

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

BIG DATA ANALYTICS (20CS6PEBDA)

Submitted by

Piyush Dubey (1BM19CS221)

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 **Piyush Dubey(1BM19CS221)**, who is a 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.

Antara Roy Choudhury
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	Employee Database using Cassandra	1-2
2	Library Database using Cassandra	3-4
3	MongoDB Student program	5-15

Course Outcome

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

Program 1: Employee Database using Cassandra

1. Create a keyspace by name Employee

```
CREATE KEYSPACE employee WITH REPLICATION={ 'class' : 'SimpleStrategy',  
'replication_factor' : 1 };
```

```
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

```
create table employee_info(emp_id int PRIMARY KEY, emp_name text, designation text,  
date_of_joining timestamp, salary double, dept_name text);
```

3. Insert the values into the table in batch

```
BEGIN BATCH
```

```
INSERT INTO
```

```
employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)  
VALUES(100,'TANYA','MANAGER','2020-09-11',30000,'TESTING')
```

```
... INSERT INTO
```

```
employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)  
VALUES(111,'SRIRAM','ASSOCIATE','2020-06-22',25000,'DEVELOPING')
```

```
... INSERT INTO
```

```
employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)  
VALUES(121,'SHIVA','MANAGER','2020-03-30',35000,'HR')
```

```
... APPLY BATCH;
```

```
SELECT * FROM employee_info;
```

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

```
UPDATE employee_info SET emp_name = 'SHAAN' WHERE emp_id = 121; SELECT *  
FROM employee_info;
```

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.

```
ALTER TABLE employee_info ADD projects text;
```

6. Update the altered table to add project names.

```
UPDATE employee_info SET projects = 'chat app' WHERE emp_id = 111;
```

```
UPDATE employee_info SET projects = 'campusx' WHERE emp_id = 121;
```

```
UPDATE employee_info SET projects = 'canteen app' WHERE emp_id = 100;  
SELECT * FROM employee_info;
```

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

```
INSERT INTO  
employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)  
VALUES(110,'SAM','ASSOCIATE','2020-01-11',33000,'TESTING') USING TTL 15;  
  
SELECT TTL(emp_name) from employee_info WHERE emp_id = 110; SELECT * FROM  
employee_info;
```

Program 2:

Library Database using Cassandra

1.Create a keyspace by name Library

```
CREATE KEYSPACE library WITH REPLICATION={ 'class' : 'SimpleStrategy',  
'replication_factor' : 1};
```

```
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

```
create table library_info(stud_id int, counter_value Counter, stud_name text,book_name text,  
date_of_issue timestamp, book_id int, PRIMARY  
KEY(stud_id,stud_name,book_name,date_of_issue,book_id));
```

3. Insert the values into the table in batch

```
UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id = 111 and  
stud_name = 'SAM' and book_name = 'ML' and date_of_issue = '2020-10-11'and book_id = 200;
```

```
UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id = 112 and  
stud_name = 'SHAAN' and book_name = 'BDA' and date_of_issue = '2020-09-21'and book_id =  
300;
```

```
UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id = 113 and  
stud_name = 'AYMAN' and book_name = 'OOMD' and date_of_issue = '2020-04-01'and  
book_id = 400;
```

```
SELECT * FROM library_info;
```

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

```
UPDATE library_info SET counter_value = counter_value + 1 WHERE stud_id = 112 and  
stud_name = 'SHAAN' and book_name = 'BDA' and date_of_issue = '2020-09-21'and book_id =  
300;
```

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

```
SELECT * FROM library_info WHERE stud_id = 112;
```

6. Export the created column to a csv file

```
COPY Library_Info(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_val ue)  
TO 'e:\libraryInfo.csv';
```

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

```
create table library_info2(stud_id int, counter_value Counter, stud_name text,book_name text,  
date_of_issue timestamp, book_id int, PRIMARY  
KEY(stud_id,stud_name,book_name,date_of_issue,book_id));  
  
COPY library_info2(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value)  
FROM 'e:\libraryInfo.csv';
```

Program 3 :

Student MongoDB Program

```
> use mySTUD;
switched to db mySTUD
> db.getCollectionNames()
[ ]
> db.createCollection("Student");
{ "ok" : 1 }
> db.getCollectionNames()
[ "Student" ]
> db.Student.insert({_id: 1, Name:"John", USN: "1B22CS001",Semester: 6,Dept_name: "CSE",
CGPA: 9.6, Hobbies : ["Reading","Gardening"]})
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id: 4, Name:"Arthur", USN: "1B22CS041",Semester: 6,Dept_name:
"CSE", CGPA: 8.6, Hobbies : ["Novel Reading"]})
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id: 3, Name:"Horris", USN: "1B22EE021",Semester: 5,Dept_name:
"EEE", CGPA: 9.3, Hobbies : ["eSports"]})
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id: 7, Name:"Hritik", USN: "1B22CS014",Semester: 5,Dept_name:
"CSE", CGPA: 8.7, Hobbies : ["Reading"]})
WriteResult({ "nInserted" : 1 })
> db.Student.find().pretty()
{
  "_id" : 1,
  "Name" : "John",
  "USN" : "1B22CS001",
  "Semester" : 6,
```



```
    "Dept_name" : "CSE",
    "CGPA" : 9.6,
    "Hobbies" : [
        "Reading",
        "Gardening"
    ]
}
{
    "_id" : 4,
    "Name" : "Arthur",
    "USN" : "1B22CS041",
    "Semester" : 6,
    "Dept_name" : "CSE",
    "CGPA" : 8.6,
    "Hobbies" : [
        "Novel Reading"
    ]
}
{
    "_id" : 3,
    "Name" : "Horris",
    "USN" : "1B22EE021",
    "Semester" : 5,
    "Dept_name" : "EEE",
    "CGPA" : 9.3,
    "Hobbies" : [
        "eSports"
    ]
}
```

```

}
{
  "_id" : 7,
  "Name" : "Hritik",
  "USN" : "1B22CS014",
  "Semester" : 5,
  "Dept_name" : "CSE",
  "CGPA" : 8.7,
  "Hobbies" : [
    "Reading"
  ]
}

```

```

> db.Student.update({_id: 3, Name:"Horris", USN: "1B22EE021",Semester: 5,Dept_name:
"EEE", CGPA: 9.3},{ $set: {Hobbies:"Skating"}},{upset:true});

```

```

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

```

> db.Student.find().pretty()

```

```

{
  "_id" : 1,
  "Name" : "John",
  "USN" : "1B22CS001",
  "Semester" : 6,
  "Dept_name" : "CSE",
  "CGPA" : 9.6,
  "Hobbies" : [
    "Reading",
    "Gardening"
  ]
}

```

```
{
  "_id" : 4,
  "Name" : "Arthur",
  "USN" : "1B22CS041",
  "Semester" : 6,
  "Dept_name" : "CSE",
  "CGPA" : 8.6,
  "Hobbies" : [
    "Novel Reading"
  ]
}
```

```
{
  "_id" : 3,
  "Name" : "Horris",
  "USN" : "1B22EE021",
  "Semester" : 5,
  "Dept_name" : "EEE",
  "CGPA" : 9.3,
  "Hobbies" : "Skating"
}
```

```
{
  "_id" : 7,
  "Name" : "Hritik",
  "USN" : "1B22CS014",
  "Semester" : 5,
  "Dept_name" : "CSE",
  "CGPA" : 8.7,
  "Hobbies" : [
```

"Reading"

]

}

```
> db.Student.find({}, {StudName:1,Semester:1,_id:0});
```

```
{ "Semester" : 6 }
```

```
{ "Semester" : 6 }
```

```
{ "Semester" : 5 }
```

```
{ "Semester" : 5 }
```

```
> db.Student.find({}, {Name:1,Semester:1,_id:0});
```

```
{ "Name" : "John", "Semester" : 6 }
```

```
{ "Name" : "Arthur", "Semester" : 6 }
```

```
{ "Name" : "Horris", "Semester" : 5 }
```

```
{ "Name" : "Hritik", "Semester" : 5 }
```

```
> db.Student.find({Semester:{Seq:5}}).pretty();
```

```
{
```

```
  "_id" : 3,
```

```
  "Name" : "Horris",
```

```
  "USN" : "1B22EE021",
```

```
  "Semester" : 5,
```

```
  "Dept_name" : "EEE",
```

```
  "CGPA" : 9.3,
```

```
  "Hobbies" : "Skating"
```

```
}
```

```
{
```

```
  "_id" : 7,
```

```
  "Name" : "Hritik",
```

```
  "USN" : "1B22CS014",
```

```

        "Semester" : 5,
        "Dept_name" : "CSE",
        "CGPA" : 8.7,
        "Hobbies" : [
            "Reading"
        ]
    }
> db.Student.count();
4
> db.Student.find().sort({Name:-1}).pretty();
{
  "_id" : 1,
  "Name" : "John",
  "USN" : "1B22CS001",
  "Semester" : 6,
  "Dept_name" : "CSE",
  "CGPA" : 9.6,
  "Hobbies" : [
    "Reading",
    "Gardening"
  ]
}
{
  "_id" : 7,
  "Name" : "Hritik",
  "USN" : "1B22CS014",
  "Semester" : 5,
  "Dept_name" : "CSE",

```

```

        "CGPA" : 8.7,
        "Hobbies" : [
            "Reading"
        ]
    }
    {
        "_id" : 3,
        "Name" : "Horris",
        "USN" : "1B22EE021",
        "Semester" : 5,
        "Dept_name" : "EEE",
        "CGPA" : 9.3,
        "Hobbies" : "Skating"
    }
    {
        "_id" : 4,
        "Name" : "Arthur",
        "USN" : "1B22CS041",
        "Semester" : 6,
        "Dept_name" : "CSE",
        "CGPA" : 8.6,
        "Hobbies" : [
            "Novel Reading"
        ]
    }
}

```

```

(base) bmsce@bmsce-Precision-T1700:~$ mongoexport --host localhost --db mySTUD
--collection Student --type=csv --fields="_id,Name,USN,Semester,Dept_name,CGPA,Hobbies"
--out /home/bmsce/Desktop/output.csv

```

2022-05-06T12:13:37.350+0530 connected to: localhost

2022-05-06T12:13:37.351+0530 exported 4 records

(base) bmsce@bmsce-Precision-T1700:~\$ mongo

MongoDB shell version v3.6.8

connecting to: mongodb://127.0.0.1:27017

Implicit session: session { "id" : UUID("aab8226-3ced-43d4-97fb-b0d55827849c") }

MongoDB server version: 3.6.8

Server has startup warnings:

2022-05-06T11:28:08.073+0530 I STORAGE [initandlisten]

2022-05-06T11:28:08.073+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine

2022-05-06T11:28:08.073+0530 I STORAGE [initandlisten] ** See <http://dochub.mongodb.org/core/prodnotes-filesystem>

2022-05-06T11:28:13.281+0530 I CONTROL [initandlisten]

2022-05-06T11:28:13.281+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.

2022-05-06T11:28:13.281+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.

2022-05-06T11:28:13.281+0530 I CONTROL [initandlisten]

> use mySTUD;

switched to db mySTUD

> db.Student.update({_id:4},{ \$set:{Location:"Network"}})

2022-05-06T12:16:35.289+0530 E QUERY [thread1] SyntaxError: illegal character @ (shell):1:42

> db.Student.update({_id:4},{ \$set:{Location:"Network"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.Student.find().pretty()

{

 "_id" : 1,

 "Name" : "John",

```
"USN" : "1B22CS001",
"Semester" : 6,
"Dept_name" : "CSE",
"CGPA" : 9.6,
"Hobbies" : [
    "Reading",
    "Gardening"
]
}
{
    "_id" : 4,
    "Name" : "Arthur",
    "USN" : "1B22CS041",
    "Semester" : 6,
    "Dept_name" : "CSE",
    "CGPA" : 8.6,
    "Hobbies" : [
        "Novel Reading"
    ],
    "Location" : "Network"
}
{
    "_id" : 3,
    "Name" : "Horris",
    "USN" : "1B22EE021",
    "Semester" : 5,
    "Dept_name" : "EEE",
    "CGPA" : 9.3,
```



```

        "Hobbies" : "Skating"
    }
    {
        "_id" : 7,
        "Name" : "Hritik",
        "USN" : "1B22CS014",
        "Semester" : 5,
        "Dept_name" : "CSE",
        "CGPA" : 8.7,
        "Hobbies" : [
            "Reading"
        ]
    }
}
> db.Student.find().sort({Name:1}).pretty();
{
    "_id" : 4,
    "Name" : "Arthur",
    "USN" : "1B22CS041",
    "Semester" : 6,
    "Dept_name" : "CSE",
    "CGPA" : 8.6,
    "Hobbies" : [
        "Novel Reading"
    ],
    "Location" : "Network"
}
{
    "_id" : 3,

```

```
    "Name" : "Horris",
    "USN" : "1B22EE021",
    "Semester" : 5,
    "Dept_name" : "EEE",
    "CGPA" : 9.3,
    "Hobbies" : "Skating"
}
{
    "_id" : 7,
    "Name" : "Hritik",
    "USN" : "1B22CS014",
    "Semester" : 5,
    "Dept_name" : "CSE",
    "CGPA" : 8.7,
    "Hobbies" : [
        "Reading"
    ]
}
{
    "_id" : 1,
    "Name" : "John",
    "USN" : "1B22CS001",
    "Semester" : 6,
    "Dept_name" : "CSE",
    "CGPA" : 9.6,
    "Hobbies" : [
        "Reading",
        "Gardening" ]
}
```