

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

BIG DATA ANALYTICS (20CS6PEBDA)

Submitted by

DISHA B (1BM19CS050)

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

May-2022 to July-2022

**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 **DISHA B (1BM19CS050)**, 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. Pallavi GB
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

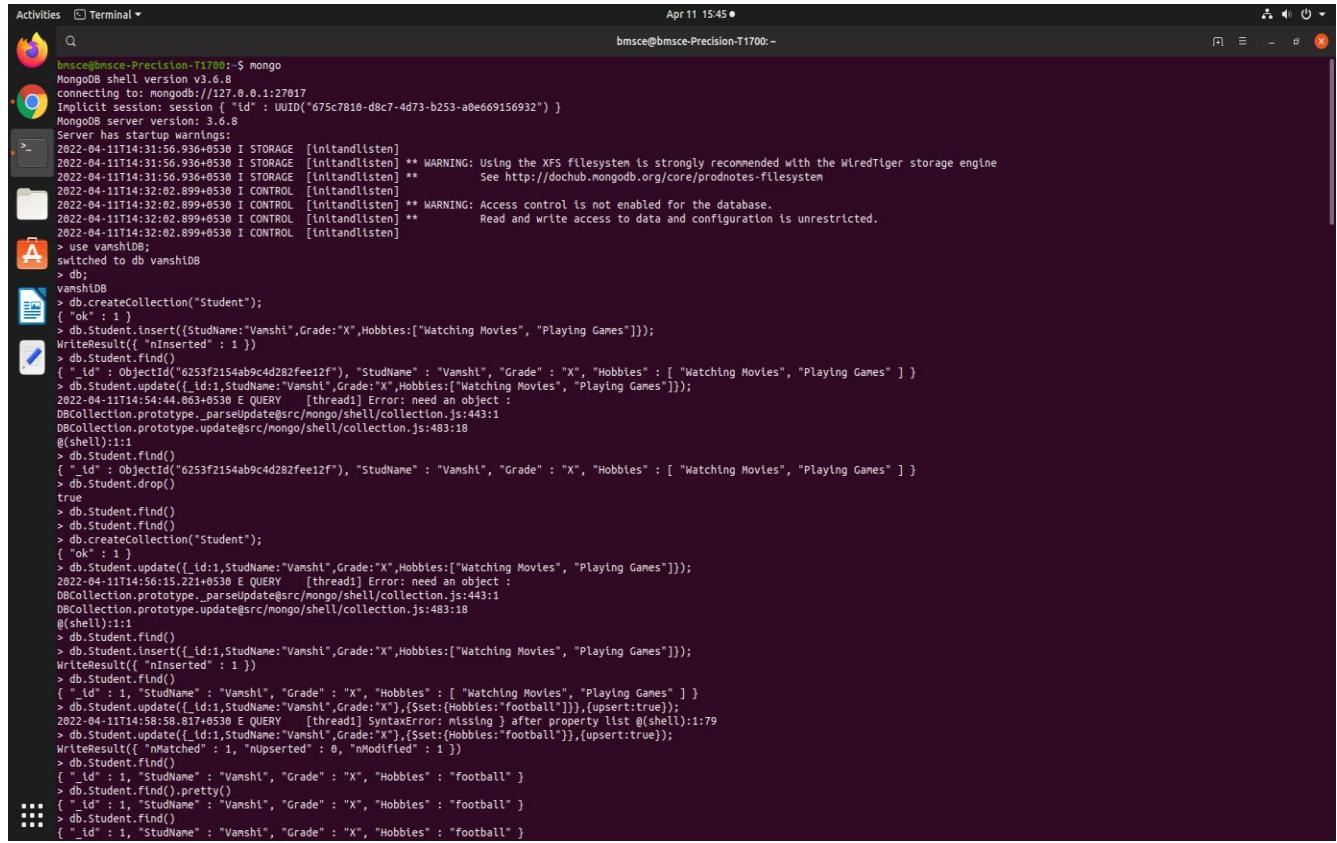
Index Sheet

| Sl. No. | Experiment Title | Page No. |
|---------|--|----------|
| 1 | MongoDB- CRUD Demonstration | 4-11 |
| 2 | Perform the following DB operations using Cassandra (Employee) | 12-15 |
| 3 | Perform the following DB operations using Cassandra (Library) | 16-19 |
| 4 | Execution of HDFS Commands for interaction with Hadoop Environment. | 20-22 |
| 5 | Screenshot of Hadoop Installed | 23 |
| 6 | 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 | 24-30 |
| 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. | 31-36 |
| 8 | Create a Map Reduce program to demonstrating join operation | 37-47 |
| 9 | Program to print word count on scala shell and print “Hello world” on scala IDE | 48 |
| 10 | 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 | 49-51 |

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 |

1. MongoDB- CRUD Demonstration



The screenshot shows a terminal window titled "Terminal" with the command "mongo" running. The session starts with the MongoDB shell version 3.6.8 connecting to the database. It displays several startup warnings related to XFS filesystem usage, access control, and configuration. The user then creates a database named "vanshiDB", selects it, and creates a collection named "Student". An object is inserted into the collection with fields "studName", "Grade", and "Hobbies". Subsequent commands show attempts to update the document using the "_id" field, which fail due to missing objects. Finally, a query is run using the "pretty" method to display the document's contents.

```
Activities Terminal * Apr 11 15:45 ●
bmsce@bmsce-Precision-T1700:~ bmsce@bmsce-Precision-T1700:~

MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("675c7810-d8c7-4d73-b253-a0e669156932") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-11T14:31:56.930+0530 I STORAGE  [initandlisten]
2022-04-11T14:31:56.930+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-11T14:31:56.930+0530 I STORAGE  [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-11T14:32:02.899+0530 I CONTROL  [initandlisten]
2022-04-11T14:32:02.899+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-11T14:32:02.899+0530 I CONTROL  [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-04-11T14:32:02.899+0530 I CONTROL  [initandlisten]

> use vanshiDB;
switched to db vanshiDB
> db
> db.createCollection("Student");
{
  "ok" : 1
}
> db.Student.insert({studName:"Vanshi",Grade:"X",Hobbies:["Watching Movies", "Playing Games"]});
WriteResult({ "nInserted" : 1 })
{
  "_id" : ObjectId("6253f2154ab9c4d282fee12f"),
  "studName" : "Vanshi",
  "Grade" : "X",
  "Hobbies" : [
    "Watching Movies",
    "Playing Games"
  ]
}
> db.Student.update([ { _id:1, StudName:"Vanshi", Grade:"X", Hobbies:["Watching Movies", "Playing Games"] } ],
  {
    "$set": {
      "Hobbies": ["Football"]
    }
  }
)
2022-04-11T14:54:44.063+0530 E QUERY    [thread1] Error: need an object :
DBCollection.prototype._parseUpdate@src/mongo/shell/collection.js:443:1
DBCollection.prototype.update@src/mongo/shell/collection.js:483:18
@shell):1:1
> db.Student.find()
{
  "_id" : ObjectId("6253f2154ab9c4d282fee12f"),
  "studName" : "Vanshi",
  "Grade" : "X",
  "Hobbies" : [
    "Watching Movies",
    "Playing Games"
  ]
}
> db.Student.drop()
true
> db.Student.find()
> db.Student.find()
> db.createCollection("Student");
{
  "ok" : 1
}
> db.Student.update([ { _id:1, StudName:"Vanshi", Grade:"X", Hobbies:["Watching Movies", "Playing Games"] } ],
  {
    "$set": {
      "Hobbies": ["Football"]
    }
  }
)
2022-04-11T14:56:15.221+0530 E QUERY    [thread1] Error: need an object :
DBCollection.prototype._parseUpdate@src/mongo/shell/collection.js:443:1
DBCollection.prototype.update@src/mongo/shell/collection.js:483:18
@shell):1:1
> db.Student.find()
{
  "_id" : 1,
  "StudName" : "Vanshi",
  "Grade" : "X",
  "Hobbies" : [
    "Watching Movies",
    "Playing Games"
  ]
}
> db.Student.update([ { _id:1, StudName:"Vanshi", Grade:"X", $set:{Hobbies:"football"} } ],{upsert:true});
2022-04-11T14:58:58.817+0530 E QUERY    [thread1] SyntaxError: missing ) after property list @shell):1:79
> db.Student.update([ { _id:1, StudName:"Vanshi", Grade:"X", $set:{Hobbies:"football"} } ],{upsert:true});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find()
{
  "_id" : 1,
  "StudName" : "Vanshi",
  "Grade" : "X",
  "Hobbies" : "football"
}
> db.Student.pretty()
{
  "_id" : 1,
  "StudName" : "Vanshi",
  "Grade" : "X",
  "Hobbies" : "football"
}
> db.Student.find()
{
  "_id" : 1,
  "StudName" : "Vanshi",
  "Grade" : "X",
  "Hobbies" : "football"
}
```

Activities Terminal Apr 11 15:45 bmsce@bmsce-Precision-T1700: ~

```
> db.Student.find()
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find({StudentName:"Vanshi"},{Grade:0,Hobbies:1})
2022-04-11T15:06:24.118+0530 E QUERY [thread1] SyntaxError: missing : after property id @shell:1:35
> db.Student.find({StudentName:"Vanshi"},{_id:1,Grade:0,Hobbies:1})
2022-04-11T15:06:24.118+0530 E QUERY [thread1] SyntaxError: missing : after property id @shell:1:35
> db.Student.find({StudentName:"Vanshi"},{_id:1,Grade:0,Hobbies:1})
2022-04-11T15:06:24.118+0530 E QUERY [thread1] SyntaxError: missing : after property id @shell:1:35
> db.Student.find({StudentName:{}}, {_id:0,Grade:1,Hobbies:1})
2022-04-11T15:08:13.713+0530 E QUERY [thread1] SyntaxError: invalid property id @shell:1:29
> db.Student.find({},{_id:0,Grade:1,Hobbies:1});
{ "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find({},{_id:1,Grade:1,Hobbies:1});
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find(Grade:{Seq:"VII"}).pretty();
> db.Student.find(Grade:{Seq:"X"}).pretty();
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find(Grade:{Seq:"X"}).pretty();
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find({_id:1,StudentName:"Sharat",Grade:"X"},Hobbies:["Swimming", "Badminton"]));
> db.writeResult({_id:1,StudentName:"Sharat",Grade:"X"},Hobbies:["Swimming", "Badminton"]));
> db.Student.find()
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
{ "_id" : 2, "StudentName" : "Sharat", "Grade" : "X", "Hobbies" : [ "Swimming", "Badminton" ] }
> db.Student.find(Grade:{Seq:"X"}).pretty();
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find(Hobbies:{$ne:["football", "Golf"]}).pretty();
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find(StudentName:/V/).pretty();
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
> db.Student.find(StudentName:/S/).pretty();
{ "_id" : 2,
  "StudentName" : "Sharat",
  "Grade" : "X",
  "Hobbies" : [
    "Swimming",
    "Badminton"
  ]
}
> db.Student.find({StudentName:/ S/}).pretty();
> db.Student.find({StudentName:/ /}).pretty();
[1]+ Stopped mongo
bmsce@bmsce-Precision-T1700: ~ use vanshidb
```

Command 'use' not found, did you mean:

```
command 'nuse' from deb nuse (3.0.2-2ds1-1)
command 'nse' from deb ns2 (2.35dfsg3)
command 'fuse' from deb fuse-emulator-gtk (1:5.7+dfsg1-1)
command 'fuse' from deb fuse-emulator-sdl (1:5.7+dfsg1-1)
command 'ase' from deb ase (3.17.0-2ubuntu1)
```

Try: sudo apt install <deb name>

```
bmsce@bmsce-Precision-T1700: ~ mongo
```

Activities Terminal Apr 11 15:45 bmsce@bmsce-Precision-T1700: ~

```
bmsce@bmsce-Precision-T1700: ~ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("5ced6709e-e96c-4120-ad2-b63c2133d6eb" ) }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-11T14:31:56.934+0530 I STORAGE [initandlisten] 
2022-04-11T14:31:56.934+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-11T14:32:02.899+0530 I CONTROL [initandlisten]
2022-04-11T14:32:02.899+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-11T14:32:02.899+0530 I CONTROL [initandlisten] ** WARNING: Read and write access to data and configuration is unrestricted.
A
> use vanshidb
switched to db vanshidb
> show dbs;
admin 0.000GB
config 0.000GB
local 0.000GB
vansh 0.000GB
vanshDB 0.000GB
> db.Student.find({Studentname:/S/}).pretty();
> db.Student.find({Studentname:/S/}).pretty();
...
> db.Student.find({Studentname:/V/}).pretty();
> db.listCollections()
2022-04-11T15:22:11.516+0530 E QUERY [thread1] TypeError: db.listCollections is not a function :
@shell:1::1
@db.listCollections()
2022-04-11T15:22:20.556+0530 E QUERY [thread1] TypeError: db.listCollections is not a function :
@shell:1::1
@db.listCollections()
2022-04-11T15:22:36.365+0530 E QUERY [thread1] TypeError: db.listCollections is not a function :
@shell:1::1
@collections
2022-04-11T15:22:42.557+0530 E QUERY [thread1] ReferenceError: collections is not defined :
@shell:1::1
> show collections
> ;
> show collections;
> db.Student.find({Studentname:/V/}).pretty();
2022-04-11T15:23:26.706+0530 E QUERY [thread1] SyntaxError: expected expression, got '^' @shell:1:26
> db.student.find()
> use vanshidb
switched to db vanshidb
> use vanshidb;
switched to db vanshidb
> db.student.find()
> show collections;
Student
> db.Student.find()
{ "_id" : 1, "StudentName" : "Vanshi", "Grade" : "X", "Hobbies" : "football" }
{ "_id" : 2, "StudentName" : "Sharat", "Grade" : "X", "Hobbies" : [ "Swimming", "Badminton" ] }
```

Activities Terminal Apr 11 15:34 • bmscse@bmscse-Precision-T1700:~

```

db.Students.update({ "Grade": "VII"}, { $set: { "Hobbies": "football" } })
> db.Student.count()
2
> db.Student.find().sort({studName:-1}).pretty();
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["Swimming", "Badminton"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Student.find().sort({studName:-1}).pretty();
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["Swimming", "Badminton"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Students.update({_id:3},{$set:{stipend:85522}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Students.update({_id:2},{$set:{stipend:85522}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Students.update({_id:2},{$set:{stipend:7000}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Students.update({_id:2},{$set:{stipend:7000}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"], "stipend": 7000}]
> db.Students.update({_id:2},{$unset:{stipend:1}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Student.update({_id:2},{$set:{Grade:null}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": null, "Hobbies": ["Swimming", "Badminton"]}]
> []

```

db.Students.find({Grade:"VII"}).limit(3).pretty();

Sort the document in Ascending order

db.Students.find().sort({studName:1}).pretty();

Note:
For descending order : db.Students.find().sort({studName:-1}).pretty();

To Skip the 1st two documents from the Students Collections

db.Students.find().skip(2).pretty()

XII. Create a collection by name "food" and add to each document add a "fruits" array

db.food.insert({ _id1, fruits:["grapes","mango","apple"] })
db.food.insert({ _id2, fruits:["grapes","mango","cherry"] })
db.food.insert({ _id3, fruits:["bananas","mango"] })

To find those documents in the "Food" collection which have the "fruits array" constitute of "grapes", "mango" and "apple".

Activities Terminal Apr 11 15:45 • bmscse@bmscse-Precision-T1700:~

```

{
    "_id": 2,
    "studName": "Sharat",
    "Grade": "XI",
    "Hobbies": [
        "Swimming",
        "Badminton"
    ]
}
> db.Students.update({_id:3},{$set:{stipend:85522}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Students.update({_id:2},{$set:{stipend:85522}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Students.update({_id:2},{$set:{stipend:7000}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Students.update({_id:2},{$set:{stipend:7000}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Student.update({_id:2},{$set:{Grade:null}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": null, "Hobbies": ["Swimming", "Badminton"]}]
> db.Student.update({_id:2},{$set:{stipend:7000}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"], "stipend": 7000}]
> db.Student.update({_id:2},{$set:{stipend:7000}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": "XI", "Hobbies": ["Swimming", "Badminton"]}]
> db.Student.update({_id:2},{$set:{Grade:null}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Student.find()
[{"_id": 1, "studName": "Vamshi", "Grade": "X", "Hobbies": ["football"]},
 {"_id": 2, "studName": "Sharat", "Grade": null, "Hobbies": ["Swimming", "Badminton"]}]
> []

```

```

Activities Terminal ▾ Apr 11 15:45
bmse@bmse-Precision-T1700:~ bmsc@bmse-Precision-T1700:~ - + x
     
bmse@bmse-Precision-T1700:~
db.Student.update({$_id:2},{set:{Grade:null}})
WriteResult({"nMatched":1,"nUpserted":0,"nModified":1})
> db.Student.find()
[{"_id": 1, "studname": "Vanshi", "Grade": "X", "Hobbies": ["Swimming", "Badminton"] }
 {"_id": 2, "studname": "Sharat", "Grade": null, "Hobbies": ["Swimming", "Badminton"] }
]
> db.Student.count({Grade:"VI"})
2022-04-11T15:33:44.666+0530 E QUERY [thread1] SyntaxError: illegal character @shell::i:24
> db.Student.count({Grade:"X"})
2022-04-11T15:33:44.666+0530 E QUERY [thread1] SyntaxError: illegal character @shell::i:24
> db.Student.count({Grade:"X"})
1
> db.Student.find()
[{"_id": 1, "studname": "Vanshi", "Grade": "X", "Hobbies": ["Swimming", "Badminton"] }
 {"_id": 2, "studname": "Sharat", "Grade": null, "Hobbies": ["Swimming", "Badminton"] }
]
> db.Student.find().sort({_studname:1}).pretty();
[
  {
    "_id": 2,
    "studname": "Sharat",
    "Grade": null,
    "Hobbies": [
      "Swimming",
      "Badminton"
    ]
  }
]
{"_id": 1, "studname": "Vanshi", "Grade": "X", "Hobbies": ["football"] }
> db.createCollection("veggies");
[ok: 1]
> db.veggies.insert({_id:1,vegetables:["cucumber","beetroot",'carrot']})
2022-04-11T15:38:04.363+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @shell::i:21
> db.veggies.insert({_id:1,vegetables:["cucumber","beetroot",'carrot']});
2022-04-11T15:38:11.866+0530 E QUERY [thread1] SyntaxError: missing ) after argument list @shell::i:21
> db.veggies.insert({_id:1,vegetables:["cucumber","beetroot",'carrot']});
WriteResult({"nInserted": 1})
> db.veggies.insert({_id:1,vegetables:["cabbage"]});
WriteResult({
  "nInserted": 0,
  "writeError": {
    "code": 11000,
    "errmsg": "E11000 duplicate key error collection: vamsiDB.veggies index: _id_ dup key: { : 1.0 }"
  }
})
> db.veggies.insert({_id:2,vegetables:["cabbage"]});
WriteResult({"nInserted": 1})
> db.veggies.find({'vegetables': 'beetroot'})
[{"_id": 1, "vegetables": ["cucumber", "beetroot", "carrot"] }
]
> db.veggies.find({'vegetables':{$size:3}})
[{"_id": 1, "vegetables": ["cucumber", "beetroot", "carrot"] }
]
> db.veggies.find({$slice:2})
2022-04-11T15:43:03.418+0530 E QUERY [thread1] SyntaxError: illegal character @shell::i:25
> db.veggies.find({_id:1}){'vegetables':{$slice:2}}
[{"_id": 1, "vegetables": ["cucumber", "beetroot"] }
]
> db.veggies.find({fruits:{$all:["cabbage"]}})
2022-04-11T15:44:29.849+0530 E QUERY [thread1] SyntaxError: illegal character @shell::i:31
> db.veggies.find({fruits:{$all:["cabbage"]}})
[[]]

```

```

Activities Terminal ▾ Apr 18 15:35
bmse@bmse-Precision-T1700:~ script
Script: /tmp/file1.sh type: script
bmse@bmse-Precision-T1700:~ $ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id": UUID("ff9ec7b5-b748-4fd6-9e83-5204d761999e") }
MongoDB server version: 3.6.8
MongoDB shell version: 3.6.8
MongoDB shell version: 3.6.8
2022-04-18T14:01:29.318+0530 I STORAGE [initandlisten]
2022-04-18T14:01:29.318+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten]
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
mymonitors 0.000GB
studs 0.000GB
studs 0.000GB
test 0.000GB
> use studDB
switched to db studDB
> db.collections
student
> db.student.drop()
true
> db.create.collection("Student")
2022-04-18T14:18:56.526+0530 E QUERY [thread1] TypeError: db.create.collection is not a function :
($shell)::i:1
> db.create.Collection("Student")
2022-04-18T14:18:57.550+0530 E QUERY [thread1] TypeError: db.create.Collection is not a function :
($shell)::i:1
> db.createCollection("Student")
> db.student.insert({_id:3, name:"Jeevan", usn:"ibm19cs084", semester:6, Dept_name:"Computer Science", Cgpa:8, hobbies:["Pokiri", "Chokri"]})
WriteResult({"nInserted": 1})
> db.student.find()
[{"_id": 3, "name": "Jeevan", "usn": "ibm19cs084", "semester": 6, "Dept_name": "Computer Science", "Cgpa": 8, "hobbies": [ "Pokiri", "Chokri" ] }
]
> db.student.insert({_id:4, name:"Shrath", usn:"ibm19cs076", semester:6, Dept_name:"Computer Science", Cgpa:8, hobbies:["Rocket", "Mango"]})
WriteResult({"nInserted": 1})
> db.student.insert({_id:5, name:"Vanshi", usn:"ibm19cs080", semester:6, Dept_name:"Computer Science", Cgpa:8, hobbies:["Rocket", "Noobs"]})
WriteResult({"nInserted": 1})
> db.student.find({Dept_name:"Computer science"})
> db.student.find({Dept_name:"Computer Science"})
[{"_id": 3, "name": "Jeevan", "usn": "ibm19cs084", "semester": 6, "Dept_name": "Computer Science", "Cgpa": 8, "hobbies": [ "Pokiri", "Chokri" ] }
 {"_id": 4, "name": "Shrath", "usn": "ibm19cs076", "semester": 6, "Dept_name": "Computer Science", "Cgpa": 8, "hobbies": [ "Rocket", "Mango" ] }
 {"_id": 5, "name": "Vanshi", "usn": "ibm19cs080", "semester": 6, "Dept_name": "Computer Science", "Cgpa": 8, "hobbies": [ "Rocket", "Noobs" ] }
]
> db.students.aggregate([{$match:{Dept_name:"Computer Science"}}, {$group:{sem:$semester, avgCgpa:$Cgpa}}])
assert: command failed: {
  "ok": 0,
  "errmsg": "The field 'sem' must be an accumulator object",
  "code": 4034,
  "codeName": "Location4034"
}

```

```
Activities Terminal ▾ Apr 18 15:35
bmsc@bmsc-Precision-T1700:~
```

```
[{"_id": 4, "name": "Shrath", "usn": "1bm19cs076", "semester": 6, "Dept_name": "Computer Science", "Gpa": 8, "hobbies": ["Rocket", "Mango"]}, {"_id": 5, "name": "Vamsi", "usn": "1bm19cs080", "semester": 6, "Dept_name": "Computer Science", "Gpa": 8, "hobbies": ["Rocket", "Noobs"]}]> db.students.aggregate([{$match:{Dept_name:"Computer Science"}}, {$group:{sem:$semester, avgGpa:$avgGpa}}])assert: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:39:54.336+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:40:28.271+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:40:28.271+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:42:15.905+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:42:15.905+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:48:06.353+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:
```

```
Activities Terminal ▾ Apr 18 15:35
bmsc@bmsc-Precision-T1700:~
```

```
> db.students.aggregate([{$match:{Dept_name:"Computer Science"}}, {$group:{_id:$_id, sem:$semester, avgGpa:$avgGpa}}])assert: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:42:15.905+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:42:15.905+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:42:15.905+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:_getErrorWithCode@src/mongo/shell/utils.js:25:13doassert@src/mongo/shell/assert.js:16:14assert.commandWorked@src/mongo/shell/assert.js:403:5DB.prototype._runAggregate@src/mongo/shell/db.js:260:9DBCollection.prototype.aggregate@src/mongo/shell/collection.js:1212:12@{shell}:::12022-04-18T14:48:06.353+0530 E QUERY [thread1] Error: command failed: { "ok": 0, "errmsg": "The field 'sem' must be an accumulator object", "code": 40234, "codeName": "Location40234"}: aggregate failed:
```

```
Activities Terminal Apr 18 15:35
bmsce@bmsce-Precision-T1700: ~
g(shell):::1
2022-04-18T14:48:06.353+0530 E QUERY [thread1] Error: command failed: {
  "ok" : 0,
  "errns" : "The field 'sum' must be an accumulator object",
  "code" : 40234,
  "codeName" : "Location40234"
}
> db.student.aggregate([
  { $group: { _id: "$Dept_name", avgCgpa: { $avg: "$Cgpa" } } }
])
> db.student.insert([{"_id": 5, name:"Vanshee", usn:"ibm19cs080", semester:4, Dept_name:"Computer Science", Cgpa:9, hobbies:["Rocket", "Noobs"]})
WriteResult({
  "nInserted" : 1,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: studDB.student index: _id_ dup key: { : 5.0 }"
  }
})
> db.student.insert([{"_id": 6, name:"Vamshee", usn:"ibm19cs080", semester:4, Dept_name:"Computer Science", Cgpa:9, hobbies:["Rocket", "Noobs"]})
WriteResult({
  "nInserted" : 1,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: studDB.student index: _id_ dup key: { : 5.0 }"
  }
})
> db.student.aggregate([{$match:{Dept_name:"Computer Science"}}, {$group:{_id:"$semester", avgCgpa:{$avg:"$Cgpa"}}})
[{"_id" : 6, "avgCgpa" : 8}
]
> db.student.insert([{"_id": 5, name:"Vanshee", usn:"ibm19cs080", semester:4, Dept_name:"Computer Science", Cgpa:9, hobbies:["Rocket", "Noobs"]})
WriteResult({
  "nInserted" : 1,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: studDB.student index: _id_ dup key: { : 5.0 }"
  }
})
> db.student.insert([{"_id": 6, name:"Vamshee", usn:"ibm19cs080", semester:4, Dept_name:"Computer Science", Cgpa:9, hobbies:["Rocket", "Noobs"]})
WriteResult({
  "nInserted" : 1,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: studDB.student index: _id_ dup key: { : 5.0 }"
  }
})
> db.student.aggregate([{$match:{Dept_name:"Computer Science"}}, {$group:{_id:"$semester", avgCgpa:{$avg:"$Cgpa"}}, {$match:{avgCgpa:{$gt:8.5}}}}
])
[{"_id" : 4, "avgCgpa" : 8}
]
> db.student.aggregate([{$match:{Dept_name:"Computer Science"}}, {$group:{_id:"$semester", avgCgpa:{$avg:"$Cgpa"}}, {$match:{avgCgpa:{$gt:7.5}}}}
])
[{"_id" : 4, "avgCgpa" : 9}
]
> db.student.aggregate([{$match:{Dept_name:"Computer Science"}}, {$group:{_id:"$semester", avgCgpa:{$avg:"$Cgpa"}}, {$match:{avgCgpa:{$gt:7}}}}
])
[{"_id" : 6, "avgCgpa" : 8}
]
> mongoexport --host localhost --db Student --collection airlines --csv --out
2022-04-18T14:53:09.084+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> db
> studDB
> mongoexport --host localhost --db studDB --collection student --csv --out /home
2022-04-18T14:54:00.371+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --csv --out
2022-04-18T14:54:08.515+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --csv --out /home/out.txt;
2022-04-18T14:54:08.515+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --csv --out /home/out.txt;
2022-04-18T14:55:03.908+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --csv --out /home/out.txt;
2022-04-18T14:55:03.908+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --csv --out /home/out.txt;
2022-04-18T14:55:28.750+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
2022-04-18T14:55:46.181+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --csv --out /home/out.csv;
2022-04-18T14:57:36.277+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --fields _id --csv --out /home/out.csv;
2022-04-18T14:57:59.430+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --host localhost --db studDB --collection student --fields _id --csv --out /home/out.csv;
2022-04-18T14:59.031+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
```

```
Activities Terminal Apr 18 15:35
bmsce@bmsce-Precision-T1700: ~
bmsce@bmsce-Precision-T1700: ~
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T14:01:29.318+0530 I STORAGE [initandlisten] 
2022-04-18T14:01:29.318+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --db studDB --collection student --fields _id --type=csv --out /home/out.csv;
2022-04-18T14:01:36.775+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --db studDB --collection student --fields _id --type=csv --out /home/out.csv;
> mongoexport --db studDB --collection student --fields _id --type=csv --out /home/out.csv;
by
bmsce@bmsce-Precision-T1700: ~
bmsce@bmsce-Precision-T1700: ~
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T14:01:29.318+0530 I STORAGE [initandlisten]
2022-04-18T14:01:29.318+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-18T14:01:29.318+0530 I STORAGE [initandlisten] See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
> use studDB
switched to db studDB
> mongoexport --db studDB --collection student --fields _id --type=csv --out homeout.csv;
2022-04-18T15:02:34.472+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --db studDB --collection student --fields _id --type=csv --out homeout.test.csv;
2022-04-18T15:03:26.760+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --db studDB --collection student --fields _id --type=csv --out homeout_test.csv;
2022-04-18T15:03:26.760+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> mongoexport --db studDB --collection student --fields _id --type=csv --out homeout_test.csv;
2022-04-18T15:12:18.456+0530 E QUERY [thread1] SyntaxError: missing ; before statement @shell:::14
> db.createCollection("Bank")
[{"ok": 1}
]
> db.Bank.insert({_id:1, accNo:121212, name:"Ramesha", bal:1000, type:"S"})
WriteResult({ "n": 1, "ok": 1 })
> db.Bank.insert({_id:2, accNo:121213, name:"Ramesha", bal:1000, type:"S"})
WriteResult({ "n": 1, "ok": 1 })
> db.Bank.insert({_id:3, accNo:121214, name:"Ramesha", bal:1000, type:"S"})
WriteResult({ "n": 1, "ok": 1 })
> db.Bank.find()
{ "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S" }
{ "_id": 2, "accNo": 121213, "name": "Ramesha", "bal": 1000, "type": "S" }
{ "_id": 3, "accNo": 121214, "name": "Ramesha", "bal": 1000, "type": "S" }
> db.Bank.update({_id:1}, {$push:{nominee:{n1:"Son1", n2:"Son2"}}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{ "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" } ] }
{ "_id": 2, "accNo": 121213, "name": "Ramesha", "bal": 1000, "type": "S" }
{ "_id": 3, "accNo": 121214, "name": "Ramesha", "bal": 1000, "type": "S" }
> db.Bank.update({_id:1}, {$push:{nominee:"Son1"}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{ "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" }, "Son1" ] }
```

```
Activities Terminal Apr 18 15:35
bmsce@bmisce-Precision-T1700:~ bmsce@bmisce-Precision-T1700:~ Apr 18 15:35
> mongoexport -d studDB -collection student --fields _id --type=csv --out home\out.csv;
2022-04-18T14:01:29.318+0530 E QUERY [thread1] SyntaxError: missing ; before statement @($shell):1:14
> mongoexport -d studDB -collection student --fields _id --type=csv --out home\out.csv;
2022-04-18T14:01:37.479+0530 E QUERY [thread1] SyntaxError: missing ; before statement @($shell):1:14
> mongoexport -d db studDB -collection student --fields _id --type=csv --out home\out.csv;`^C
bye
bmsce@bmisce-Precision-T1700:~ $ ^C
MongoDB shell version v3.6.8
Implicit session: session { "id": UUID("1ee86771-069e-4ef0-b098-7ec1647969e0") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T14:01:38.129+0530 I STORAGE [initandlisten]
2022-04-18T14:01:29.318+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten]
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T14:01:33.775+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
> use studDB
switched to db studDB
> mongoexport -d db studDB -collection student --fields _id --type=csv --out home\out.csv;
2022-04-18T14:02:34.472+0530 E QUERY [thread1] SyntaxError: missing ; before statement @($shell):1:14
> mongoexport -d db studDB -collection student --fields _id --type=csv --out home\out_test.csv;
2022-04-18T14:01:36.760+0530 E QUERY [thread1] SyntaxError: missing ; before statement @($shell):1:14
> mongoexport -d db studDB -collection student --fields _id --type=csv --out Home\out_test.csv;
2022-04-18T15:12:09.833+0530 E QUERY [thread1] SyntaxError: missing ; before statement @($shell):1:14
> mongoexport -d db studDB -collection student --fields _id --type=csv --out Home\out_test.csv;
2022-04-18T15:12:18.456+0530 E QUERY [thread1] SyntaxError: missing ; before statement @($shell):1:14
> db.createCollection("Bank")
{
  "_id": 1
}
> db.Bank.insert({_id:1, accNo:121212, name:"Ramesha", bal:1000, type:"S"})
WriteResult({ "nInserted": 1 })
> db.Bank.insert({_id:2, accNo:121213, name:"Rameshaa", bal:1000, type:"S"})
WriteResult({ "nInserted": 1 })
> db.Bank.insert({_id:3, accNo:121214, name:"Rameshaaa", bal:1000, type:"S"})
WriteResult({ "nInserted": 1 })
> db.Bank.update({_id:1}, {push:{nominee:{n1:"Son1", n2:"Son2"}}})
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S" }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {push:{nominee:{n1:"Son1", n2:"Son2"}}})
...
2022-04-18T15:23:035+0530 E QUERY [thread1] SyntaxError: missing } after property list @($shell):1:63
> db.Bank.update({_id:1}, {push:{nominee:{n1:"Son1", n2:"Son2"}}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" } ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {push:{nominee:"Son1"}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" }, "Son1" ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {push:{nominee:"Son1"}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" }, "Son1" ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {pop:{nominee:"Son1"}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0, "writeError": { "code": 9, "errmsg": "Expected a number in: nominee: \\\"Son1\\\""} })
...
}> db.Bank.update({_id:2}, {pop:{nominee:"Son1"}})
WriteResult({ "nMatched": 0, "nUpserted": 0, "nModified": 0, "writeError": { "code": 9, "errmsg": "Expected a number in: nominee: \\\"Son1\\\""} })
...
> db.Bank.update({_id:2}, {pop:{nominee:"Son1"}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ "Son1" ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S", "nominee": [ ] }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {pop:{nominee:"Son1"}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.createIndex({name:1})
{
  "createdCollectionAutomatically": false,
  "numIndexesBefore": 1,
  "numIndexesAfter": 2,
  "ok": 1
}
> db.Bank.find()
```

```
Activities Terminal Apr 18 15:35
bmsce@bmisce-Precision-T1700:~ bmsce@bmisce-Precision-T1700:~ Apr 18 15:35
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" } ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {push:{nominee:"Son1"}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" }, "Son1" ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {push:{nominee:"Son1"}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ { "n1": "Son1", "n2": "Son2" }, "Son1" ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {push:{nominee:"Son1"}, {n:1}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
...
}> db.Bank.update({_id:2}, {pop:{nominee:"Son1"}, {n:1}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ "Son1" ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.update({_id:1}, {pop:{nominee:"Son1"}, {n:1}})
WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
> db.Bank.find()
{
  "_id": 1, "accNo": 121212, "name": "Ramesha", "bal": 1000, "type": "S", "nominee": [ "Son1" ] }
{
  "_id": 2, "accNo": 121213, "name": "Rameshaa", "bal": 1000, "type": "S" }
{
  "_id": 3, "accNo": 121214, "name": "Rameshaaa", "bal": 1000, "type": "S" }
db.Bank.createIndex({name:1})
{
  "createdCollectionAutomatically": false,
  "numIndexesBefore": 1,
  "numIndexesAfter": 2,
  "ok": 1
}
```

2. 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 Employee

1. Create a key space by name Employee

```
cqlsh> describe keyspaces;
system_schema system      system_distributed
system_auth   test_keyspace system_traces

cqlsh> create keyspace Employee with replication={'class':'SimpleStrategy','replication_factor':2};
cqlsh> describe keyspace Employee;
CREATE KEYSPACE employee WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '2'} AND durable_writes = true;

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(cluster_col text,Emp_id int,Emp_Name text,Designation text,Date_of_Joining timestamp,Salary int,Dept_Name text,primary key(cluster_col,Salary)) WITH CLUSTERING ORDER BY(Salary DESC);
cqlsh:employee> select*from employee;
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured table employee"
cqlsh:employee> select*from employee_info;

cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name
-----+-----+-----+-----+-----+-----+-----
```

3. Insert the values into the table in batch

```
(0 rows)
cqlsh:employee> BEGIN BATCH
... INSERT INTO Employee_Info(cluster_col,emp_id,emp_name,designation,Date_of_Joining,Salary,Dept_Name) VALUES ('xyz',1,'Ravi','MTS','2020-08-24',12000,'TESTING');
... INSERT INTO Employee_Info(cluster_col,emp_id,emp_name,designation,Date_of_Joining,Salary,Dept_Name) VALUES ('xyz',2,'Vamshi','MANAGER','2021-03-28',50000,'DEVELOPMENT');
... INSERT INTO Employee_Info(cluster_col,emp_id,emp_name,designation,Date_of_Joining,Salary,Dept_Name) VALUES ('xyz',121,'Kiran','SDE','2019-04-21',10000,'PRODUCTION');
... INSERT INTO Employee_Info(cluster_col,emp_id,emp_name,designation,Date_of_Joining,Salary,Dept_Name) VALUES ('xyz',3,'Ramesh','ANALYST','2020-05-07',20000,'QUALITY');
... APPLY BATCH;
cqlsh:employee> SELECT*FROM Employee_Info;

cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name
-----+-----+-----+-----+-----+-----+-----+
xyz | 50000 | 2021-03-19 18:30:00.000000+0000 | DEVELOPMENT | MANAGER | 2 | Vamshi
xyz | 20000 | 2020-05-06 18:30:00.000000+0000 | QUALITY | ANALYST | 3 | Ramesh
xyz | 12000 | 2020-08-23 18:30:00.000000+0000 | TESTING | MTS | 1 | Ravi
xyz | 10000 | 2019-04-20 18:30:00.000000+0000 | PRODUCTION | SDE | 121 | Kiran
(4 rows)
```

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

```
cqlsh:employee> update Employee_Info SET emp_name='karthik',dept_name='Compliance' where cluster_col='xyz' and salary=10000 IF emp_id=121;
[applied]
-----
True

cqlsh:employee> SELECT*FROM Employee_Info;

cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name | projects
-----+-----+-----+-----+-----+-----+-----+-----+
xyz | 50000 | 2021-03-19 18:30:00.000000+0000 | DEVELOPEMENT | MANAGER | 2 | Vamshi | {'AI', 'DS'}
xyz | 20000 | 2020-05-06 18:30:00.000000+0000 | QUALITY | ANALYST | 3 | Ramesh | {'DEVOPS'}
xyz | 12000 | 2020-08-23 18:30:00.000000+0000 | TESTING | MTS | 1 | Ravi | {'ML'}
xyz | 10000 | 2019-04-20 18:30:00.000000+0000 | Compliance | SDE | 121 | karthik | {'QUANTUM COMPUTING'}
```

5. Sort the details of Employee records based on salary

```
cqlsh:employee> SELECT*FROM Employee_Info;

cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name | projects
-----+-----+-----+-----+-----+-----+-----+-----+
xyz | 50000 | 2021-03-19 18:30:00.000000+0000 | DEVELOPEMENT | MANAGER | 2 | Vamshi | null
xyz | 20000 | 2020-05-06 18:30:00.000000+0000 | QUALITY | ANALYST | 3 | Ramesh | null
xyz | 12000 | 2020-08-23 18:30:00.000000+0000 | TESTING | MTS | 1 | Ravi | null
xyz | 10000 | 2019-04-20 18:30:00.000000+0000 | Finance | SDE | 121 | Jignesh | null
```

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> alter table Employee_Info add Projects set<text>;
cqlsh:employee> SELECT*FROM Employee_Info;

cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name | projects
-----+-----+-----+-----+-----+-----+-----+-----+
xyz | 50000 | 2021-03-19 18:30:00.000000+0000 | DEVELOPEMENT | MANAGER | 2 | Vamshi | null
xyz | 20000 | 2020-05-06 18:30:00.000000+0000 | QUALITY | ANALYST | 3 | Ramesh | null
xyz | 12000 | 2020-08-23 18:30:00.000000+0000 | TESTING | MTS | 1 | Ravi | null
xyz | 10000 | 2019-04-20 18:30:00.000000+0000 | Finance | SDE | 121 | Jignesh | null

(4 rows)
```

7. Update the altered table to add project names.

```
cqlsh:employee> update Employee_Info SET projects=projects+['ML'] where cluster_col='xyz' and salary=12000 IF emp_id=1;
[applied]
-----
True

cqlsh:employee> update Employee_Info SET projects=projects+['AI','DS'] where cluster_col='xyz' and salary=50000 IF emp_id=2;
[applied]
-----
True

cqlsh:employee> update Employee_Info SET projects=projects+['DEVOPS'] where cluster_col='xyz' and salary=20000 IF emp_id=3;
[applied]
-----
True

cqlsh:employee> update Employee_Info SET projects=projects+['QUANTUM COMPUTING'] where cluster_col='xyz' and salary=10000 IF emp_id=121;
[applied]
-----
True

cqlsh:employee> SELECT*FROM Employee_Info;
cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name | projects
-----+-----+-----+-----+-----+-----+-----+-----+
xyz | 50000 | 2021-03-19 18:30:00.000000+0000 | DEVELOPEMENT | MANAGER | 2 | Vamshi | {'AI', 'DS'}
xyz | 20000 | 2020-05-06 18:30:00.000000+0000 | QUALITY | ANALYST | 3 | Ramesh | {'DEVOPS'}
xyz | 12000 | 2020-08-23 18:30:00.000000+0000 | TESTING | MTS | 1 | Ravi | {'ML'}
xyz | 10000 | 2019-04-20 18:30:00.000000+0000 | Finance | SDE | 121 | Jignesh | {'QUANTUM COMPUTING'}
```

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

```
(4 rows)
cqlsh:employee> INSERT INTO Employee_Info(cluster_col,emp_id,emp_name,designation,Date_of_Joining,Salary,Dept_Name) VALUES ('xyz',121,'Modi','SDE','2022-04-20 18:30:00.000000+0000') using TTL 15;
cqlsh:employee> SELECT*FROM Employee_Info;
cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name | projects
-----+-----+-----+-----+-----+-----+-----+-----+
xyz | 50000 | 2021-03-19 18:30:00.000000+0000 | DEVELOPEMENT | MANAGER | 2 | Vamshi | {'AI', 'DS'}
xyz | 20000 | 2020-05-06 18:30:00.000000+0000 | QUALITY | ANALYST | 3 | Ramesh | {'DEVOPS'}
xyz | 12000 | 2020-08-23 18:30:00.000000+0000 | TESTING | MTS | 1 | Ravi | {'ML'}
xyz | 10000 | 2019-04-20 18:30:00.000000+0000 | Finance | SDE | 121 | Jignesh | {'QUANTUM COMPUTING'}
xyz | 1000 | 2022-04-20 18:30:00.000000+0000 | PRODUCTION | SDE | 121 | Modi | null

(5 rows)
cqlsh:employee> SELECT*FROM Employee_Info;
cluster_col | salary | date_of_joining | dept_name | designation | emp_id | emp_name | projects
-----+-----+-----+-----+-----+-----+-----+-----+
xyz | 50000 | 2021-03-19 18:30:00.000000+0000 | DEVELOPEMENT | MANAGER | 2 | Vamshi | {'AI', 'DS'}
xyz | 20000 | 2020-05-06 18:30:00.000000+0000 | QUALITY | ANALYST | 3 | Ramesh | {'DEVOPS'}
xyz | 12000 | 2020-08-23 18:30:00.000000+0000 | TESTING | MTS | 1 | Ravi | {'ML'}
xyz | 10000 | 2019-04-20 18:30:00.000000+0000 | Finance | SDE | 121 | Jignesh | {'QUANTUM COMPUTING'}

(4 rows)
```

3. 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

1. Create a key space by name Library

```
cqlsh> describe keyspaces;

system_schema system      system_distributed system_traces
system_auth   test_keyspace employee

cqlsh> CREATE KEYSPACE library WITH replication={'class':'SimpleStrategy','replication_factor':1};
cqlsh> describe keyspace library;

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

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> use library;
cqlsh:library> create table Library_Info (Stud_Id int,Counter_value counter,Stud_Name text,Book_Name text,Book_Id int,Date_of_issue timestamp,PRIMARY KEY(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_of_issue));
cqlsh:library> describe table Library_Info

CREATE TABLE library.library_info (
    stud_id int,
    stud_name text,
    book_name text,
    book_id int,
    date_of_issue timestamp,
    counter_value counter,
    PRIMARY KEY (stud_id, stud_name, book_name, book_id, date_of_issue)
) WITH CLUSTERING ORDER BY (stud_name ASC, book_name ASC, book_id ASC, date_of_issue ASC)
AND bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND crc_check_chance = 1.0
AND dclocal_read_repair_chance = 0.1
AND default_time_to_live = 0
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_chance = 0.0
AND speculative_retry = '99PERCENTILE';
```

3. Insert the values into the table in batch

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

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where stud_id=1 and stud_name='Vamshi' and book_name='OOND' and book_id=100 and date_of_issue='2022-04-17 18:38:00.000000+0000';
cqlsh:library> update Library_Info set counter_value=counter_value+1 where stud_id=2 and stud_name='Ravi' and book_name='CNS' and book_id=101 and date_of_issue='2022-03-14 18:38:00.000000+0000';
cqlsh:library> update Library_Info set counter_value=counter_value+1 where stud_id=112 and stud_name='Ramesh' and book_name='BDA' and book_id=102 and date_of_issue='2022-03-18 18:38:00.000000+0000';
cqlsh:library> select*from Library_Info;

stud_id | stud_name | book_name | book_id | date_of_issue | counter_value
-----+-----+-----+-----+-----+-----+
1 | Vamshi | OOND | 100 | 2022-04-17 18:38:00.000000+0000 | 1
2 | Ravi | CNS | 101 | 2022-03-14 18:38:00.000000+0000 | 1
112 | Ramesh | BDA | 102 | 2022-03-18 18:38:00.000000+0000 | 1

(3 rows)
```

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

```
cqlsh:library> update Library_Info set counter_value=counter_value+1 where stud_id=112 and stud_name='Ramesh' and book_name='BDA' and book_id=102 and date_of_issue='2022-03-18 18:38:00.000000+0000';
cqlsh:library> select*from Library_Info;

stud_id | stud_name | book_name | book_id | date_of_issue | counter_value
-----+-----+-----+-----+-----+-----+
1 | Vamshi | OOND | 100 | 2022-04-17 18:38:00.000000+0000 | 1
2 | Ravi | CNS | 101 | 2022-03-14 18:38:00.000000+0000 | 1
112 | Ramesh | BDA | 102 | 2022-03-18 18:38:00.000000+0000 | 2

(3 rows)
```

6. Export the created column to a csv file

```
cqlsh:library> copy Library_Info (stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) to 'C:\Users\shara\OneDrive\Documents\Library_Info.csv' with header=true

Using 7 child processes

Starting copy of library.library_info with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 3 rows; Rate: 2 rows/s; Avg. rate: 1 rows/s
3 rows exported to 1 files in 3.164 seconds.
```

| | A | B | C | D | E | F | G | H |
|---|---------|-----------|-----------|---------|---------------|---------------|---|---|
| 1 | stud_id | stud_name | book_name | book_id | date_of_issue | counter_value | | |
| 2 | 112 | Ramesh | BDA | 102 | 2022-03-18 | 2 | | |
| 3 | 1 | Vamshi | OOMD | 100 | 2022-04-17 | 1 | | |
| 4 | 2 | Ravi | CNS | 101 | 2022-03-14 | 1 | | |
| 5 | | | | | | | | |

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

```
cqlsh:library> create table Library_test(stud_id int,counter_value counter,stud_name text,book_name text,book_id int,date_of_issue timestamp,primary key(stud_id,stud_name,book_name,book_id,date_of_issue));
cqlsh:library> COPY Library_test(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) FROM 'C:\Users\shara\OneDrive\Documents\Library_Info.csv' with header=true;
;
Using 7 child processes
```

```
Starting copy of library.library_test with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Process ImportProcess-8: 1 rows/s; Avg. rate: 1 rows/s
```

```
AttributeError: 'NoneType' object has no attribute 'add_timer'
Processed: 3 rows; Rate: 1 rows/s; Avg. rate: 0 rows/s
3 rows imported from 1 files in 6.260 seconds (0 skipped).
```

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

| stud_id | stud_name | book_name | book_id | date_of_issue | counter_value |
|---------|-----------|-----------|---------|---------------------------------|---------------|
| 1 | Vamshi | OOMD | 100 | 2022-04-17 18:30:00.000000+0000 | 1 |
| 2 | Ravi | CNS | 101 | 2022-03-14 18:30:00.000000+0000 | 1 |
| 112 | Ramesh | BDA | 102 | 2022-03-18 18:30:00.000000+0000 | 2 |

```
(3 rows)
cqlsh:library>
```

4. Execution of HDFS Commands for interaction with Hadoop Environment.

1. Successful installation proof

```
hadoop@sharat-VirtualBox:~/Desktop$ hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
or   hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
  where CLASSNAME is a user-provided Java class

  OPTIONS is none or any of:
  buildpaths      attempt to add class files from build tree
  --config dir    Hadoop config directory
  --debug          turn on shell script debug mode
  --help           usage information
  hostnames list[,of,host,names] hosts to use in slave mode
  hosts filename   list of hosts to use in slave mode
  loglevel level   set the log4j level for this command
  workers          turn on worker mode

  SUBCOMMAND is one of:
  Admin Commands:
  daemonlog       get/set the log level for each daemon
  Client Commands:
  jps

hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ jps
5809 NodeManager
5170 DataNode
5650 ResourceManager
5015 NameNode
5415 SecondaryNameNode
6442 Jps
```

2. Mkdir

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -mkdir /lab5
2022-06-08 09:57:10,719 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

3. Ls

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hadoop fs -ls /
2022-06-08 09:57:33,445 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--  1 hdoop  supergroup      3745 2022-06-06 23:56 /Hadoop_Installation_Commands.txt
drwxr-xr-x  - hdoop  supergroup          0 2022-06-08 09:57 /lab5
```

4. put

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -put /home/hadoop/Desktop/a.txt
/labs
2022-06-08 10:02:57,394 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs fs -ls /
2022-06-08 10:03:08,676 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 hdoop supergroup      3745 2022-06-06 23:56 /Hadoop_Installation_Commands.txt
drwxr-xr-x  - hdoop supergroup          0 2022-06-08 10:02 /lab5
```

5. copyFromLocal

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -copyFromLocal /home/hadoop/Desktop/b.txt /home/hadoop/Desktop/c.txt /lab5
2022-06-08 10:07:13,375 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -cat /lab5
2022-06-08 10:07:36,122 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
cat: '/lab5': Is a directory
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -ls /lab5
2022-06-08 10:08:26,214 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 3 items
-rw-r--r-- 1 hdoop supergroup      15 2022-06-08 10:02 /lab5/a.txt
-rw-r--r-- 1 hdoop supergroup      0 2022-06-08 10:07 /lab5/b.txt
-rw-r--r-- 1 hdoop supergroup      0 2022-06-08 10:07 /lab5/c.txt
```

6. Get

i.get

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -get /lab5/a.txt /home/hadoop/Desktop/test.txt
2022-06-08 10:10:29,649 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -ls /home/hadoop/Desktop
2022-06-08 10:11:08,053 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
ls: '/home/hadoop/Desktop': No such file or directory
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ ls /home/hadoop/Desktop
a.txt b.txt c.txt test.txt
```

ii.getmerge

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -getmerge /lab5/b.txt /lab5/c.txt /home/hadoop/Desktop/merge.txt
2022-06-08 10:15:24,732 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ ls /home/hadoop/Desktop
a.txt b.txt c.txt merge.txt test.txt
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ cat /home/hadoop/Desktop/merge.txt
```

iii.getfacl

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -getfacl /lab5
2022-06-08 10:17:26,801 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
# file: /lab5
# owner: hdoop
# group: supergroup
user::rwx
group::r-x
other::r-x
```

7.copyToLocal

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -copyToLocal /lab5/a.txt /home/hadoop/Documents
2022-06-08 10:19:08,660 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ ls /home/hadoop/Documents
a.txt
```

8.cat

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hdfs dfs -cat /lab5/a.txt
2022-06-08 10:19:48,077 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
this is a test
```

9.mv

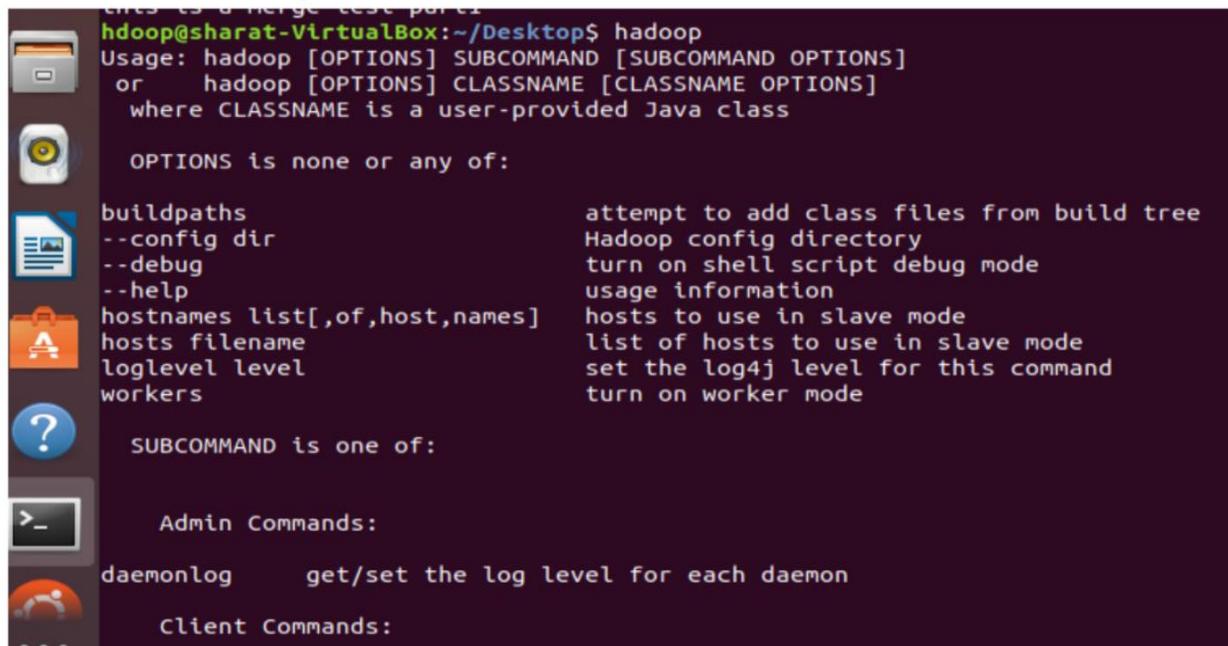
```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hadoop fs -mv /lab5/a.txt /lab5_part2
2022-06-08 10:22:18,644 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

10.cp

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hadoop fs -cp /lab5/b.txt /lab5_part2
2022-06-08 10:23:16,644 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~/hadoop-3.2.3$ hadoop fs -ls /lab5_part2
2022-06-08 10:23:21,944 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 hdoop supergroup      15 2022-06-08 10:02 /lab5_part2/a.txt
-rw-r--r-- 1 hdoop supergroup       0 2022-06-08 10:23 /lab5_part2/b.txt
```

5. Screenshot of Hadoop installed

1. Successful installation proof



```
hadoop@sharat-VirtualBox:~/Desktop$ hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
or   hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
  where CLASSNAME is a user-provided Java class

  OPTIONS is none or any of:

  buildpaths          attempt to add class files from build tree
  --config dir        Hadoop config directory
  --debug             turn on shell script debug mode
  --help              usage information
  hostnames list[,of,host,names] hosts to use in slave mode
  hosts filename      list of hosts to use in slave mode
  loglevel level     set the log4j level for this command
  workers             turn on worker mode

  SUBCOMMAND is one of:

    Admin Commands:
    daemonlog      get/set the log level for each daemon

    Client Commands:
```

- 6. 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**

a)

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;
```

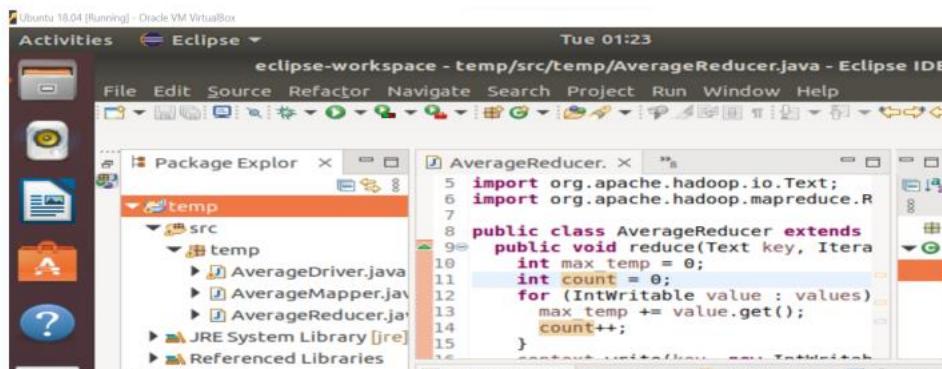
```

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:



```

Ubuntu 18.04 [Running] - Oracle VM VirtualBox
Activities  Eclipse
eclipse-workspace - temp/src/temp/AverageReducer.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Tue 01:23
Package Explorer  AverageReducer. X  *8
temp
src
temp
AverageDriver.java
AverageMapper.java
AverageReducer.java
JRE System Library [jre]
Referenced Libraries
WordCount
Problems X  Javadoc Declaration
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));
    }
}

put: /home/hadoop/Desktop/1901.txt: No such file or directory
hadoop@sharat-VirtualBox:~$ hdfs dfs -put /home/hadoop/Desktop/1901 /inputt
2022-06-28 01:12:47,278 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~$ hdfs dfs -ls /inputt
2022-06-28 01:13:05,646 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 4 items
-rw-r--r--  1 hdoop supergroup      888190 2022-06-28 01:12 /inputt/1901
-rw-r--r--  1 hdoop supergroup          15 2022-06-20 16:51 /inputt/a.txt
-rw-r--r--  1 hdoop supergroup         38 2022-06-27 22:01 /inputt/b.txt
drwxr-xr-x - hdoop supergroup             0 2022-06-20 16:52 /inputt/output

```

```
hadoop@sharat-VirtualBox:~$ hadoop jar weathertwo.jar temp.AverageDriver /inputt
/1901 /inputt/outputweather
2022-06-28 01:21:32,366 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
2022-06-28 01:21:33,696 INFO client.RMProxy: Connecting to ResourceManager at /
127.0.0.1:8032
2022-06-28 01:21:34,100 WARN mapreduce.JobResourceUploader: Hadoop command-line
option parsing not performed. Implement the Tool interface and execute your ap
plication with ToolRunner to remedy this.
2022-06-28 01:21:34,131 INFO mapreduce.JobResourceUploader: Disabling Erasure C
oding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1656358828291_0001
2022-06-28 01:21:35,309 INFO input.FileInputFormat: Total input files to proces
s : 1
2022-06-28 01:21:35,410 INFO mapreduce.JobSubmitter: number of splits:1
2022-06-28 01:21:35,589 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1656358828291_0001
2022-06-28 01:21:35,590 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-06-28 01:21:36,346 INFO conf.Configuration: resource-types.xml not found
2022-06-28 01:21:36,346 INFO resource.ResourceUtils: Unable to find 'resource-t
ypes.xml'.
2022-06-28 01:21:37,378 INFO impl.YarnClientImpl: Submitted application applica
tion_1656358828291_0001
2022-06-28 01:21:38,336 INFO mapreduce.Job: The url to track the job: http://sh
arat-VirtualBox:8088/proxy/application_1656358828291_0001/
2022-06-28 01:21:38,338 INFO mapreduce.Job: Running job: job_1656358828291_0001
2022-06-28 01:21:48,759 INFO mapreduce.Job: Job job_1656358828291_0001 running
in uber mode : false
2022-06-28 01:21:48,760 INFO mapreduce.Job: map 0% reduce 0%
```

```
Reduce input groups=1
Reduce shuffle bytes=72210
Reduce input records=6564
Reduce output records=1
Spilled Records=13128
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=754
CPU time spent (ms)=1840
Physical memory (bytes) snapshot=645009408
Virtual memory (bytes) snapshot=5166370816
Total committed heap usage (bytes)=658505728
Peak Map Physical memory (bytes)=450666496
Peak Map Virtual memory (bytes)=2579943424
Peak Reduce Physical memory (bytes)=194342912
```

```
Bytes Written=8
hadoop@sharat-VirtualBox:~$ hdfs dfs -ls /inputt/outputweather
2022-06-28 01:22:16,506 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 hdoop supergroup 0 2022-06-28 01:21 /inputt/outputweath
er/_SUCCESS
-rw-r--r-- 1 hdoop supergroup 8 2022-06-28 01:21 /inputt/outputweath
er/part-r-00000
```

```
hadoop@sharat-VirtualBox:~$ hdfs dfs -cat /inputt/outputweather/part-r-00000
2022-06-28 01:23:07,585 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
1901 46
```

b)

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;

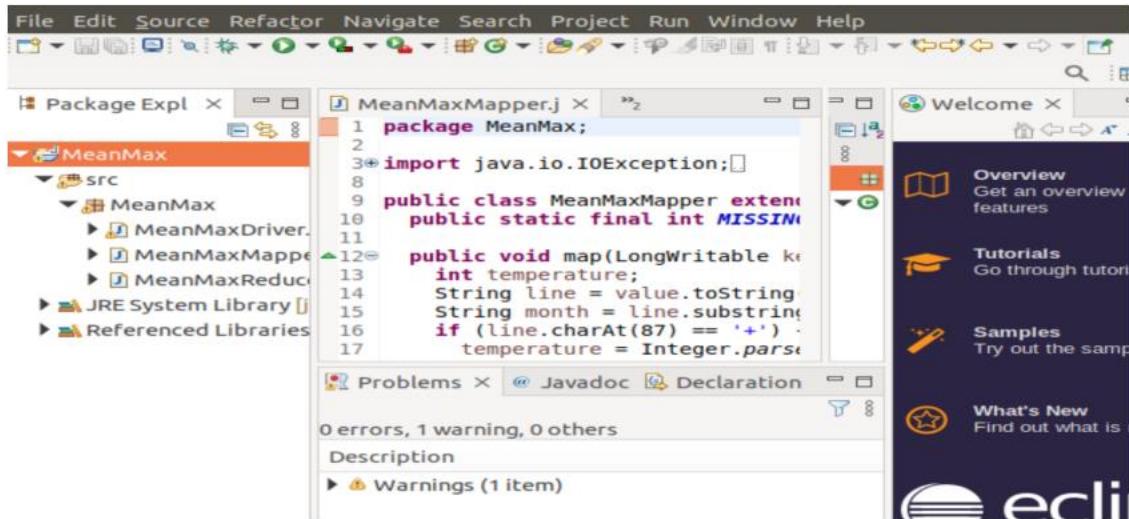
```

```

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@sharat-VirtualBox:~$ hadoop jar MeanMaxweather2.jar MeanMax.MeanMaxDriver  
/inputt/1901 /inputt/outputmeanmax  
2022-06-28 02:35:15,863 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
2022-06-28 02:35:16,403 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032  
2022-06-28 02:35:16,741 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.  
2022-06-28 02:35:16,774 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1656363425892_0001  
2022-06-28 02:35:17,464 INFO input.FileInputFormat: Total input files to process : 1  
2022-06-28 02:35:17,959 INFO mapreduce.JobSubmitter: number of splits:1  
2022-06-28 02:35:18,176 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1656363425892_0001  
2022-06-28 02:35:18,177 INFO mapreduce.JobSubmitter: Executing with tokens: []  
2022-06-28 02:35:18,417 INFO conf.Configuration: resource-types.xml not found  
2022-06-28 02:35:18,418 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.  
2022-06-28 02:35:18,932 INFO impl.YarnClientImpl: Submitted application application_1656363425892_0001
```

```
hadoop@sharat-VirtualBox:~$ hdfs dfs -ls /inputt/outputmeanmax  
2022-06-28 02:36:40,638 WARN util.NativeCodeLoader: Unable to load library for your platform... using builtin-java classes where applicable  
Found 2 items  
-rw-r--r-- 1 hdoop supergroup 0 2022-06-28 02:35 /inputt/outputmeanmax/_SUCCESS  
-rw-r--r-- 1 hdoop supergroup 74 2022-06-28 02:35 /inputt/outputmeanmax/part-r-00000  
hadoop@sharat-VirtualBox:~$ hdfs dfs -cat /inputt/outputmeanmax/part-r-00000  
2022-06-28 02:36:57,109 WARN util.NativeCodeLoader: Unable to load library for your platform... using builtin-java classes where applicable  
01 4  
02 0  
03 7  
04 44  
05 100  
06 168  
07 219  
08 198  
09 141  
10 100  
11 19  
12 3
```

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.

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> {

```

```

    protected void reduce(Text key, Iterable<IntWritable> values,
IntWritable outputValue) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        outputValue.set(sum);
    }
}
```

```

    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;

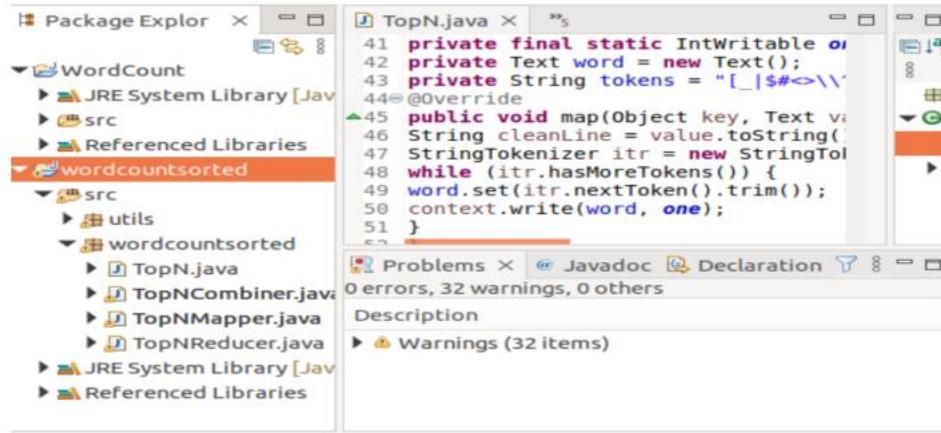
public class TopNReducer extends Reducer<Text, IntWritable,
Text, IntWritable> {
    private Map<Text, IntWritable> countMap = new
HashMap<>();

    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:



```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3/sbin$ hdfs dfs -mkdir /input
2022-06-27 21:59:42,586 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
mkdir: '/input': File exists
hadoop@sharat-VirtualBox:~/hadoop-3.2.3/sbin$ hdfs dfs -put /home/hadoop/Documents/b.txt /input
2022-06-27 22:00:59,014 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
put: '/input/b.txt': File exists
hadoop@sharat-VirtualBox:~/hadoop-3.2.3/sbin$ hdfs dfs -put /home/hadoop/Documents/b.txt /input
2022-06-27 22:01:16,095 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hadoop@sharat-VirtualBox:~/hadoop-3.2.3/sbin$ hdfs dfs -ls /input
2022-06-27 22:01:33,726 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 3 items
-rw-r--r-- 1 hadoop supergroup          15 2022-06-20 16:51 /inputt/a.txt
-rw-r--r-- 1 hadoop supergroup          38 2022-06-27 22:01 /inputt/b.txt
drwxr-xr-x  - hadoop supergroup          0 2022-06-20 16:52 /inputt/output
```

```
hadoop@sharat-VirtualBox:~/hadoop-3.2.3/sbin$ hdfs dfs -ls inputt/outputword
2022-06-27 22:08:26,995 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 hadoop supergroup          0 2022-06-27 22:05 inputt/outputword/_SUCCESS
-rw-r--r-- 1 hadoop supergroup        35 2022-06-27 22:05 inputt/outputword/part-r-00000
hadoop@sharat-VirtualBox:~/hadoop-3.2.3/sbin$ hdfs dfs -cat inputt/outputword/part-r-00000
2022-06-27 22:09:12,199 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
test    2
is      2
this   2
a      1
important  1
```

8. Create a Map Reduce program to demonstrating join operation

CODE:

```
// JoinDriver.java
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,  
Posts.class);  
  
MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class,  
User.class);  
  
FileOutputFormat.setOutputPath(conf, outputPath);  
  
conf.setPartitionerClass(KeyPartitioner.class);  
  
conf.setOutputValueGroupingComparator(TextPair.FirstComparator.cl  
ass);  
  
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);
```

```
}

// JoinReducer.java
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);
}
}
}

// User.java
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 User 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]));
}
}

//Posts.java
import java.io.IOException;

import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
```

```
public class Posts 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[3], "0"), new
Text(SingleNodeData[9]));
}
}

// TextPair.java
import java.io.*;
import org.apache.hadoop.io.*;

public class TextPair implements WritableComparable<TextPair> {

private Text first;
private Text second;

public TextPair() {
set(new Text(), new Text());
}

public TextPair(String first, String second) {
set(new Text(first), new Text(second));
}

public TextPair(Text first, Text second) {
```

```
    set(first, second);
}

public void set(Text first, Text second) {
    this.first = first;
    this.second = second;
}

public Text getFirst() {
    return first;
}

public Text getSecond() {
    return second;
}

@Override
public void write(DataOutput out) throws IOException {
    first.write(out);
    second.write(out);
}

@Override
public void readFields(DataInput in) throws IOException {
    first.readFields(in);
    second.readFields(in);
}

@Override
public int hashCode() {
    return first.hashCode() * 163 + second.hashCode();
}

@Override
public boolean equals(Object o) {
    if (o instanceof TextPair) {
```

```
TextPair tp = (TextPair) o;
return first.equals(tp.first) && second.equals(tp.second);
}
return false;
}

@Override
public String toString() {
return first + "\t" + second;
}

@Override
public int compareTo(TextPair tp) {
int cmp = first.compareTo(tp.first);
if (cmp != 0) {
return cmp;
}
return second.compareTo(tp.second);
}
// ^^ TextPair

// vv TextPairComparator
public static class Comparator extends WritableComparator {

private static final Text.Comparator TEXT_COMPARATOR = new
Text.Comparator();

public Comparator() {
super(TextPair.class);
}

@Override
public int compare(byte[] b1, int s1, int l1,
byte[] b2, int s2, int l2) {

try {
```

```
int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);
int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);
int cmp = TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2,
firstL2);
if (cmp != 0) {
return cmp;
}
return TEXT_COMPARATOR.compare(b1, s1 + firstL1, l1 - firstL1,
b2, s2 + firstL2, l2 - firstL2);
} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}
}

static {
WritableComparator.define(TextPair.class, new Comparator());
}
public static class FirstComparator extends WritableComparator {

private static final Text.Comparator TEXT_COMPARATOR = new
Text.Comparator();

public FirstComparator() {
super(TextPair.class);
}

@Override
public int compare(byte[] b1, int s1, int l1,
byte[] b2, int s2, int l2) {

try {
int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1);
int firstL2 = WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2);
return TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);
}
```

```

} catch (IOException e) {
throw new IllegalArgumentException(e);
}
}

@Override
public int compare(WritableComparable a, WritableComparable b) {
if (a instanceof TextPair && b instanceof TextPair) {
return ((TextPair) a).first.compareTo((TextPair) b).first);
}
return super.compare(a, b);
}
}
}

```

OUTPUT:

```

hadoop@sharat-VirtualBox:~$ hdfs dfs -copyFromLocal DeptName.txt DeptStrength.txt /
2022-06-28 01:49:34,172 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
copyFromLocal: 'DeptStrength.txt': No such file or directory
hadoop@sharat-VirtualBox:~$ hdfs dfs -copyFromLocal DeptName.txt DeptEmpStrength.txt /
2022-06-28 01:50:03,670 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
copyFromLocal: '/DeptName.txt': File exists
hadoop@sharat-VirtualBox:~$ hdfs dfs -copyFromLocal DeptEmpStrength.txt /
2022-06-28 01:50:14,698 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
copyFromLocal: '/DeptEmpStrength.txt': File exists

```

```

hadoop@sharat-VirtualBox:~$ hadoop jar MapReduceJoin.jar /DeptEmpStrength.txt /DeptName.txt /output_mapreducejoin
2022-06-28 01:54:22,260 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2022-06-28 01:54:22,634 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2022-06-28 01:54:22,756 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2022-06-28 01:54:22,936 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1656358828291_0002
2022-06-28 01:54:23,108 INFO mapred.FileInputFormat: Total input files to process : 1
2022-06-28 01:54:23,121 INFO mapred.FileInputFormat: Total input files to process : 1
2022-06-28 01:54:23,607 INFO mapreduce.JobSubmitter: number of splits:4
2022-06-28 01:54:23,771 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1656358828291_0002
2022-06-28 01:54:23,772 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-06-28 01:54:23,909 INFO conf.Configuration: resource-types.xml not found
2022-06-28 01:54:23,909 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-06-28 01:54:23,967 INFO impl.YarnClientImpl: Submitted application application_1656358828291_0002

```

```
Bytes Written=85
hadoop@sharat-VirtualBox:~$ hdfs dfs -ls /outputoutput_mapreducejoin
2022-06-28 01:55:29,436 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
ls: `/outputoutput_mapreducejoin': No such file or directory
hadoop@sharat-VirtualBox:~$ hdfs dfs -ls /output_mapreducejoin
2022-06-28 01:55:36,422 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 hadoop supergroup 0 2022-06-28 01:54 /output_mapreducejoin/_SUCCESS
-rw-r--r-- 1 hadoop supergroup 85 2022-06-28 01:54 /output_mapreducejoin/part-00000
hadoop@sharat-VirtualBox:~$ hdfs dfs -cat /output_mapreducejoin/part-00000
2022-06-28 01:56:01,106 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
A11      50          Finance
B12     100          HR
C13     250        Manufacturing
Dept_ID Total_Employee      Dept_Name
```

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

CODE:

```
package wordcount

import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.rdd.RDD.rddToPairRDDFunctions

object WordCount {
def
main(args: Array[String]) = {
//Start the Spark context
val conf = new SparkConf().setAppName("WordCount").setMaster("local")
val sc = new SparkContext(conf)
//Read some example file to a test RDD
val test =sc.textFile("input.txt")
test.flatMap {
line => //for
each line
line.split(" ") //split
the line in word by word.
} .map {
word => //for
each word
(word, 1) //Return a key/value tuple, with the word as key and 1 as value
.reduceByKey(_ + _) //Sum
all of the value with same key
.saveAsTextFile("output.txt") //Save
to a text file
//Stop the Spark context
sc.stop
}
}
```

OUTPUT:

```
scala> val test=sc.textFile("/home/hadoop/spark_word_count.txt")
test: org.apache.spark.rdd.RDD[String] = /home/hadoop/spark_word_count.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> test.collect;
[Stage 0:>                                         (0 + 0) /
[Stage 0:>                                         (0 + 2) /
res4: Array[String] = Array(This is a test, This is an evaluation, do you want a test, why do you want a test)

scala> val count=test.flatMap(line=>line.split(" "))
count: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at <console>:23

scala> count.collect
res5: Array[String] = Array(This, is, a, test, This, is, an, evaluation, do, u, want, a, test, why, do, you, want, a, test)

scala> val map_frequency=count.map(entry=>(entry,1))
map_frequency: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] map at <console>:23

scala> map_frequency.collect
res6: Array[(String, Int)] = Array((This,1), (is,1), (a,1), (test,1), (This,1), (is,1), (an,1), (evaluation,1), (do,1), (you,1), (want,1), (a,1), (test,1), (hy,1), (do,1), (you,1), (want,1), (a,1), (test,1))
```

```
scala> map_frequency.reduceByKey(_+_)
res7: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:24

scala> map_frequency.collect
res8: Array[(String, Int)] = Array((This,1), (is,1), (a,1), (test,1), (This,1), (is,1), (an,1), (evaluation,1), (do,1), (you,1), (want,1), (a,1), (test,1), (hy,1), (do,1), (you,1), (want,1), (a,1), (test,1))

scala>

scala> val final_output=map_frequency.reduceByKey(_+_)
final_output: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[5] at reduceByKey at <console>:23

scala> final_output.collect
[Stage 4:>                                         (0 + 2) /
res9: Array[(String, Int)] = Array((is,2), (evaluation,1), (This,2), (why,1), (want,2), (test,3), (you,2), (a,3), (do,2), (an,1))
```

10. 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

CODE:

```
val textFile = sc.textFile("/home/Desktop/test.txt")
val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
import scala.collection.immutable.ListMap
val sorted=ListMap(counts.collect.sortWith(_.value > _.value):_*)// sort in descending order based on values
println(sorted)
for((k,v)<-sorted)
{
  if(v>4)
  {
    print(k+",")
    print(v)
    println()
  }
}
```

OUTPUT:

```
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(test -> 5, is -> 2, This -> 2, want -> 1, why -> 1, you -> 1)
scala> for((k,v)<-sorted)
| {
|   if(v>4)
|   {
|     print(k+",")
|     print(v)
|     println()
|   }
| }
```

```
test,5
scala> val word_count=sc.textFile("/home/hadoop/spark_word_count.txt")
word_count: org.apache.spark.rdd.RDD[String] = /home/hadoop/spark_word_count.txt MapPartitionsRDD[1] at textFile at <console>:23
scala> val frequency=word_count.flatMap((line)=>line.split(" ")).map(word=>(word,1)).reduceByKey(_+_)
frequency: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23
scala> val sorted=ListMap(frequency.collect.sortWith(_.value > _.value):_*)
<console>:23: error: not found: value ListMap
      val sorted=ListMap(frequency.collect.sortWith(_.value > _.value):_*)
                           ^
scala> import scala.collection.immutable.ListMap
import scala.collection.immutable.ListMap
scala> val sorted=ListMap(frequency.collect.sortWith(_.value > _.value):_*)
[Stage 0:>]
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap(test -> 5, is -> 2, This -> 2, want -> 1, why -> 1, you -> 1)
```

```
sorted: scala.collection.immutable.ListMap[String,Int] = ListMap()
> 3, is -> 2, This -> 2, want -> 2, do -> 2, why -> 1, you -> 1
scala> for((k,v)<-sorted)
| {
| if(v>4)
| {
|   print(k+",")
|   print(v)
|   println()
| }
| }
test,5
```