

BIG DATA ANALYTICS LAB RECORD



Submitted by

**RAVI KUMAR
(1BM17CS150)**

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

Under the Guidance of
Dr. Latha N R
Assistant Professor
Department of CSE
BMSCE



B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
2020-2021

INDEX

SL NO.	DATE	PROGRAM	PAGE NO.
1.	24-09-2020	MongoDB : Student Database	3-7
2.	05-10-2020	MongoDB : Customer Database	8-12
3.	12-10-2020	Cassandra : Employee Keyspace	13 - 14
4.	02-11-2020	Cassandra : Library Keyspace	15 - 16
5.	09-11-2020	Hadoop : Word Count	17 - 20
6.	07-12-2020	Hadoop : Average Temperature	21 - 24
7.	14-12-2020	Hive : Employee Table	25 - 29

LAB 1

MongoDB : Student Database

Perform the following DB operations using MongoDB

- 1. Create a database “Student” with the following attributes Rollno, Age, ContactNo,Email Id.**
- 2. Insert appropriate values**
- 3. Write query to update Email-Id of a student with rollno 10.**
- 4. Replace the student name from “ABC” to “FEM” of rollno 11.**
- 5. Export the created table into local file system**
- 6. Drop the table**
- 7. Import a given csv dataset from local file system into mongodb collection.**

use StudentDB

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

```
db.createCollection("Students")
```

2. Insert appropriate values

```
db.Students.inser({RollNo:10,Age:23, Mob:9999988888,Name:"Abc",  
Email:"abc@gmail.com"})
```

```
db.Students.inser({RollNo:11,Age:20, Mob:9999988887,Name:"Efg",  
Email:"efg@gmail.com"})
```

```
db.Students.inser({RollNo:8,Age:22, Mob:9999988886,Name:"Lmn",  
Email:"lmn@gmail.com"})
```

```
db.Students.find()
```

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

```
db.Students.update({RollNo:10},{$set:{Email:"ABC@gmail.com"}});
```

```
db.Students.find({RollNo:10})
```

4. Replace the student name from “ABC” to “FEM” of rollno 11

```
db.Students.update({RollNo:11},{$set:{Name:"FEM"}});
```

```
db.Students.find({RollNo:11})
```

5. Export the created table into local file system

```
mongoexport -d College –c Students –f RollNo,Age,Mob,Name, Email --type=csv –o Student.csv
```

6. Drop the table

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

7. Import a given csv dataset from local file system

```
mongoimport -d College -c Students --type csv --file Student.csv --headerline
```

Output:

The screenshot shows the Robo 3T MongoDB interface. On the left, the sidebar displays the database structure under 'BDA_LAB (5)'. The main area shows a mongo shell session with the following commands and output:

```
use College
db.createCollection('Students')

db.Students.insert({ RollNo:10, Age:23, Mob:9999988888, Name:"Abc", Email:"abc@gmail.com"})
db.Students.insert({ RollNo:11, Age:20, Mob:9999988887, Name:"Efg", Email:"efg@gmail.com"})
db.Students.insert({ RollNo:8, Age:22, Mob:9999988886, Name:"Lmn", Email:"lmn@gmail.com"})

db.Students.find()

db.Students.update({RollNo:10}, {$set:{Email:"ABC@gmail.com"}})

db.Students.update({Name:"Abc"}, {$set:{Name:"FEM"}})

mongoexport -d Database -c Student -f RollNo,Age,Mob,Name,Email --type=csv -o Student.csv

db.Students.drop()

mongoimport -d Database -c Student --type csv --file
```

Below the mongo shell, the 'Students' collection is listed with its data. The data consists of three documents, each with fields: _id, RollNo, Age, Mob, Name, and Email.

Key	Value	Type
(1) ObjectId("5f841f86a0edc3593cb3cc0a")	{ 6 fields }	Object
_id	ObjectId("5f841f86a0edc3593cb3cc0a")	ObjectId
RollNo	10.0	Double
Age	23.0	Double
Mob	9999988888.0	Double
Name	Abc	String
Email	abc@gmail.com	String
(2) ObjectId("5f841f8ba0edc3593cb3cc0b")	{ 6 fields }	Object
(3) ObjectId("5f841f8ea0edc3593cb3cc0c")	{ 6 fields }	Object

This screenshot shows the same Robo 3T interface after modifications have been applied. The mongo shell session now reflects the changes made in the previous screenshot, specifically the updates to the 'Email' field and the 'Name' field.

```
use College
db.createCollection('Students')

db.Students.insert({ RollNo:10, Age:23, Mob:9999988888, Name:"Abc", Email:"abc@gmail.com"})
db.Students.insert({ RollNo:11, Age:20, Mob:9999988887, Name:"Efg", Email:"efg@gmail.com"})
db.Students.insert({ RollNo:8, Age:22, Mob:9999988886, Name:"Lmn", Email:"lmn@gmail.com"})

db.Students.find()

db.Students.update({RollNo:10}, {$set:{Email:"ABC@gmail.com"}})
db.Students.find()

db.Students.update({Name:"Abc"}, {$set:{Name:"FEM"}})

mongoexport -d Database -c Student -f RollNo,Age,Mob,Name,Email --type=csv -o Student.csv

db.Students.drop()
```

The 'Students' collection table shows the updated data. The first document's 'Email' field has been changed to 'ABC@gmail.com' and its 'Name' field has been changed to 'FEM'. The second document's 'Email' field has also been updated to 'ABC@gmail.com'.

Key	Value	Type
(1) ObjectId("5f841f86a0edc3593cb3cc0a")	{ 6 fields }	Object
_id	ObjectId("5f841f86a0edc3593cb3cc0a")	ObjectId
RollNo	10.0	Double
Age	23.0	Double
Mob	9999988888.0	Double
Name	Abc	String
Email	ABC@gmail.com	String
(2) ObjectId("5f841f8ba0edc3593cb3cc0b")	{ 6 fields }	Object
(3) ObjectId("5f841f8ea0edc3593cb3cc0c")	{ 6 fields }	Object

BDA_LAB (5)

Welcome * db.Students.find()

```

use College
db.createCollection('Students')

db.Students.insert({ RollNo:10, Age:23, Mob:9999988888, Name:"Abc", Email:"abc@gmail.com"})
db.Students.insert({ RollNo:11, Age:20, Mob:9999988887, Name:"Efg", Email:"efg@gmail.com"})
db.Students.insert({ RollNo:8, Age:22, Mob:9999988886, Name:"Lmn", Email:"lmn@gmail.com"})

db.Students.find()

db.Students.update({RollNo:10}, {$set:{Email:"ABC@gmail.com"}})
db.Students.find()

db.Students.update({Name:"Abc"}, {$set:{Name:"FEM"}})
db.Students.find()

mongoexport -d Database -c Student -f RollNo,Age,Mob,Name,Email --type=csv -o Student.csv

db.Students.drop()

```

Students 0.002 sec.

Key	Value	Type
(1) ObjectId("5fb41f86a0edc3593cb3cc0a")	{ 6 fields }	Object
_id	ObjectId("5fb41f86a0edc3593cb3cc0a")	Objectid
RollNo	10.0	Double
Age	23.0	Double
Mob	9999988880	Double
Name	FEM	String
Email	ABC@gmail.com	String
(2) ObjectId("5fb41f8ba0edc3593cb3cc0b")	{ 6 fields }	Object
(3) ObjectId("5fb41f8ea0edc3593cb3cc0c")	{ 6 fields }	Object

File View Options Window Help

Welcome * mongorestore -d College <St... *

```

use College
db.Students.insert({ RollNo:10, Age:23, Mob:9999988888, Name:"Abc", Email:"abc@gmail.com"})
db.Students.insert({ RollNo:11, Age:20, Mob:9999988887, Name:"Efg", Email:"efg@gmail.com"})
db.Students.insert({ RollNo:8, Age:22, Mob:9999988886, Name:"Lmn", Email:"lmn@gmail.com"})

db.Students.find()

db.Students.update({RollNo:10}, {$set:{Email:"ABC@gmail.com"}})
db.Students.find()

db.Students.update({Name:"Abc"}, {$set:{Name:"FEM"}})
db.Students.find()

```

Administrator: Command Prompt

(c) 2017 Microsoft Corporation. All rights reserved.

```

C:\Windows\system32>cd C:\Program Files\MongoDB\Server\4.0\bin

C:\Program Files\MongoDB\Server\4.0\bin>mongoexport -d College -c Students -f RollNo,Age,Mob,Name,Email --type=csv -o Student.csv
2020-10-12T15:10:03.951+0530    connected to: localhost
2020-10-12T15:10:04.024+0530    exported 3 records

```

Student [Read-Only] - N

	A	B	C	D	E	F
1	RollNo	Age	Mob	Name	Email	
2	10	23	1.00E+10	FEM	ABC@gmail.com	
3	11	20	1.00E+10	Efg	efg@gmail.com	
4	8	22	1.00E+10	Lmn	lmn@gmail.com	
5						
6						

```

BDA_LAB (5)                                     *db.Students.find()
> System
> College
> Company
> config
BDA_LAB   localhost:27017  College
db.Students.insert({ RollNo:10, Age:23, Mob:9999988888, Name:"Abc", Email:"abc@gmail.com"})
db.Students.insert({ RollNo:11, Age:20, Mob:9999988887, Name:"Efg", Email:"efg@gmail.com"})
db.Students.insert({ RollNo:12, Age:22, Mob:9999988886, Name:"Lmn", Email:"lmn@gmail.com"})

db.Students.find()

db.Students.update({RollNo:10}, {$set:{Email:"ABC@gmail.com"}})
db.Students.find()

db.Students.update({Name:"Abc"}, {$set:{Name:"FEM"}})
db.Students.find()

mongoexport -d College -c Students -f RollNo,Age,Mob,Name,Email --type=csv -o Students.csv

db.Students.drop()
db.Students.find()

mongoimport -d College -c Students --type csv --file
0.003 sec.
Fetched 0 record(s) in lms

```

The screenshot shows the Robo 3T interface with two panes. The left pane displays the MongoDB shell and command-line output, while the right pane shows a Microsoft Excel spreadsheet titled 'Student [Read-Only]'.

MongoDB Shell Output:

```

e View Options Window Help
BDA_LAB (5)                                     *db.Students.find()
> System
> College
> Company
> config
Administrator: Command Prompt
2020-10-12T15:10:03.951+0530 connected to: localhost
2020-10-12T15:10:04.024+0530 exported 3 records
C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d College -c Students --type csv --file
2020-10-12T15:18:25.053Z error parsing command line options: expected argument for flag `/file'
2020-10-12T15:18:25.176+0530 try 'mongoimport --help' for more information
C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d College -c Students --type csv --file Student.csv --headerline
2020-10-12T15:19:49.046+0530 connected to: localhost
2020-10-12T15:19:49.376+0530 imported 3 documents
C:\Program Files\MongoDB\Server\4.0\bin>
mongorestore -d College -c Students --type csv --file Student.csv --headerline
db.Students.find()

Students  0.003 sec.



| A      | B   | C   | D        | E     | F             |
|--------|-----|-----|----------|-------|---------------|
| RollNo | Age | Mob | Name     | Email |               |
| 1      | 10  | 23  | 1.00E+10 | FEM   | ABC@gmail.com |
| 2      | 11  | 20  | 1.00E+10 | Efg   | efg@gmail.com |
| 3      | 8   | 22  | 1.00E+10 | Lmn   | lmn@gmail.com |


```

Excel Spreadsheet:

The Excel sheet has columns A through F. Column A is labeled 'RollNo', column B is 'Age', column C is 'Mob', column D is 'Name', column E is 'Email', and column F is empty. The data is as follows:

A	B	C	D	E	F
RollNo	Age	Mob	Name	Email	
1	10	23	1.00E+10	FEM	ABC@gmail.com
2	11	20	1.00E+10	Efg	efg@gmail.com
3	8	22	1.00E+10	Lmn	lmn@gmail.com

LAB 2

MongoDB : Customer Database

Perform the following DB operations using MongoDB.

1. Create a collection by name Customers with the following attributes. Cust_id, Acc_Bal, Acc_Type
2. Insert at least 5 values into the table
3. Write a query to display those records whose total account balance is greater than 1200 of account type 'Z' for each customer_id.
4. Determine Minimum and Maximum account balance for each customer_id.
5. Export the created collection into local file system
6. Drop the table
7. Import a given csv dataset from local file system into mongodb collection.

use Database

1. Create a collection by name Customers with the following attributes. Cust_id, Acc_Bal, Acc_Type

```
db.createCollection("Customer")
```

2. Insert at least 5 values into the table

```
db.Customer.insert({cust_id:1,AccBal:1600,AccType:"Z"})
db.Customer.insert({cust_id:2, AccBal:4000, AccType:"A"})
db.Customer.insert({cust_id:1, AccBal:5000, AccType:"B"})
db.Customer.insert({cust_id:4, AccBal:3500, AccType:"Z"})
db.Customer.insert({cust_id:2, AccBal:6000, AccType:"A"})
```

```
db.Customer.find()
```

3. Write a query to display those records whose total account balance is greater than 1200 of account type 'Z' for each customer_id.

```
db.Customer.find({AccBal:{$gt:1200}, AccType:"Z"})
```

4. Determine Minimum and Maximum account balance for each customer_id.

```
db.Customer.aggregate([{"$group": {"_id": "$cust_id", "min_bal": {"$min": "$AccBal"}, "max_bal": {"$max": "$AccBal"}}}])
```

5. Export the created collection into local file system

```
mongoexport -d Database -c Customer -f cust_id,AccBal,AccType --type=csv -o Customer.csv
```

6. Drop the table

```
db.Customer.drop()
```

7. Import a given csv dataset from local file system into mongodb collection

```
mongoimport -d Database -c Customer --type csv --file Customer.csv --headerline
```

Output:

The screenshot shows the Robo 3T MongoDB interface. On the left, the sidebar displays databases: BDA_LAB (5), System, College, Company, and config. The main window has two tabs: 'Welcome' and '* db.Customer.find()'. The 'db.Customer.insert()' command is run in the terminal, followed by 'db.Customer.find()'. Below the terminal, the 'Customer' collection is shown in a table format. The table has columns: Key, Value, and Type. The data consists of five documents, each with an '_id' field (ObjectId) and three data fields: 'cust_id' (Double), 'AccBal' (Double), and 'AccType' (String). The 'AccBal' values are 1.0, 1600.0, 5000.0, 3500.0, and 6000.0 respectively, and the 'AccType' values are 'Z', 'A', 'B', 'Z', and 'A' respectively.

Key	Type
{(1) ObjectId("5f842a41a0edc3593cb3cc0d")}	Object
_id	ObjectId("5f842a41a0edc3593cb3cc0d")
cust_id	1.0
AccBal	1600.0
AccType	Z
{(2) ObjectId("5f842a45a0edc3593cb3cc0e")}	Object
{(3) ObjectId("5f842a48a0edc3593cb3cc0f")}	Object
{(4) ObjectId("5f842a4ba0edc3593cb3cc10")}	Object
{(5) ObjectId("5f842a4ea0edc3593cb3cc11")}	Object

Logs

```

BDA_LAB (5)
> System
> College
> Company
> config

Welcome * db.Customer.find({AccBal:@gt;1200, AccType:"Z"})
BDA_LAB localhost:27017 Database

use Database
db.createCollection('Customer')

db.Customer.insert({cust_id:1,AccBal:1600,AccType:"Z"})
db.Customer.insert({cust_id:2,AccBal:4000,AccType:"A"})
db.Customer.insert({cust_id:1,AccBal:5000,AccType:"B"})
db.Customer.insert({cust_id:4,AccBal:3500,AccType:"Z"})
db.Customer.insert({cust_id:2,AccBal:6000,AccType:"A"})
db.Customer.find()

db.Customer.find({AccBal:@gt;1200, AccType:"Z"})


Customer 0.092 sec.

Key Value Type
(1) ObjectId("5f842a41a0edc3593cb3cc0d")
  _id ObjectId
  cust_id 1.0 Double
  AccBal 1600.0 Double
  AccType Z String
(2) ObjectId("5f842a4ba0edc3593cb3cc10")
  _id ObjectId
  cust_id 4.0 Double
  AccBal 3500.0 Double
  AccType Z String

```

Logs

```

BDA_LAB (5)
> System
> College
> Company
> config

Welcome * db.Customer.aggregate([
  ...
])
BDA_LAB localhost:27017 Database

db.Customer.insert({cust_id:2,AccBal:4000,AccType:"A"})
db.Customer.insert({cust_id:1,AccBal:5000,AccType:"B"})
db.Customer.insert({cust_id:4,AccBal:3500,AccType:"Z"})
db.Customer.insert({cust_id:2,AccBal:6000,AccType:"A"})
db.Customer.find()

db.Customer.find({AccBal:@gt;1200, AccType:"Z"})

db.Customer.aggregate([
  {
    $group: {
      id: "$cust_id",
      min_bal: {$min: "$AccBal"},
      max_bal: {$max: "$AccBal"}
    }
  }
])

Customer 0.594 sec.

Key Value Type
(1) 2.0
  _id 2.0 Double
  min_bal 4000.0 Double
  max_bal 6000.0 Double
(2) 4.0
  _id 4.0 Double
  min_bal 3500.0 Double
  max_bal 3500.0 Double
(3) 1.0
  _id 1.0 Object
  id Double

```

Administrator: Command Prompt

```

2020-10-12T15:18:25.175+0530 error parsing command line options: expected argument for flag '/file'
2020-10-12T15:18:25.176+0530 try 'mongoimport --help' for more information

C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d College -c Students --type csv --file Student.csv --headerline
2020-10-12T15:19:49.046+0530 connected to: localhost
2020-10-12T15:19:49.370+0530 imported 3 documents

C:\Program Files\MongoDB\Server\4.0\bin>mongoexport -d Database -c Customer -f cust_id,AccBal,AccType --type=csv -o Customer.csv
2020-10-12T15:38:24.897+0530 connected to: localhost
2020-10-12T15:38:24.901+0530 exported 5 records

C:\Program Files\MongoDB\Server\4.0\bin>

```

BDA_LAB (5)

Welcome * db.Customer.aggregate([...])

```

db.Customer.insert({cust_id:2,AccBal:4000,AccType:"A"})
db.Customer.insert({cust_id:1,AccBal:5000,AccType:"B"})
db.Customer.insert({cust_id:4,AccBal:3500,AccType:"Z"})
db.Customer.insert({cust_id:2,AccBal:6000,AccType:"A"})
db.Customer.find()

db.Customer.find({AccBal:{$gt:1200}, AccType:"Z"})

db.Customer.aggregate([
  {
    $group: {
      _id: "$cust_id",
      min_bal: {$min: "$AccBal"},
      max_bal: {$max: "$AccBal"}
    }
  }
])

mongoexport -d Database -c Customer -f cust_id,AccBal,AccType --type=csv -o Customer.csv
db.Customer.drop()

```

0.117 sec.

true

Home Insert Page Lay Formula Data Review View

A1

	A	B	C	D	E	F
1	cust_id	AccBal	AccType			
2	1	1600	Z			
3	2	4000	A			
4	1	5000	B			
5	4	3500	Z			
6	2	6000	A			

File View Options Window Help

BDA_LAB (5) Welcome * db.Customer.find()

Administrator: Command Prompt

```
2020-10-12T15:38:24.897+0530 connected to: localhost
2020-10-12T15:38:24.901+0530 exported 5 records

C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d Database -c Customer --type csv --file Customer.csv --headerline
2020-10-12T15:44:34.031+0530 connected to: localhost
2020-10-12T15:44:34.333+0530 imported 5 documents

C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d Database -c Customer --type csv --file Customer.csv --headerline
2020-10-12T15:45:21.167+0530 connected to: localhost
2020-10-12T15:45:21.462+0530 imported 5 documents

C:\Program Files\MongoDB\Server\4.0\bin>
db.Customer.drop()
mongoimport -d Database -c Customer --type csv --file Customer.csv --headerline
db.Customer.find()
```

Customer 0.002 sec.

Key	Value	Type
(1) ObjectId("5fb42cb900bf4c7c3a8683fe")	{ 4 fields }	Object
_id	ObjectId("5fb42cb900bf4c7c3a8683fe")	Objectid
cust_id	1	Int32
AccBal	1600	Int32
AccType	Z	String
(2) ObjectId("5fb42cb900bf4c7c3a8683ff")	{ 4 fields }	Object
(3) ObjectId("5fb42cb900bf4c7c3a868400")	{ 4 fields }	Object
(4) ObjectId("5fb42cb900bf4c7c3a868401")	{ 4 fields }	Object
(5) ObjectId("5fb42cb900bf4c7c3a868402")	{ 4 fields }	Object

LAB 3

Cassandra : Employee Keyspace

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 3. Update Employee name and Department of Emp-Id 121
4. Sort the details of Employee records based on salary
5. . Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
6. Update the altered table to add project names.
7. Create a TTL of 15 seconds to display the values of Employees.

1. Create a keyspace by name Employee

```
Administrator: Command Prompt - cqlsh
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\apache-cassandra-3.11.4\bin
C:\apache-cassandra-3.11.4\bin>cqlsh

WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE Employees WITH replication ={'class':'SimpleStrategy','replication_factor':3};
cqlsh> use Employees;
```

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

3. Insert the values into the table in batch
and

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

```
cqlsh:employees> CREATE COLUMNFAMILY employee_info(Emp_id INT PRIMARY KEY,Emp_name VARCHAR,Designation VARCHAR,Date_of_joining VARCHAR,Salary FLOAT,Dept_name VARCHAR);
cqlsh:employees> BEGIN BATCH INSERT INTO employee_info(emp_id,emp_name,designation,Date_of_joining,Salary,Dept_name) values(118,'ABC','CEO','01/01/1998',10000,'IT');INSERT INTO employee_info(emp_id,emp_name,designation,Date_of_joining,Salary,Dept_name) values(119,'EFG','HR Head','01/05/1999',15000,'SALES');INSERT INTO employee_info(emp_id,emp_name,designation,Date_of_joining,Salary,Dept_name) values(121,'LMN','CTO','05/01/1998',20000,'IT');INSERT INTO employee_info(emp_id,emp_name,designation,Date_of_joining,Salary,Dept_name) values(122,'XYZ','Manager','07/06/2002',20000,'IT');APPLY BATCH;
cqlsh:employees> Select * from employee_info;
+-----+-----+-----+-----+-----+-----+
| emp_id | date_of_joining | dept_name | designation | emp_name | salary |
+-----+-----+-----+-----+-----+-----+
| 118 | 01/01/1998 | IT | CEO | ABC | 10000 |
| 122 | 07/06/2002 | IT | Manager | XYZ | 20000 |
| 121 | 05/01/1998 | SALES | CTO | LMN | 20000 |
| 119 | 01/05/1999 | SALES | HR Head | EFG | 15000 |
+-----+-----+-----+-----+-----+-----+
(4 rows)
cqlsh:employees> UPDATE employee_info set emp_name='New Name',dept_name='New DEPT' where emp_id=121;
cqlsh:employees> Select * from employee_info;
+-----+-----+-----+-----+-----+-----+
| emp_id | date_of_joining | dept_name | designation | emp_name | salary |
+-----+-----+-----+-----+-----+-----+
| 118 | 01/01/1998 | IT | CEO | ABC | 10000 |
| 122 | 07/06/2002 | IT | Manager | XYZ | 20000 |
| 121 | 05/01/1998 | New DEPT | CTO | New Name | 20000 |
| 119 | 01/05/1999 | SALES | HR Head | EFG | 15000 |
+-----+-----+-----+-----+-----+-----+
(4 rows)
```

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

And

6. Update the altered table to add project names.

And

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

```
cqlsh:employees> ALTER table employee_info add PROJECT SET<VARCHAR>;
cqlsh:employees> Select * from employee_info order by emp_id;
InvalidRequest: Error from server: code=2200 [Invalid query] message="ORDER BY is only supported when the partition key is restricted by an EQ or an IN."
cqlsh:employees> Select * from employee_info;
+-----+-----+-----+-----+-----+-----+
| emp_id | date_of_joining | dept_name | designation | emp_name | project |
+-----+-----+-----+-----+-----+-----+
| 118 | 01/01/1998 | IT | CEO | ABC | null |
| 122 | 07/06/2002 | IT | Manager | XYZ | null |
| 121 | 05/01/1998 | New DEPT | CTO | New Name | null |
| 119 | 01/05/1999 | SALES | HR Head | EFG | null |
+-----+-----+-----+-----+-----+-----+
(4 rows)
cqlsh:employees> UPDATE employee_info set project=project+['P-1','P-2'] WHERE emp_id=122;
cqlsh:employees> Select * from employee_info;
+-----+-----+-----+-----+-----+-----+-----+
| emp_id | date_of_joining | dept_name | designation | emp_name | project | salary |
+-----+-----+-----+-----+-----+-----+-----+
| 118 | 01/01/1998 | IT | CEO | ABC | null | 10000 |
| 122 | 07/06/2002 | IT | Manager | XYZ | ['P-1', 'P-2'] | 20000 |
| 121 | 05/01/1998 | New DEPT | CTO | New Name | null | 20000 |
| 119 | 01/05/1999 | SALES | HR Head | EFG | null | 15000 |
+-----+-----+-----+-----+-----+-----+-----+
(4 rows)
cqlsh:employees> BEGIN BATCH INSERT INTO employee_info(emp_id,emp_name,designation,Date_of_joining,Salary,Dept_name) values
... (123,'New Name 1','Manager','05/06/2002',20000,'IT') USING TTL 15; APPLY BATCH;
cqlsh:employees> Select TTL(designation) from employee_info where emp_id=123;
+-----+
| ttl(designation) |
+-----+
| 6 |
+-----+
(1 rows)
cqlsh:employees> Select TTL(designation) from employee_info where emp_id=123;
+-----+
| ttl(designation) |
+-----+
| 0 |
+-----+
(0 rows)
```

LAB 4

Cassandra : Library Keyspace

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 keyspace by name **Library**

and

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**

```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.4 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE Library WITH replication ={'class':'SimpleStrategy','replication_factor':3};
cqlsh> use Library;
cqlsh:library> CREATE COLUMNFAMILY library_info(stud_id uuid,counter_value counter,stud_name VARCHAR,book_name
... VARCHAR,book_id INT,DOI VARCHAR,PRIMARY KEY(stud_id,stud_name,book_name,book_id,doi));
```

3.Insert the values into the table in batch

```
cqlsh:library> UPDATE library_info set counter_value=counter_value+1 where stud_id=uuid() and stud_name='ABC' and
... book_name='BDA' and book_id=2 and doi='01/06/2002';
cqlsh:library> UPDATE library_info set counter_value=counter_value+1 where stud_id=uuid() and stud_name='PQR' and
... book_name='WEB' and book_id=3 and doi='05/06/2008';
cqlsh:library> select * from library_info;

stud_id          | stud_name | book_name | book_id | doi      | counter_value
-----+-----+-----+-----+-----+-----+
cdc0af87-ed55-4916-bace-454b13383893 |      ABC |       BDA |       2 | 01/06/2002 |           1
d69a66f7-0572-4a2b-ba35-4c9b15432bd6 |      PQR |       WEB |       3 | 05/06/2008 |           1

(2 rows)
```

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=cdc0af87-ed55-4916-bace-454b13383893 and stud_name='ABC' and book_name='BDA' and book_id=2 and doi='01/06/2002';
cqlsh:library> select * from library_info;
stud_id          | stud_name | book_name | book_id | doi           | counter_value
-----+-----+-----+-----+-----+-----+
cdc0af87-ed55-4916-bace-454b13383893 | ABC       | BDA       | 2        | 01/06/2002   | 2
d69a66f7-0572-4a2b-ba35-4c9b15432bd6 | PQR       | WEB       | 3        | 05/06/2008   | 1
(2 rows)
cqlsh:library> select * from library_info where counter_value=2 allow filtering;
stud_id          | stud_name | book_name | book_id | doi           | counter_value
-----+-----+-----+-----+-----+-----+
cdc0af87-ed55-4916-bace-454b13383893 | ABC       | BDA       | 2        | 01/06/2002   | 2
(1 row)

```

5.Export the created column to a csv file

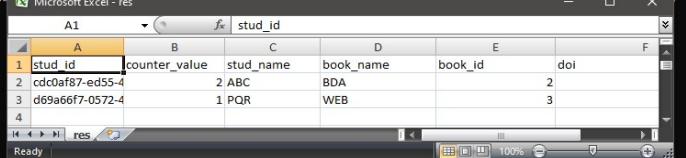
and

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

```

cqlsh:library> COPY library_info(stud_id,counter_value,stud_name,book_name,book_id,doi)TO 'C:\Users\VINOD\Desktop\res.csv' WITH HEADER =TRUE;
Using 1 child processes
Starting copy of library.library_info with columns [stud_id, counter_value, stud_name, book_name, book_id, doi].
Processed: 2 rows; Rate:      9 rows/s; Avg. rate:      0 rows/s
2 rows exported to 1 files in 5.986 seconds.
cqlsh:library> COPY library_info(stud_id,counter_value,stud_name,book_name,book_id,doi) FROM 'C:\Users\VINOD\Desktop\res.csv' WITH HEADER =TRUE;
Using 1 child processes
Starting copy of library.library_info with columns [stud_id, counter_value, stud_name, book_name, book_id, doi].
Process ImportProcess-2:      1 rows/s; Avg. rate:      1 rows/s
Traceback (most recent call last):
  File "C:\Python27\lib\multiprocessing\process.py", line 267, in _bootstrap
    self._run()
  File "C:\apache-cassandra-3.11.4\bin..\pylib\cqlshlib\copyutil.py", line 2328, in run
    self._close()
  File "C:\apache-cassandra-3.11.4\bin..\pylib\cqlshlib\copyutil.py", line 2332, in close
    self._session.cluster.shutdown()
  File "C:\apache-cassandra-3.11.4\bin..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 1259, in shutdown
    self._control_connection.shutdown()
  File "C:\apache-cassandra-3.11.4\bin..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\cluster.py", line 2850, in shutdown
    self._connection.close()
  File "C:\apache-cassandra-3.11.4\bin..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 373, in close
    AsyncoreConnection.create_timer(0, partial(asyncore.dispatcher.close, self))
  File "C:\apache-cassandra-3.11.4\bin..\lib\cassandra-driver-internal-only-3.11.0-bb96859b.zip\cassandra-driver-3.11.0-bb96859b\cassandra\io\asyncorereactor.py", line 335, in create_timer
    cls._loop.add_timer(timer)
AttributeError: 'NoneType' object has no attribute 'add_timer'
Processed: 2 rows; Rate:      0 rows/s; Avg. rate:      1 rows/s
2 rows imported from 1 files in 3.718 seconds (0 skipped).
cqlsh:library>

```



	A	B	C	D	E	F
	stud_id	counter_value	stud_name	book_name	book_id	doi
1	cdc0af87-ed55-4916-bace-454b13383893	2	ABC	BDA	2	01/06/2002
2	d69a66f7-0572-4a2b-ba35-4c9b15432bd6	1	PQR	WEB	3	05/06/2008
3						
4						

5. Hadoop : Word Count

Develop a MapReduce program to count the number of occurrences of words in a given file. (The program has been executed on Windows)

To start all the Hadoop deamons

```
> start-dfs.sh
```

```
> start-yarn.sh
```

To create a directory in hdfs

```
> hadoop fs -mkdir /input_dir
```

To view all the directories in hdfs

```
>hadoop fs -ls /
```

To copy a file from local system to hdfs directory

```
>hadoop fs -put C:/input_file.txt /input_dir
```

To view all files in /rgs1 hdfs directory

```
>hadoop fs -ls /input_dir/
```

To run a MapReduce program

```
> hadoop jar C:/MapReduceClient.jar wordcount /input_dir /output_dir
```

To view the output text

```
hadoop dfs -cat /output_dir/*
```

To stop all the Hadoop deamons

```
>stop-yarn.sh
```

```
>stop-dfs.sh
```

```

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class WordCount{
    public static class Map extends Mapper<LongWritable,Text,Text,IntWritable> {
        public void map(LongWritable key, Text value, Context context) throws
IOException,InterruptedException{
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                value.set(tokenizer.nextToken());
                context.write(value, new IntWritable(1));
            }
        }
    }
    public static class Reduce extends Reducer<Text,Iterable<IntWritable>,Text,IntWritable> {
        public void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException,InterruptedException {
            int sum=0;
            for(IntWritable x: values)
            {
                sum+=x.get();
            }
            context.write(key, new IntWritable(sum));
        }
    }
    public static void main(String[] args) throws Exception {
        Configuration conf= new Configuration();
        Job job = new Job(conf,"My Word Count Program");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);
        job.setOutputKeyClass(Text.class);
    }
}

```

```
job.setOutputValueClass(IntWritable.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
Path outputPath = new Path(args[1]);
//Configuring the input/output path from the filesystem into the job
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
//deleting the output path automatically from hdfs so that we don't have to delete it
explicitly
outputPath.getFileSystem(conf).delete(outputPath);
//exiting the job only if the flag value becomes false
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}
```

Output

```

Administrator: Command Prompt

C:\WINDOWS\system32>hadoop dfs -cat /input_dir/Lab_1/
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
ravi kumar , bda lab , sample program for in Hadoop for counting the number of words in file .
C:\WINDOWS\system32>hadoop jar C:\hadoop-3.0.0\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.0.0.jar wordcount /input_dir/Lab_1 /output_dir/Lab_word_count
2020-12-19 19:27:43,564 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-19 19:27:46,258 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/VINOD/.staging/job_1608385473820_0001
2020-12-19 19:27:47,897 INFO input.FileInputFormat: Total input files to process : 1
2020-12-19 19:27:48,632 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-19 19:27:48,793 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-12-19 19:27:49,149 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1608385473820_0001
2020-12-19 19:27:49,154 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-19 19:27:58,278 INFO conf.Configuration: resource-types.xml not found
2020-12-19 19:27:58,318 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-19 19:27:51,345 INFO impl.YarnClientImpl: Submitted application application_1608385473820_0001
2020-12-19 19:27:51,488 INFO mapreduce.Job: The url to track the job: http://DESKTOP-MNVEQSP:8088/proxy/application_1608385473820_0001
2020-12-19 19:27:51,493 INFO mapreduce.Job: Running job: job_1608385473820_0001
2020-12-19 19:28:27,818 INFO mapreduce.Job: Job job_1608385473820_0001 running in uber mode : false
2020-12-19 19:28:27,825 INFO mapreduce.Job: map 0% reduce 0%
2020-12-19 19:28:46,742 INFO mapreduce.Job: map 100% reduce 0%
2020-12-19 19:29:07,111 INFO mapreduce.Job: map 100% reduce 100%
2020-12-19 19:29:18,254 INFO mapreduce.Job: Job job_1608385473820_0001 completed successfully
2020-12-19 19:29:18,553 INFO mapreduce.Job: Counters: 53
    File System Counters
        FILE: Number of bytes read=194
        FILE: Number of bytes written=416215
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=196
        HDFS: Number of bytes written=120
        HDFS: Number of read operations=8
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
    Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Bytes Written=120

C:\WINDOWS\system32>hadoop dfs -cat /output_dir/Lab_word_count/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
,          2
.          1
Hadoop   1
bda      1
counting     1
file      1
for       2
in        2
kumar     1
lab       1
number    1
of        1
program   1
ravi      1
sample    1
the       1
words     1

C:\WINDOWS\system32>

```

6. Hadoop : Average Temperature

Develop a MapReduce program to find average temperature. (The program has been executed on Windows)

To start all the Hadoop deamons

```
> start-dfs.sh
```

```
> start-yarn.sh
```

To create a directory in hdfs

```
> hadoop fs -mkdir /input_dir
```

To view all the directories in hdfs

```
>hadoop fs -ls /
```

To copy a file from local system to hdfs directory

```
>hadoop fs -put C:/1901 /input_dir
```

To view all files in /rgs1 hdfs directory

```
>hadoop fs -ls /input_dir/
```

To run a MapReduce program

```
> hadoop jar C:/AverageTempearture.jar wordcount /input_dir /output_dir
```

To view the output text

```
hadoop dfs -cat /output_dir/*
```

To stop all the Hadoop deamons

```
>stop-yarn.sh
```

```
>stop-dfs.sh
```

AverageDriver.java

```
import org.apache.hadoop.io.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.*;
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.java

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

public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{
```

```

public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException
{
    String line = value.toString();
    String year = line.substring(15,19);
    int temperature;
    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 != MISSING && quality.matches("[01459]"))
        context.write(new Text(year),new IntWritable(temperature));
}
}

```

AverageReducer.java

```

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;

public class AverageReducer extends Reducer <Text, IntWritable, Text, IntWritable >
{
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException, InterruptedException
{
    int max_temp = 0;
    int count = 0;
    for (IntWritable value : values)
    {
        max_temp += value.get();
        count+=1;
    }
    context.write(key, new IntWritable(max_temp/count));
}
}

```

```

C:\hadoop-3.0.0\sbin>hadoop jar C:\hadoop-3.0.0\share\hadoop\mapreduce\AverageProgram.jar /Average_Program /output_dir_avg_prg
2020-12-16 16:17:44,713 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-16 16:17:47,293 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2020-12-16 16:17:47,422 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/VINOD/.staging/job_1608114850843_0001
2020-12-16 16:17:48,349 INFO input.FileInputFormat: Total input files to process : 1
2020-12-16 16:17:49,000 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-16 16:17:49,168 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-12-16 16:17:49,201 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1608114850843_0001
2020-12-16 16:17:49,785 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-16 16:17:50,931 INFO conf.Configuration: resource-types.xml not found
2020-12-16 16:17:50,948 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-16 16:17:52,301 INFO impl.YarnClientImpl: Submitted application application_1608114850843_0001
2020-12-16 16:17:52,475 INFO mapreduce.Job: The url to track the job: http://DESKTOP-NNVEQSP:8088/proxy/application_1608114850843_0001/
2020-12-16 16:17:52,478 INFO mapreduce.Job: Running job: job_1608114850843_0001
2020-12-16 16:19:04,567 INFO mapreduce.Job: Job job_1608114850843_0001 running in uber mode : false
2020-12-16 16:19:04,719 INFO mapreduce.Job: map 0% reduce 0%
2020-12-16 16:19:34,962 INFO mapreduce.Job: map 100% reduce 0%
2020-12-16 16:20:05,694 INFO mapreduce.Job: map 100% reduce 100%
2020-12-16 16:20:24,870 INFO mapreduce.Job: Job job_1608114850843_0001 completed successfully
2020-12-16 16:20:25,251 INFO mapreduce.Job: Counters
File System Counters
    FILE: Number of bytes read=72210
    FILE: Number of bytes written=559331
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=888296
    HDFS: Number of bytes written=8
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=2
    HDFS: Number of write operations=2
Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=23599
    Total time spent by all reduces in occupied slots (ms)=26783
    Total time spent by all map tasks (ms)=23599

```

```

Total megabyte-milliseconds taken by all reduce tasks=27425792
Map-Reduce Framework
    Map input records=6565
    Map output records=6564
    Map output bytes=50076
    Map output materialized bytes=72210
    Input split bytes=106
    Combine input records=0
    Combine output records=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)=349
    CPU time spent (ms)=4651
    Physical memory (bytes) snapshot=477446144
    Virtual memory (bytes) snapshot=655536128
    Total committed heap usage (bytes)=358088704
    Peak Map Physical memory (bytes)=282361856
    Peak Map Virtual memory (bytes)=376946688
    Peak Reduce Physical memory (bytes)=195084288
    Peak Reduce Virtual memory (bytes)=278589440
Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
File Input Format Counters
    Bytes Read=888190
File Output Format Counters
    Bytes Written=8

```

```

C:\hadoop-3.0.0\sbin>hadoop dfs -cat /output_dir_avg_prg/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
1901      46

```

7. Hive : Employee Table

Write Queries in Hive to do the following

- 1. Create an external table named with the following attributes -> Empl_ID ->Emp_Name -> Designation -> Salary**
- 2. Load data into table from a given file**
- 3. Create a view to Generate a query to retrieve the employee details who earn a salary of more than Rs 30000.**
- 4. Alter the table to add a column Dept_Id and Generate a query to retrieve the employee details in order by using Dept_Id**
- 5. Generate a query to retrieve the number of employees in each department whose salary is greater than 30000**
- 6. Create another table Department with attributes -> Dept_Id ->Dept_name ->Emp_Id**
- 7. Display the cumulative details of each employee along with department details**

- 1. Create an external table named with the following attributes -> Empl_ID>Emp_Name->Designation -> Salary**

```
>CREATE DATABASE IF NOT EXISTS Emp_Database COMMENT 'Employee Details' WITH DBPROPERTIES('creator'='Abc');
```

```
>SHOW DATABASES;
```

```
>DESCRIBE DATABASE Emp_Database;
```

```
>USE Emp_Database;
```

```
> CREATE EXTERNAL TABLE IF NOT EXISTS ext_employee (EMP_ID INT,EMP_NAME STRING,DESIGNATION STRING,SALARY FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION '/EMP_LAB';
```

- 2. Load data into table from a given file**

```
>load data local inpath 'C:\Users\Vinod\Desktop\BDA_LAB_HIVE.txt' overwrite into table ext_employee;
```

```
>Select * from ext_employee;
3. Create a view to Generate a query to retrieve the employee details who earn a salary
of more than Rs 30000.
>CREATE VIEW EMP_VIEW AS SELECT * FROM ext_employee;

>SELECT EMP_ID,EMP_NAME,DESIGNATION,SALARY FROM EMP_VIEW WHERE
SALARY>30000;

4. Alter the table to add a column Dept_Id and Generate a query to retrieve the
employee details in order by using Dept_Id

>ALTER TABLE ext_employee ADD COLUMNS (DEPT_ID INT COMMENT
'DEPARTMENT ID' );

>DESCRIBE FORMATTED ext_employee;

> Select * from ext_employee;

> load data local inpath 'C:\Users\Vinod\Desktop\BDA_LAB_HIVE.txt' overwrite into
table ext_employee;

> Select * from ext_employee;

>SELECT * FROM ext_employee ORDER BY dept_id;

5. Generate a query to retrieve the number of employees in each department whose
salary is greater than 30000

>SELECT COUNT(*),DEPT_ID, FROM EXT_EMPLOYEE WHERE SALARY >
30000 GROUP BY DEPT_ID;

6. Create another table Department with attributes -> Dept_Id ->Dept_name ->Emp_Id

>CREATE EXTERNAL TABLE IF NOT EXISTS EXT_DEPARTMENT (DEPT_ID INT,
DEPT_NAME STRING, EMP_ID INT) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\t' LOCATION '/EMP_LAB';

> load data local inpath 'C:\Users\Vinod\Desktop\BDA_LAB_DEPT.txt' overwrite into
table ext_department;

>Select * from ext_department;

7. Display the cumulative details of each employee along with department details
```

SELECT * FROM EXT_EMPLOYEE e JOIN EXT_DEPARTMENT d ON e.DEPT_ID = d. DEPT_ID;

```
hive> create database if not exists Emp_Database comment 'Employee Details' with dbproperties('creator'='ABC')
2020-12-20T20:22:11,130 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:22:11,134 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T20:22:11,235 WARN [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager is set to instance of HiveAuthorizerFactory.
OK
Time taken: 0.427 seconds
2020-12-20T20:22:11,569 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:22:11,570 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to main
hive> show databases;
2020-12-20T20:22:22,493 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:22:22,494 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
OK
default
emp_database
Time taken: 0.042 seconds, Fetched: 2 row(s)
2020-12-20T20:22:22,607 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:22:22,610 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
hive> describe database emp_database
;
2020-12-20T20:22:46,410 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:22:46,411 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
OK
emp_database Employee Details      hdfs://0.0.0.19000/user/hive/warehouse/emp_database.db      VINOD      USER
Time taken: 0.144 seconds, Fetched: 1 row(s)
2020-12-20T20:22:46,607 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:22:46,610 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to main
hive> use emp_database;
```

```
hive> create external table if not exists ext_employee(Emp_ID INT, Emp_Name STRING,Designation STRING,Salary FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATE
ION '/EMP_LAB';
2020-12-20T20:32:58,394 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:32:58,395 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
OK
Time taken: 2.514 seconds
2020-12-20T20:33:00,923 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:33:00,932 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to main
hive> select * from ext_employee;
```

```
hive> load local table inputpath 'C:\Users\VINOD\Desktop\BDA LAB_HIVE.txt' overwrite into table ext_employee;
2020-12-20T20:44:02,424 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:44:02,424 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
Loading data to table emp_database.ext_employee
2020-12-20T20:44:02,830 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.19000/EMP_LAB
OK
Time taken: 1.06 seconds
2020-12-20T20:44:03,493 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:44:03,494 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to main
hive> select * from ext_employee;
2020-12-20T20:44:06,966 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:44:06,967 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T20:44:07,275 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20_20-44-06_971_105647044355075027-1-mm-10001/.hive-staging_hive_2020-12-20_20-44-06_971_105647044355075027-1
OK
1      ABC      HR      15000.0
2      POR      MANAGER    18000.0
3      XYZ      SUPERVISOR 70000.0
Time taken: 0.385 seconds, Fetched: 3 row(s)
```

```
hive> create view Emp_View as select * from ext_employee;
2020-12-20T20:47:36,475 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:47:36,476 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
OK
Time taken: 0.747 seconds
2020-12-20T20:47:37,270 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:47:37,272 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to main
hive> select emp_id,emp_name,designation from Emp_View;
2020-12-20T20:48:23,251 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:48:23,253 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T20:48:23,741 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20_20-48-23_257_7373329983316641720-1-mm-10001/.hive-staging_hive_2020-12-20_20-48-23_257_7373
329983316641720-1
OK
1      ABC      HR
2      POR      MANAGER
3      XYZ      SUPERVISOR
Time taken: 0.746 seconds, Fetched: 3 row(s)
2020-12-20T20:48:24,078 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:48:24,079 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to main
hive> select emp_id,emp_name,designation from Emp_View where Salary>30000;
2020-12-20T20:48:49,586 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:48:49,587 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T20:48:51,163 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20_20-48-49_590_626409204284628194-1-mm-10001/.hive-staging_hive_2020-12-20_20-48-49_590_626409204284628194-1
OK
2      POR      MANAGER
3      XYZ      SUPERVISOR
Time taken: 1.699 seconds, Fetched: 2 row(s)
```

```

hive> select emp_id,emp_name,designation,salary from Emp_view where Salary>30000;
2020-12-20T20:49:16,665 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:49:16,666 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T20:49:17,130 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20_20-49-16_670_7074403349731210635-1/-mr-10001/.hive-staging_hive_2020-12-20_20-49-16_670_7074
03349731210635-1
OK
2      POR      MANAGER 18000.0
3      XYZ      SUPERVISOR    70000.0
Time taken: 0.679 seconds, Fetched: 2 row(s)

hive> ALTER TABLE Ext_employee ADD COLUMNS (Dept_ID INT COMMENT 'DEPARTMENT ID');
2020-12-20T20:56:52,770 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:56:52,774 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
OK
Time taken: 0.667 seconds
2020-12-20T20:56:53,446 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b
083-429c-b314-76431ef6822d
2020-12-20T20:56:53,447 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to main
hive> SELECT * FROM EXT_EMPLOYEE;
2020-12-20T20:57:12,258 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T20:57:12,259 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to C7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T20:57:12,669 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20-57-12_263_6750815377388133904-1/-mr-10001/.hive-staging_hive_2020-12-20-57-12_263_6750
815377388133904-1
OK
1      ABC      HR      15000.0 NULL
2      POR      MANAGER 18000.0 NULL
3      XYZ      SUPERVISOR    70000.0 NULL
Time taken: 0.449 seconds, Fetched: 3 row(s)

hive> load data local inpath 'C:\Users\VINOD\Desktop\BDA_LAB_HIVE.txt' overwrite into table ext_employee;
2020-12-20T21:47:33,210 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T21:47:33,211 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
Loading data to table emp_database.ext_employee
2020-12-20T21:47:33,616 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/EMP_LAB
OK
Time taken: 1.299 seconds
2020-12-20T21:47:34,425 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b
083-429c-b314-76431ef6822d
2020-12-20T21:47:34,426 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.ql.session.SessionState - Resetting thread name to main
hive> SELECT * FROM EXT_EMPLOYEE;
2020-12-20T21:47:41,726 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T21:47:41,728 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T21:47:42,452 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20-21-47-41_732_8121402402166991982-1/-mr-10001/.hive-staging_hive_2020-12-20-21-47-41_732_8121
402402166991982-1
OK
1      ABC      HR      15000.0 10
2      POR      MANAGER 18000.0 8
3      XYZ      SUPERVISOR    70000.0 15
Time taken: 0.767 seconds, Fetched: 3 row(s)

hive> select * from ext_employee order by dept_id;
2020-12-20T21:49:46,594 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T21:49:46,595 INFO [main] org.apache.hadoop.hive.ql.session.SessionState - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T21:49:47,058 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20_21-49-46_599_41435238036894675-1/-mr-10001/.hive-staging_hive_2020-12-20_21-49-46_599_414352
38036894675-1
Query ID = VINOD_20201220214946_2dc66e7-1c4b-427e-b18c-5fbe4e946655
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
2020-12-20T21:49:48,155 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.conf.Configuration.deprecation - mapred.submit.replication is deprecated. Instead, use mapreduce.client.submit.file.replication
2020-12-20T21:49:54,174 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2020-12-20T21:49:55,238 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.conf.Configuration - resource-types.xml not found
Starting Job = job_1608474912263_0001, Tracking URL = http://DESKTOP-NVNEQSP:8088/proxy/application_1608474912263_0001/
Kill Command = C:\hadoop-3.0.0\bin\mapred job -kill job_1608474912263_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-12-20 21:50:43,416 Stage-1 map = %, reduce = %
2020-12-20 21:51:02,408 Stage-1 map = 100%, reduce = %, Cumulative CPU 3.998 sec
2020-12-20 21:51:32,588 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 9.809 sec
MapReduce Total cumulative CPU time: 9 seconds 809 msec
Ended Job = job_1608474912263_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1  Cumulative CPU: 9.809 sec  HDFS Read: 11655 HDFS Write: 196 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 809 msec
OK
2      POR      MANAGER 18000.0 8
1      ABC      HR      15000.0 10
3      XYZ      SUPERVISOR    70000.0 15
Time taken: 119.516 seconds, Fetched: 3 row(s)

```

```

hive> select count(*),dept_id from ext_employee group by dept_id,salary having salary>'30000';
2020-12-20T22:10:58,716 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T22:10:58,717 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T22:10:59,130 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20_22-10-58_720_3903900699453963976-1/-mr-10001/.hive-staging_hive_2020-12-20_22-10-58_720_3903
900699453963976-1
Query ID = VINOD_20201220221058_0107d894-02a8-402d-8fbd-f3bbala279b
Total jobs = 1
Launching Job 1 out of 1
2020-12-20T22:10:59,245 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
2020-12-20T22:11:01,842 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
Starting Job = job_1608474912263_0003, Tracking URL = http://DESKTOP-NVNEQSP:8088/proxy/application_1608474912263_0003/
Kill Command = C:\hadoop-3.0.0\bin\mapred job -kill job_1608474912263_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-12-20 22:11:33,972 Stage-1 map = %, reduce = %
2020-12-20 22:11:50,216 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.982 sec
2020-12-20 22:12:11,953 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 10.809 sec
MapReduce Total cumulative CPU time: 10 seconds 809 msec
Ended Job = job_1608474912263_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 10.809 sec HDFS Read: 14935 HDFS Write: 120 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 809 msec
OK
1      15
1      8
Time taken: 86.381 seconds, Fetched: 2 row(s)

```

```

hive> create external table if not exists ext_department(Dept_ID INT, Dept_Name STRING,Emp_Id int) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION '/EMP_LAB';
2020-12-20T22:16:19,628 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T22:16:19,629 INFO [main] org.apache.hadoop.hive.session.SessionState - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
OK
Time taken: 0.499 seconds
2020-12-20T22:16:20,131 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b
083-429c-b314-76431ef6822d
2020-12-20T22:16:20,131 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Resetting thread name to  main
hive> load data local inpath 'c:/Users/VINOD/Desktop\BDA_LAB_DEPT.txt' overwrite into table ext_department;
2020-12-20T22:42:27,079 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T22:42:27,079 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
Loading data to table emp_database.ext_department
2020-12-20T22:42:27,492 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/EMP_LAB
OK
Time taken: 0.944 seconds
2020-12-20T22:42:28,026 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b
083-429c-b314-76431ef6822d
2020-12-20T22:42:28,028 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.conf.HiveConf - Resetting thread name to  main
hive> select * from ext_department;
2020-12-20T22:42:48,487 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-20T22:42:48,489 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-20T22:42:48,493 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-20_22-42-48_492_2367592337050693105-1/-mr-10001/.hive-staging_hive_2020-12-20_22-42-48_492_2367
592337050693105-1
OK
8      MANAGEMENT      2
10     HR      1
15     FINANCE      3
Time taken: 0.535 seconds, Fetched: 3 row(s)

```

```

hive> SELECT e.* FROM EXT_DEPARTMENT d Join ext_employee e on e.dept_id=d.dept_id;
2020-12-21T11:52:33,993 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Using the default value passed in for log id: c7e601b8-b083-429c-b314-76431ef6822d
2020-12-21T11:52:33,993 INFO [main] org.apache.hadoop.hive.conf.HiveConf - Updating thread name to c7e601b8-b083-429c-b314-76431ef6822d main
2020-12-21T11:52:34,.887 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.hive.common.FileUtils - Creating directory if it doesn't exist: hdfs://0.0.0.
0:19000/tmp/hive/VINOD/c7e601b8-b083-429c-b314-76431ef6822d/hive_2020-12-21_11-52-33_907_7800528759286007367-1/-mr-10001/.hive-staging_hive_2020-12-21_11-52-33_907_7800
528759286007367-1
Query ID = VINOD_20201221115233_0c910dc4-088c-48b0-8e13-10c73646349f
Total jobs = 1
Launching Job 1 out of 1
2020-12-21T11:52:34,.900 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
2020-12-21T11:52:36,.995 INFO [c7e601b8-b083-429c-b314-76431ef6822d main] org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
Starting Job = job_1608474912263_0030, Tracking URL = http://DESKTOP-NVNEQSP:8088/proxy/application_1608474912263_0030/
Kill Command = C:\hadoop-3.0.0\bin\mapred job -kill job_1608474912263_0030
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-12-21 11:53:10,910 Stage-1 map = %, reduce = %
2020-12-21 11:53:28,401 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 4.811 sec
2020-12-21 11:53:53,662 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 9.887 sec
MapReduce Total cumulative CPU time: 9 seconds 887 msec
Ended Job = job_1608474912263_0030
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 9.887 sec HDFS Read: 12122 HDFS Write: 196 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 887 msec
OK
2      POR      MANAGER 180000.0      8
1      ABC      HR      15000.0 10
3      XYZ      SUPERVISOR 70000.0 15
Time taken: 92.721 seconds, Fetched: 3 row(s)

```

