VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

BIG DATA ANALYTICS
(20CS6PEBDA)

Submitted by

SHREEHARI KULKARNI(1BM19CS153)

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 **SHREEHARI KULKARNI(1BM19CS153),** 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.

Antara Roy Choudhury
Nameof the Lab-Incharge
Designation
Department of CSE
BMSCE, Bengaluru

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

`

Index Sheet

Course Outcome

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

Program 1. Perform the following DB operations using Cassandra.

1. Create a key space by name Employee
2. Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name
3. Insert the values into the table in batch
4. Update Employee name and Department of Emp-Id 121
5. Sort the details of Employee records based on salary
6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.
7. Update the altered table to add project names.
8 Create a TTL of 15 seconds to display the values of Employees.

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

"Employee" system_auth                    system_schema system_views
system          system_distributed system_traces system_virtual_schema cqlsh>

USE "Employee";

cqlsh:Employee> create table employee_info( Emp_Id int PRIMARY KEY, Emp_Name text,
Designation text, Date_Of_joining timestamp, Salary int, Dept_Name text); cqlsh:Employee> describe
employee_info;

CREATE TABLE "Employee".employee_info (
        emp_id int PRIMARY KEY,
        date_of_joining timestamp, dept_name
        text,
        designation text,
        emp_name text,
        salary int
) WITH additional_write_policy = '99p' AND
        bloom_filter_fp_chance = 0.01
        AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'} AND cdc =
        false
        AND comment = ''
        AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy',
'max_threshold': '32', 'min_threshold': '4'}
        AND compression = {'chunk_length_in_kb': '16', 'class':
'org.apache.cassandra.io.compress.LZ4Compressor'}
        AND crc_check_chance = 1.0
        AND default_time_to_live = 0
        AND extensions = {}
        AND gc_grace_seconds = 864000
        AND max_index_interval = 2048
        AND memtable_flush_period_in_ms = 0
        AND min_index_interval = 128
        AND read_repair = 'BLOCKING'
        AND speculative_retry = '99p';

cqlsh:Employee> BEGIN BATCH
                    ... INSERT INTO
employee_info(Emp_Id,Emp_Name,Designation,Date_Of_joining,Salary,Dept_Name)
VALUES(121,'Rose','Software Developer','2021-03-16',80000,'IT')
                    ... INSERT INTO
employee_info(Emp_Id,Emp_Name,Designation,Date_Of_joining,Salary,Dept_Name)
VALUES(122,'Jane','Software Tester','2020-04-16',70000,'IT')
                    ... INSERT INTO
employee_info(Emp_Id,Emp_Name,Designation,Date_Of_joining,Salary,Dept_Name)
VALUES(123,'John','Manager','2020-05-25',65000,'Sales')
                    ... APPLY BATCH;
cqlsh:Employee> SELECT * FROM employee_info;
```

```
  emp_id | date_of_joining                  | dept_name | designation                |
emp_name | salary
--------+---------------------------------+-----------+--------------------
+----------+--------
       123 | 2020-05-25 00:00:00.000000+0000 |                Sales |                Manager |
John | 65000
```

```
      122 | 2020-04-16 00:00:00.000000+0000 |                  IT |          Software Tester |
Jane | 70000
      121 | 2021-03-16 00:00:00.000000+0000 |                  IT | Software Developer | Rose
| 80000

(3 rows)

cqlsh:Employee> UPDATE employee_info SET Emp_Name='Rosy', Dept_Name='Software'
WHERE Emp_Id=121;
cqlsh:Employee> SELECT * FROM employee_info;

 emp_id | date_of_joining                 | dept_name | designation                |
emp_name | salary
--------+---------------------------------+-----------+--------------------
+----------+--------
      123 | 2020-05-25 00:00:00.000000+0000 |       Sales |                  Manager |
John |      65000
      122 | 2020-04-16 00:00:00.000000+0000 |          IT |          Software Tester |
Jane |      70000
      121 | 2021-03-16 00:00:00.000000+0000 |    Software |    Software Developer |
Rosy |      80000

(3 rows)

cqlsh:Employee> ALTER TABLE employee_info
                ... ADD projects set<text>; cqlsh:Employee>
SELECT * FROM employee_info;

 emp_id | date_of_joining                 | dept_name | designation                |
emp_name | projects | salary
--------+---------------------------------+-----------+--------------------
+----------+----------+--------
      123 |    2020-05-25 00:00:00.000000+0000 |       Sales |                  Manager |
John |          null | 65000
      122 |    2020-04-16 00:00:00.000000+0000 |          IT |          Software Tester |
Jane |          null | 70000
      121 |    2021-03-16 00:00:00.000000+0000 |    Software |    Software Developer |
Rosy |          null | 80000

(3 rows)


cqlsh:Employee> UPDATE employee_info SET projects={'sales improvement proj','ad management
sys'} WHERE Emp_ID=123;
cqlsh:Employee> UPDATE employee_info SET projects={'company website','Employee management
app'} WHERE Emp_ID=121;
cqlsh:Employee> UPDATE employee_info SET projects={'company website testing'} WHERE
Emp_ID=122;
cqlsh:Employee> SELECT * FROM employee_info;
 emp_id | date_of_joining                 | dept_name | designation                |
emp_name | projects                                          | salary
--------+---------------------------------+-----------+--------------------
+----------+---------------------------------------------------+--------
      123 | 2020-05-25 00:00:00.000000+0000 |       Sales |                  Manager |
```

John | {'ad management sys', 'sales improvement proj'} | 65000
122 | 2020-04-16 00:00:00.000000+0000 | IT | Software Tester | Jane | {'company website testing'} | 70000
121 | 2021-03-16 00:00:00.000000+0000 | Software | Software Developer | Rosy | {'Employee management app', 'company website'} | 80000

```
(3 rows)


cqlsh:Employee> BEGIN BATCH
                ... INSERT INTO
employee_info(Emp_Id,Emp_Name,Designation,Date_Of_joining,Salary,Dept_Name,projects
) VALUES(124,'Joe','Intern','2021-03-20',25000,'IT',{'LMS'}) USING TTL 15
                ... APPLY BATCH;

cqlsh:Employee> SELECT * FROM employee_info;
 emp_id | date_of_joining                 | dept_name | designation          |
emp_name | projects                                                | salary
--------+---------------------------------+-----------+----------------------
+----------+---------------------------------------------------+--------
      124 | 2021-03-20 00:00:00.000000+0000 |        IT |               Intern |
Joe |                                                  {'LMS'} | 25000
      123 | 2020-05-25 00:00:00.000000+0000 |     Sales |              Manager |
John | {'ad management sys', 'sales improvement proj'} | 65000
      122 | 2020-04-16 00:00:00.000000+0000 |        IT | Software Tester |
Jane |                        {'company website testing'} | 70000
      121 | 2021-03-16 00:00:00.000000+0000 | Software | Software Developer | Rosy |
{'Employee management app', 'company website'} | 80000

(4 rows)

cqlsh:Employee> SELECT * FROM employee_info;
 emp_id | date_of_joining                 | dept_name | designation          |
emp_name | projects                                                | salary
--------+---------------------------------+-----------+----------------------
+----------+---------------------------------------------------+--------
      123 | 2020-05-25 00:00:00.000000+0000 |     Sales |              Manager |
John | {'ad management sys', 'sales improvement proj'} | 65000
      122 | 2020-04-16 00:00:00.000000+0000 |        IT | Software Tester |
Jane |                        {'company website testing'} | 70000
      121 | 2021-03-16 00:00:00.000000+0000 | Software | Software Developer | Rosy |
{'Employee management app', 'company website'} | 80000

(3 rows)
```

MONGO DB

```
>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:{$eq: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("aabd8226-3ced-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"
        ]
}
```

Program 2:

1 Create a key space 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

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


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';
```

```
cqlsh:library> UPDATE Library_Info SET Counter_Value=Counter_Value+1 where
Stud_Id=1 and Stud_Name='Ankit' and Book_name='BDA' and Book_id=111 and
Date_Of_Issue='2021-03-15';
cqlsh:library> UPDATE Library_Info SET Counter_value=Counter_value+1 where
Stud_Id=2 and Stud_Name='Jennifer' and Book_name='OOMD' and Book_id=112 and
Date_Of_Issue='2021-02-14';
cqlsh:library> UPDATE Library_Info SET Counter_value=Counter_value+1 where
Stud_Id=112 and Stud_Name='Aswin' and Book_name='BDA' and Book_id=1123 and
Date_Of_Issue='2021-01-18';
cqlsh:library> UPDATE Library_Info SET Counter_value=Counter_value+1 where
Stud_Id=112 and Stud_Name='Aswin' and Book_name='BDA' and Book_id=1123 and
Date_Of_Issue='2021-01-18';
```

```
select * from library_info;
cqlsh:library> UPDATE Library_Info SET Counter_Value=Counter_Value+1 where
Stud_Id=1 and Stud_Name='Ankit' and Book_name='BDA' and Book_id=111 and
```

Date_Of_Issue='2021-03-15';

select * from library_info where stud_id=112;

cqlsh:library> COPY Library_Info(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_value) TO '/home/bmsce/Desktop/library_info.csv';
Using 11 child processes

Starting copy of library.library_info with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 4 rows; Rate: 32 rows/s; Avg. rate: 32 rows/s
4 rows exported to 1 files in 0.135 seconds.

cqlsh:library> CREATE TABLE Library_Info_Import(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_Info_Import from '/home/bmsce/Desktop/library_info.csv';
Using 11 child processes

Starting copy of library.library_info_import with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 4 rows; Rate: 7 rows/s; Avg. rate: 10 rows/s
4 rows imported from 1 files in 0.
383 seconds (0 skipped).