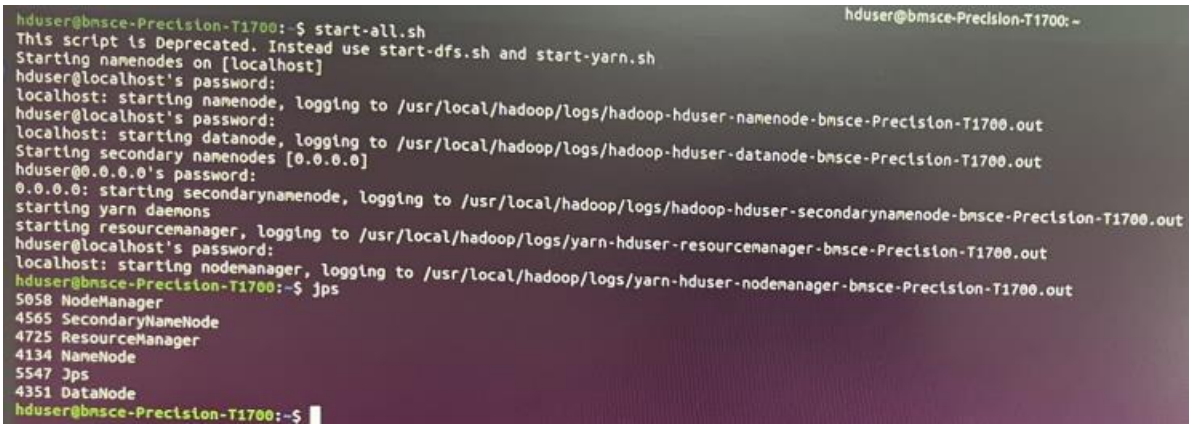
This Hadoop fs shell command displays the contents of the filename on console or stdout.

8. mv
Hadoop HDFS mv Command Usage
mv
Hadoop HDFS mv Command Example
hadoop fs -mv /abc /FFF
hadoop fs -ls /FFF

This basic HDFS command moves the file or directory indicated by the source to destination, within HDFS.

9. cp
Hadoop HDFS cp Command Usage
cp
Hadoop HDFS cp Command Example
hadoop fs -cp /CSE/ /LLL
hadoop fs -ls /LLL
The cp command copies a file from one directory to another directory within the HDFS.

Output:



```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir/afifah
-mkdir/afifah: Unknown command
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir / afifah
mkdir: `/': File exists
mkdir: `afifah': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /afifah
hduser@bmsce-Precision-T1700:~$ hdfs dfs cd afifah
cd: Unknown command
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /
Found 13 items
drwxr-xr-x   - hduser supergroup          0 2019-10-23 16:07 /STUDENT_INFO
drwxr-xr-x   - hduser supergroup          0 2023-04-27 12:34 /abc
drwxr-xr-x   - hduser supergroup          0 2023-05-11 13:54 /afifah
drwxr-xr-x   - hduser supergroup          0 2019-10-23 15:08 /arv
drwxr-xr-x   - hduser supergroup          0 2023-05-04 13:05 /inputbda
drwxr-xr-x   - hduser supergroup          0 2023-04-27 11:48 /lab5hadoop
drwxr-xr-x   - hduser supergroup          0 2023-05-08 09:40 /new_folder
drwxr-xr-x   - hduser supergroup          0 2022-06-14 10:14 /output
drwxr-xr-x   - hduser supergroup          0 2023-05-04 13:15 /outputbda
drwxr-xr-x   - hduser supergroup          0 2022-06-14 10:09 /rgs
drwxr-xr-x   - hduser supergroup          0 2023-04-27 11:47 /test
drwxrwxr-x   - hduser supergroup          0 2019-10-23 15:36 /tmp
drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs  -put /home/hduser/Desktop/Welcome.txt /abc/WC.txt
put: `/home/hduser/Desktop/Welcome.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs  -put /home/hduser/Desktop/sample.txt /abc/WC.txt
hduser@bmsce-Precision-T1700:~$
```

```
put: `/home/hduser/Desktop/Welcome.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs  -put /home/hduser/Desktop/sample.txt /abc/WC.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs  -copyFromLocal /home/hduser/Desktop/sample.txt /abc/WC.txt
copyFromLocal: `/abc/WC.txt': File exists
hduser@bmsce-Precision-T1700:~$ hdfs dfs  -copyFromLocal /home/hduser/Desktop/sample.txt /abc/WC1.txt
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs  -copyFromLocal /home/hduser/Desktop/sample.txt /abc/WC1.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/WC1.txt
hi how are you
how is your job
how is your family
how is your brother
how is your sister
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/sample1/insample.txt
get: `/home/hduser/Downloads/sample1/insample.txt': File exists
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/sample1/sample1.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/WC.txt
hi how are you
how is your job
how is your family
how is your brother
how is your sister
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /home/hduser/Downloads/sample1/sample1.txt
cat: `/home/hduser/Downloads/sample1/sample1.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs ls /abc
ls: Unknown command
Did you mean -ls?  This command begins with a dash.
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /abc
Found 4 items
-rw-r--r--   1 hduser supergroup         89 2023-05-11 13:57 /abc/WC.txt
-rw-r--r--   1 hduser supergroup         89 2023-05-11 13:58 /abc/WC1.txt
-rw-r--r--   1 hduser supergroup         89 2023-04-27 12:34 /abc/m.txt
-rw-r--r--   1 hduser supergroup          0 2023-04-27 12:30 /abc/t.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/sample1/insample.txt
get: `/home/hduser/Downloads/sample1/insample.txt': File exists
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/sample1/sample1.txt
get: `/home/hduser/Downloads/sample1/sample1.txt': File exists
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/sample1/sample2.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /home/hduser/Downloads/sample1/sample2.txt
ls: `/home/hduser/Downloads/sample1/sample2.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /abc
Found 4 items
-rw-r--r--   1 hduser supergroup         89 2023-05-11 13:57 /abc/WC.txt
-rw-r--r--   1 hduser supergroup         89 2023-05-11 13:58 /abc/WC1.txt
-rw-r--r--   1 hduser supergroup         89 2023-04-27 12:34 /abc/m.txt
-rw-r--r--   1 hduser supergroup          0 2023-04-27 12:30 /abc/t.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/samp
get: `/home/hduser/Downloads/sample1/insample.txt': File exists
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/samp
get: `/home/hduser/Downloads/sample1/sample1.txt': File exists
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/WC.txt /home/hduser/Downloads/samp
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /home/hduser/Downloads/sample1/sample2.t
ls: `/home/hduser/Downloads/sample1/sample2.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hadoop -getfacl /abc/
Error: No command named `-getfacl' was found. Perhaps you meant `hadoop getfacl'
hduser@bmsce-Precision-T1700:~$ hadoop fs -getfacl /abc/
# file: /abc
# owner: hduser
# group: supergroup
user::rwx
group::r-x
other::r-x
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /abc/WC1.txt /home/hduser/Desktop
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /abc/WC1.txt /home/hduser/Desktop/hduser
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/WC.txt
hi how are you
how is your job
how is your family
how is your brother
how is your sister
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -mv /abc /FFF
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /FFF
Found 4 items
-rw-r--r--   1 hduser supergroup         89 2023-05-11 13:57 /FFF/WC.txt
-rw-r--r--   1 hduser supergroup         89 2023-05-11 13:58 /FFF/WC1.txt
-rw-r--r--   1 hduser supergroup         89 2023-04-27 12:34 /FFF/m.txt
-rw-r--r--   1 hduser supergroup          0 2023-04-27 12:30 /FFF/t.txt
hduser@bmsce-Precision-T1700:~$
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /FFF/ /LLL
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /LLL
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2023-05-11 14:31 /LLL/FFF
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /LLL/FFF
Found 4 items
-rw-r--r--   1 hduser supergroup         89 2023-05-11 14:31 /LLL/FFF/WC.txt
-rw-r--r--   1 hduser supergroup         89 2023-05-11 14:31 /LLL/FFF/WC1.txt
-rw-r--r--   1 hduser supergroup         89 2023-05-11 14:31 /LLL/FFF/m.txt
-rw-r--r--   1 hduser supergroup          0 2023-05-11 14:31 /LLL/FFF/t.txt
hduser@bmsce-Precision-T1700:~$
```

Program no: **06** Program Title: **Avg/MeanMax Weather using Eclipse**

Aim: Create a Map Reduce program to

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

b) find the mean max temperature for every month

Code:

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));
  }
}
```

Output:

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -copyFromLocal /home/hduser/Desktop/1901 /rgs/test2.txt
hduser@bmsce-Precision-T1700:~$ hadoop jar /home/hduser/Desktop/AverageTemperature.jar AverageDriver /rgs/test2.txt /home/hduser/Desktop/abc1.txt
```

```
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /home/hduser/Desktop/abc1.txt
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2023-05-17 10:53 /home/hduser/Desktop/abc1.txt/_SUCCESS
-rw-r--r--   1 hduser supergroup          8 2023-05-17 10:53 /home/hduser/Desktop/abc1.txt/part-r-00000
hduser@bmsce-Precision-T1700:~$ ^C
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat /home/hduser/Desktop/abc1.txt/part-r-00000
1901 46
```

Code:

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));
  }
}
```

Output:

```
hadoop jar /home/hadoop/Desktop/meanmax.jar MeanMaxDriver /rgs/abc.txt output3
hdfs dfs -copyFromLocal /home/hadoop/Desktop/1901.txt /rgs/abc.txt
hadoop fs -ls /user/hadoop/output3
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2023-05-17 11:15 /user/hadoop/output3/_SUCCESS
-rw-r--r--   1 hadoop supergroup         74 2023-05-17 11:15 /user/hadoop/output3/part-r-00000
hadoop@bmscecse-OptiPlex-3060:/$ hadoop fs -cat /user/hadoop/output3/part-r-00000
01 4
02 0
03 7
04 44
05 100
06 168
07 219
08 198
09 141
10 100
11 19
12 3
```

Program no: **07** Program Title: **Top-N**

Aim: 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.

Code:

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> {
```

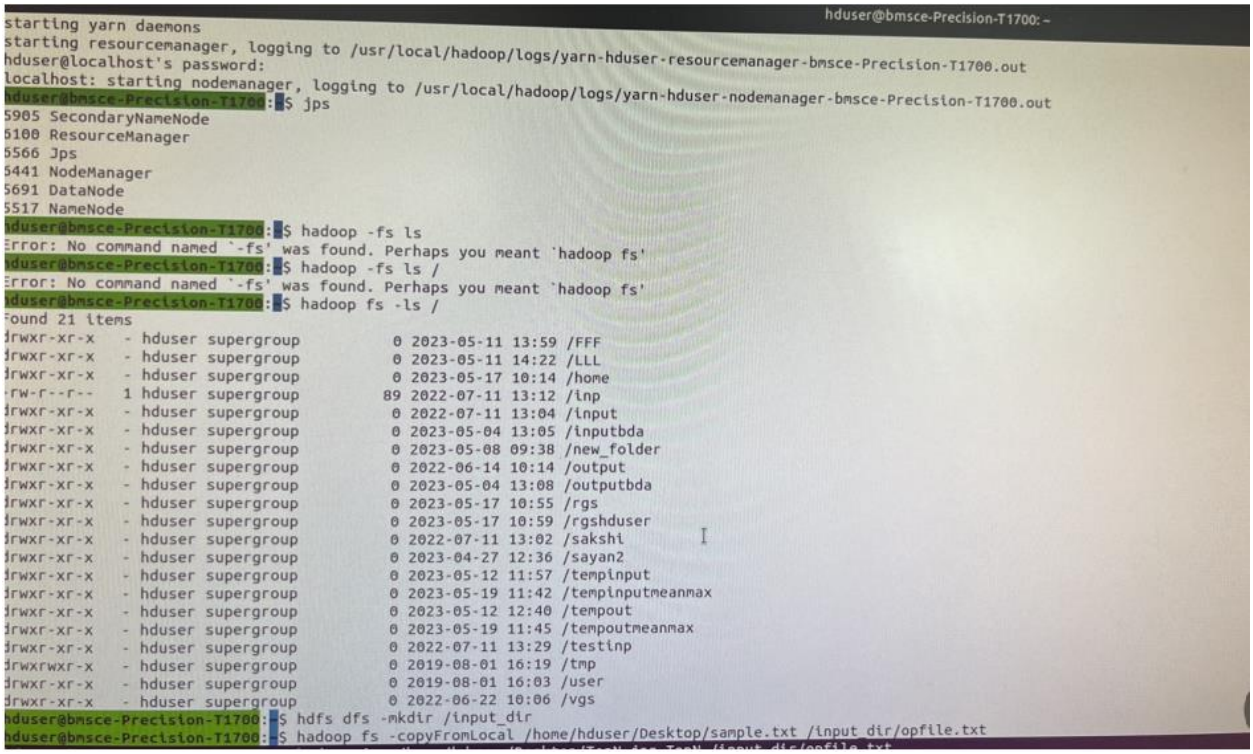
```
  private static final IntWritable one = new IntWritable(1);

  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);
    }
  }
}
```

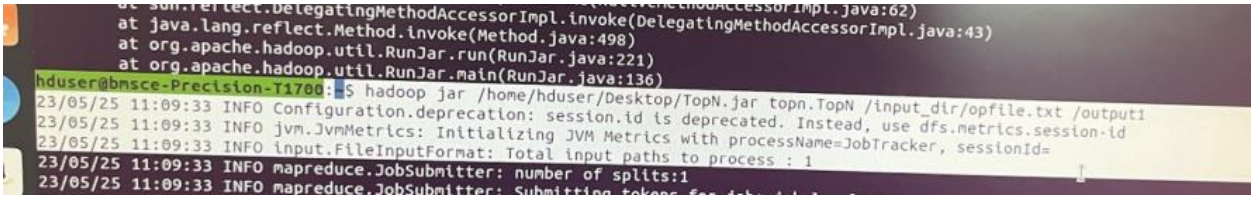
TopNReducer.class

```
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));
    }
  }
}
```

Output:





```
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /output1
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2023-05-25 11:09 /output1/_SUCCESS
-rw-r--r--   1 hduser supergroup         69 2023-05-25 11:09 /output1/part-r-00000
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat ^C
hduser@bmsce-Precision-T1700:~$ hadoop fs -ls /output1/part-r-00000
-rw-r--r--   1 hduser supergroup         69 2023-05-25 11:09 /output1/part-r-00000
hduser@bmsce-Precision-T1700:~$ hadoop fs -cat /output1/part-r-00000
how     5
your    4
is      4
brother 1
are     1
hi      1
sister  1
family  1
you     1
job     1
```

Program no: **08** Program Title: **Join Operation**

Aim: Create a Map Reduce program to demonstrating join operation.

Code:

DeptNameMapper.java

```
package MapReduceJoin;

import java.io.IOException;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapred.*;


public class DeptNameMapper extends MapReduceBase implements Mapper<LongWritable,
Text, TextPair, Text>  {


	@Override

	public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reporter reporter)

				throws IOException

	{

		String valueString = value.toString();

		String[] SingleNodeData = valueString.split("\t");

		output.collect(new TextPair(SingleNodeData[0], "0"), new
Text(SingleNodeData[1]));

	}

}
```

DeptEmpStrengthMapper.java

```
package MapReduceJoin;
```

```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FSDataOutputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

import org.apache.hadoop.io.IntWritable;

public class DeptEmpStrengthMapper extends MapReduceBase implements
Mapper<LongWritable, Text, TextPair, Text> {

	@Override
	public void map(LongWritable key, Text value, OutputCollector<TextPair, Text> output,
Reporter reporter)
				throws IOException
	{

		String valueString = value.toString();
		String[] SingleNodeData = valueString.split("\t");
		output.collect(new TextPair(SingleNodeData[0], "1"), new
Text(SingleNodeData[1]));
	}
}
```

JoinReducer.java

```
package MapReduceJoin;
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class JoinReducer extends MapReduceBase implements Reducer<TextPair, Text, Text, Text> {

	@Override
	public void reduce (TextPair key, Iterator<Text> values, OutputCollector<Text, Text> output, Reporter reporter)
		throws IOException
	{

		Text nodeId = new Text(values.next());
		while (values.hasNext()) {
			Text node = values.next();
			Text outValue = new Text(nodeId.toString() + "\t\t" + node.toString());
			output.collect(key.getFirst(), outValue);
		}
	}
}
```

JoinDriver.java

```
package MapReduceJoin;
```

```
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.mapred.lib.MultipleInputs;
import org.apache.hadoop.util.*;

public class JoinDriver extends Configured implements Tool {

	public static class KeyPartitioner implements Partitioner<TextPair, Text> {
		@Override
		public void configure(JobConf job) {}

		@Override
		public int getPartition(TextPair key, Text value, int numPartitions) {
			return (key.getFirst().hashCode() & Integer.MAX_VALUE) %
numPartitions;
		}
	}

	@Override
	public int run(String[] args) throws Exception {

		if (args.length != 3) {
			System.out.println("Usage: <Department Emp Strength input>
<Department Name input> <output>");
			return -1;
		}
```

```
        JobConf conf = new JobConf(getConf(), getClass());
        conf.setJobName("Join 'Department Emp Strength input' with 'Department Name input'");

        Path AInputPath = new Path(args[0]);
        Path BInputPath = new Path(args[1]);
        Path outputPath = new Path(args[2]);

        MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class, DeptNameMapper.class);
        MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class, DeptEmpStrengthMapper.class);

        FileOutputFormat.setOutputPath(conf, outputPath);

        conf.setPartitionerClass(KeyPartitioner.class);
        conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);

        conf.setMapOutputKeyClass(TextPair.class);

        conf.setReducerClass(JoinReducer.class);

        conf.setOutputKeyClass(Text.class);

        JobClient.runJob(conf);

        return 0;
    }
```

```
        public static void main(String[] args) throws Exception {

                int exitCode = ToolRunner.run(new JoinDriver(), args);
                System.exit(exitCode);
        }
}
```

Output:

Program no: **09** Program Title: **Scala Programming**

Aim: Program to print word count on scala shell and print “Hello world” on scala IDE.

Code:

data using sc.textFile

val data=sc.textFile("sparkdata.txt")

data.collect;

val splitdata = data.flatMap(line => line.split(" "));

splitdata.collect;

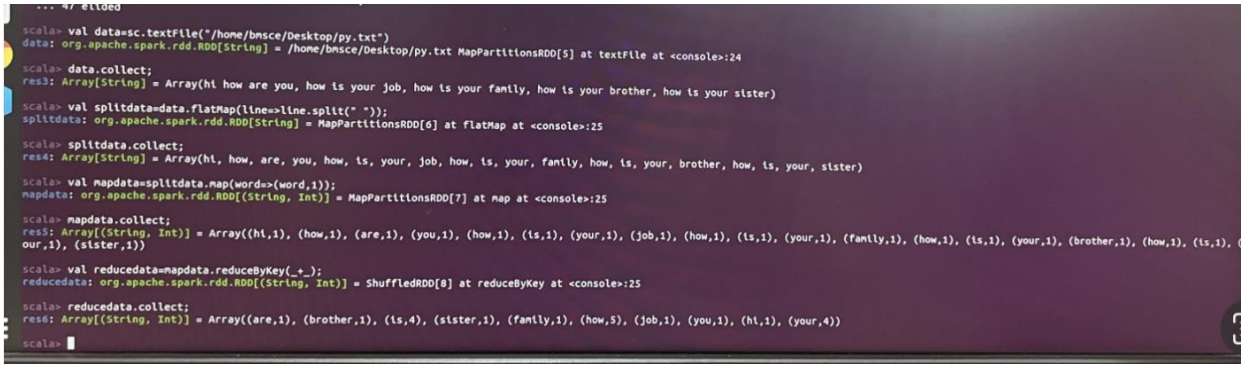
val mapdata = splitdata.map(word => (word,1));

mapdata.collect;

val reducedata = mapdata.reduceByKey(_+_);

reducedata.collect;

Output:



```
scala> val data=sc.textFile("/home/bmsce/Desktop/py.txt")
data: org.apache.spark.rdd.RDD[String] = /home/bmsce/Desktop/py.txt MapPartitionsRDD[5] at textFile at <console>:24

scala> data.collect;
res3: Array[String] = Array(hi how are you, how is your job, how is your family, how is your brother, how is your sister)

scala> val splitdata=data.flatMap(line=>line.split(" "));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[6] at flatMap at <console>:25

scala> splitdata.collect;
res4: Array[String] = Array(hi, how, are, you, how, is, your, job, how, is, your, family, how, is, your, brother, how, is, your, sister)

scala> val mapdata=splitdata.map(word=>(word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[7] at map at <console>:25

scala> mapdata.collect;
res5: Array[(String, Int)] = Array((hi,1), (how,1), (are,1), (you,1), (how,1), (is,1), (your,1), (job,1), (how,1), (is,1), (your,1), (family,1), (how,1), (is,1), (your,1), (brother,1), (how,1), (is,1), (
our,1), (sister,1))

scala> val reducedata=mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[8] at reduceByKey at <console>:25

scala> reducedata.collect;
res6: Array[(String, Int)] = Array((are,1), (brother,1), (is,4), (sister,1), (family,1), (how,5), (job,1), (you,1), (hi,1), (your,4))

scala>
```

Code:

/* Online Scala Compiler */

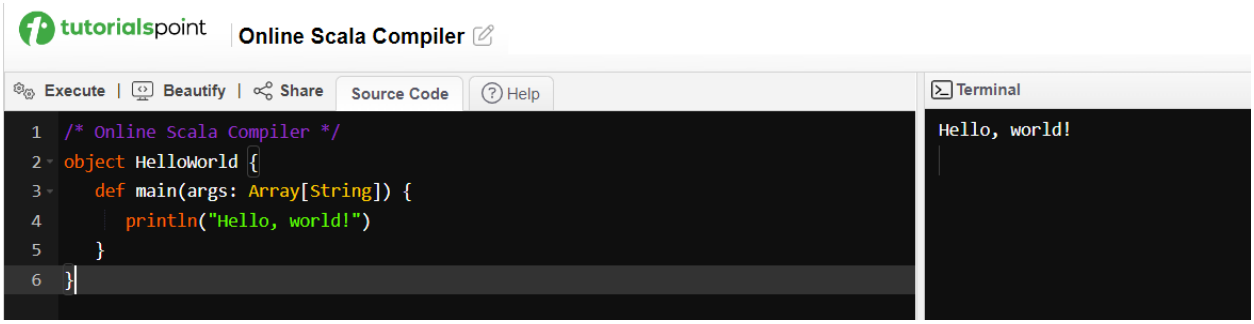
object HelloWorld {

  def main(args: Array[String]) {

    println("Hello, world!")

```
  }
}
```

Output:


tutorialspoint
Online Scala Compiler
Execute | Beautify | Share
Source Code
Help
Terminal
/* Online Scala Compiler */
object HelloWorld {
def main(args: Array[String]) {
println("Hello, world!")
}
}
Hello, world!

Program no: **10** Program Title: **Spark**

Aim: Using RDD and FlaMap 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

Code:

```
val textFile = sc.textFile(&quot;/home/bhoom/Desktop/wc.txt&quot;)

val counts = textFile.flatMap(line => line.split(&quot; &quot;)).map(word => (word,
1)).reduceByKey(_ + _)

import scala.collection.immutable.ListMap

val sorted=ListMap(counts.collect.sortWith(_._2 > _._2):_*)// sort in descending order based

on values

println(sorted)

for((k,v)<-sorted)

{

if(v>4)

{

print(k+" ")

print(v)

println()

}

}
```

Output:

```
scala> val data=sc.textFile("/home/bmsce/Desktop/py.txt")
data: org.apache.spark.rdd.RDD[String] = /home/bmsce/Desktop/py.txt MapPartitionsRDD[5] at textFile at <console>:24

scala> data.collect;
res3: Array[String] = Array(hi how are you, how is your job, how is your family, how is your brother, how is your sister)

scala> val splitdata=data.flatMap(line=>line.split(" "));
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[6] at flatMap at <console>:25

scala> splitdata.collect;
res4: Array[String] = Array(hi, how, are, you, how, is, your, job, how, is, your, family, how, is, your, brother, how, is, your, sister)

scala> val mapdata=splitdata.map(word=>(word,1));
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[7] at map at <console>:25

scala> mapdata.collect;
res5: Array[(String, Int)] = Array((hi,1), (how,1), (are,1), (you,1), (how,1), (is,1), (your,1), (job,1), (how,1), (is,1), (your,1), (family,1), (how,1), (is,1), (your,1), (brother,1), (how,1), (is,1), (y
our,1), (sister,1))

scala> val reducedata=mapdata.reduceByKey(_+_);
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[8] at reduceByKey at <console>:25

scala> reducedata.collect;
res6: Array[(String, Int)] = Array((are,1), (brother,1), (is,4), (sister,1), (family,1), (how,5), (job,1), (you,1), (hi,1), (your,4))

scala> val textFile=sc.textFile("/home/bmsce/Desktop/py.txt");
textFile: org.apache.spark.rdd.RDD[String] = /home/bmsce/Desktop/py.txt MapPartitionsRDD[10] at textFile at <console>:24

scala> val counts=textFile.flatMap(line=>line.split(" ")).map(word=>(word,1)).reduceByKey(_+_)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[13] at reduceByKey at <console>:25

scala> import scala.collection.immutable.ListMap
import scala.collection.immutable.ListMap

scala> val sorted=ListMap(counts.collect.sortWith(_._2>_._2):_*)
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(how -> 5, is -> 4, your -> 4, are -> 1, brother -> 1, sister -> 1, family -> 1, job -> 1, you -> 1, hi -> 1)

scala> println(sorted)
ListMap(how -> 5, is -> 4, your -> 4, are -> 1, brother -> 1, sister -> 1, family -> 1, job -> 1, you -> 1, hi -> 1)

scala> for((k,v)<-sorted)
     | {
     | if(v>4)
     | {
     | print(k+",")
     | print(v)
     | println()
     | }
     | }
how,5
```