VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

BIG DATA ANALYTICS
(20CS6PEBDA)

Submitted by

PRATHIKSHA KAMATH(1BM19CS118)

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 **PRATHIKSHA KAMATH(1BM19CS118),** 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
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

`

Index Sheet

Sl. No.	Experiment Title	Page No.
1	MongoDB- CRUD Demonstration	**1**
2	Employee database using Cassandra	**4**
3	Library database using Cassandra	**8**
4	Execution of HDFS Commands	**11**

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

LAB PROGRAM 1: MongoDB- CRUD Demonstration

1) Using MongoDB
i) Create a database for Students and Create a Student Collection (_id,Name, USN, Semester, Dept_Name, CGPA, Hobbies(Set)).

```
use student2;

db.createCollection("Student");
```

ii) Insert required documents to the collection.

```
> db.Student.insert({_id:1,Name: "Arun", sem:"V",dept: "CSE",CGPA: 8.2,hobbies: ['cycling','swimming']});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:2,Name: "Ananya", sem:"VII",dept: "ECE",CGPA: 6.8,hobbies: ['knitting','reading novels']});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:3,Name: "Bhuvan", sem:"III",dept: "ME",CGPA: 8.8,hobbies: ['chess','collecting coins']});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:4,Name: "Ajay", sem:"VII",dept: "CSE",CGPA: 9.1,hobbies: ['playing','reading novels']});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:5,Name: "Colin", sem:"V",dept: "CSE",CGPA: 7.1,hobbies: ['playing','watching TV']});
WriteResult({ "nInserted" : 1 })
```

```
> db.Student.find();
{ "_id" : 1, "Name" : "Arun", "sem" : "V", "dept" : "CSE", "CGPA" : 8.2, "hobbies" : [ "cycling", "swimming" ] }
{ "_id" : 2, "Name" : "Ananya", "sem" : "VII", "dept" : "ECE", "CGPA" : 6.8, "hobbies" : [ "knitting", "reading novels" ] }
{ "_id" : 3, "Name" : "Bhuvan", "sem" : "III", "dept" : "ME", "CGPA" : 8.8, "hobbies" : [ "chess", "collecting coins" ] }
{ "_id" : 4, "Name" : "Ajay", "sem" : "VII", "dept" : "CSE", "CGPA" : 9.1, "hobbies" : [ "playing", "reading novels" ] }
{ "_id" : 5, "Name" : "Colin", "sem" : "V", "dept" : "CSE", "CGPA" : 7.1, "hobbies" : [ "playing", "watching TV" ] }
>
```

iii) First Filter on “Dept_Name:CSE” and then group it on “Semester” and compute the Average CPGA for that semester and filter those documents where the “Avg_CPGA” is greater than 7.5.

```
>
db.Student.aggregate({$match:{dept:"CSE"}},{$group:{_id:"$sem",AverageCGPA:{$avg:"$CGPA"}}},{$match:{AverageCGPA:{$gt:7.5}}});
{ "_id" : "VII", "AverageCGPA" : 9.1 }
{ "_id" : "V", "AverageCGPA" : 7.6499999999999995 }
```

```
> db.Student.aggregate({$match:{dept:"CSE"}},{$group:{_id:"$sem",AverageCGPA:{$avg:"$CGPA"}}},{$match:{AverageCGPA:{$gt:7.5}}});
{ "_id" : "V", "AverageCGPA" : 7.6499999999999995 }
{ "_id" : "VII", "AverageCGPA" : 9.1 }
```

iv) Insert the document for “Bhuvan” in to the Students collection only if it does not already exist in the collection. However, if it is already present in the collection, then

update the document with new values. (Update his Hobbies to “Skating”) Use “Update else insert” (if there is an existing document, it will attempt to update it,
if there is no existing document then it will insert it).

```
> db.Student.update({_id: 3,Name:"Bhuvan"},{$set:{ Hobbies:"Skating"}},{upsert:true}
);
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

v)To display only the StudName and Grade from all the documents of the Students collection. The identifier_id should be suppressed and NOT displayed.

```
> db.Student.find({},{name:1,sem: 1, _id:0});
{ "sem" : "V" }
{ "sem" : "VII" }
{ "sem" : "III" }
{ "sem" : "VII" }
{ "sem" : "V" }
```

vi) To find those documents where the Grade is set to ‘VII’

```
> db.Student.find({sem:{$eq:"VII"}});
{ "_id" : 2, "Name" : "Ananya", "sem" : "VII", "dept" : "ECE", "CGPA" : 6.8, "hobbies" : [ "knitting", "reading novels" ] }
{ "_id" : 4, "Name" : "Ajay", "sem" : "VII", "dept" : "CSE", "CGPA" : 9.1, "hobbies" : [ "playing", "reading novels" ] }
```

vii)To find those documents from the Students collection where the Hobbies is set to either ‘Chess’ or is set to ‘Skating’.

```
> db.Student.find({Hobbies:{$in:['Chess','Skating']}});
{ "_id" : 3, "Name" : "Bhuvan", "sem" : "III", "dept" : "ME", "CGPA" : 8.8, "hobbies" : [ "chess", "collecting coins" ], "Hobbies" : "Skating" }
```

viii)To find documents from the Students collection where the StudName begins with “B”

```
> db.Student.find({Name: /^B/});
{ "_id" : 3, "Name" : "Bhuvan", "sem" : "III", "dept" : "ME", "CGPA" : 8.8, "hobbies" : [ "chess", "collecting coins" ], "Hobbies" : "Skating" }
```

ix) To find the number of documents in the Students collection.

```
> db.Student.count();
5
```

x) To sort the documents from the Students collection in the descending order of StudName.

```
> db.Student.find().sort({Name: -1});
{ "_id" : 5, "Name" : "Colin", "sem" : "V", "dept" : "CSE", "CGPA" : 7.1, "hobbies" : [ "playing", "watching TV" ] }
{ "_id" : 3, "Name" : "Bhuvan", "sem" : "III", "dept" : "ME", "CGPA" : 8.8, "hobbies" : [ "chess", "collecting coins" ], "Hobbies" : "Skating" }
{ "_id" : 1, "Name" : "Arun", "sem" : "V", "dept" : "CSE", "CGPA" : 8.2, "hobbies" : [ "cycling", "swimming" ] }
{ "_id" : 2, "Name" : "Ananya", "sem" : "VII", "dept" : "ECE", "CGPA" : 6.8, "hobbies" : [ "knitting", "reading novels" ] }
{ "_id" : 4, "Name" : "Ajay", "sem" : "VII", "dept" : "CSE", "CGPA" : 9.1, "hobbies" : [ "playing", "reading novels" ] }
```

xi) Command used to export MongoDB JSON documents from “Student” Collection into the “Students” database into a CSV file “Output.txt”.

```
> mongoexport --host localhost --db studentDB --collection Student --csv --out /Downloads/student.txt -fields "Name","sem"
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:14
```

LAB PROGRAM 2: Employee database using Cassandra

Program 1. Perform the following DB operations using Cassandra.

```
bmsce@bmsce-Precision-T1700:~/cassandra/apache-cassandra-3.11.0/bin$ 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.
```

1. Create a key space by name Employee

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

cqlsh> use Employee;
cqlsh:employee> describe keyspaces;

students       system_auth  system_distributed  system_traces
system_schema  system       employee

cqlsh> describe keyspace employee;

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

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

```
cqlsh:employee> CREATE TABLE Employee_Info(
           ... emp_id int PRIMARY KEY,
           ... emp_name text,
           ... designation text,
           ... date_of_joining timestamp,
           ... salary double,
           ... dept_name text
           ... );
cqlsh:employee> describe tables

employee_info
```

```
cqlsh:employee> describe table 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 double
) 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';
```

3. Insert the values into the table in batch

```
cqlsh:employee> BEGIN BATCH
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(1,'Arun','Technical head','2020-03-01',50000,'Technical')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(2,'Ajay','HR manager','2020-06-11',60000,'HR')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(3,'Riya','Editor','2022-01-11',22000,'Markrting')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(4,'Kshma','Software Engineer','2021-05-11',35000,'Technical')
           ... insert into employee_info(emp_id,emp_name,designation,date_of_joining,salary,dept_name)
           ... values(5,'Ram','HR employee','2021-02-11',25000,'HR')
           ... APPLY BATCH;
```

```
cqlsh:employee> select * from employee_info;
```

```
 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 | Markrting |            Editor |     Riya |  22000
```

```
cqlsh:employee> select * from employee_info
           ... ;

 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 | Markrting |            Editor |     Riya |  22000

(5 rows)
```

4. Update Employee name and Department of Emp-Id 3

cqlsh:employee> UPDATE employee_info SET emp_name = 'Raj' , dept_name = 'Sales' where emp_id = 3;

cqlsh:employee> select * from employee_info;

```
 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 |     Sales |            Editor |      Raj |  22000
```

```
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | designation       | emp_name | salary
--------+---------------------------------+-----------+-------------------+----------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |  50000
      2 | 2020-06-10 18:30:00.000000+0000 |        HR |        HR manager |     Ajay |  60000
      4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |  35000
      3 | 2022-01-10 18:30:00.000000+0000 |     Sales |            Editor |      Raj |  22000

(5 rows)
cqlsh:employee>
```

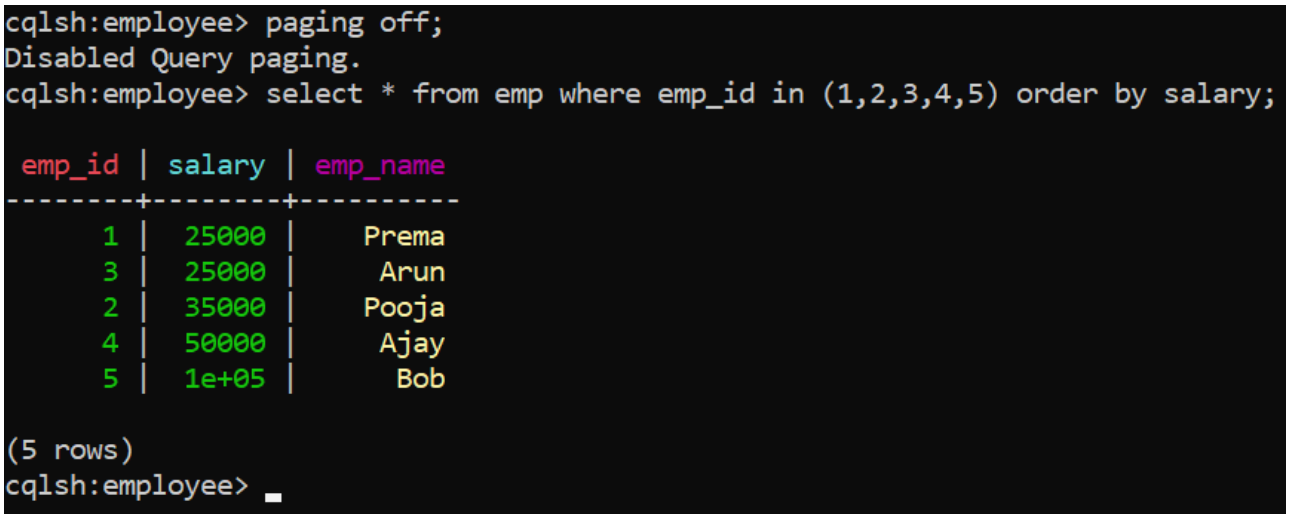
5. Sort the details of Employee records based on salary

```
CREATE TABLE emp(
        ...  emp_id int,
        ... salary double,
        ... emp_name text,
        ... PRIMARY KEY(emp_id,salary));
```

```
BEGIN BATCH
... insert into emp(emp_id,emp_name,salary) values(1,'Prema',25000)
... insert into emp(emp_id,emp_name,salary) values(2,'Pooja',35000)
... insert into emp(emp_id,emp_name,salary) values(3,'Arun',25000)
... insert into emp(emp_id,emp_name,salary) values(4,'Ajay',50000)
... insert into emp(emp_id,emp_name,salary) values(5,'Bob',100000)
... APPLY BATCH;
```

PAGING OFF;
select * from emp where emp_id in(1,2,3,4,5) order by salary;

```
 emp_id | salary | emp_name
--------+--------+----------
      1 |  25000 |    Prema
      3 |  25000 |     Arun
      2 |  35000 |    Pooja
      4 |  50000 |     Ajay
      5 |  1e+05 |      Bob
```



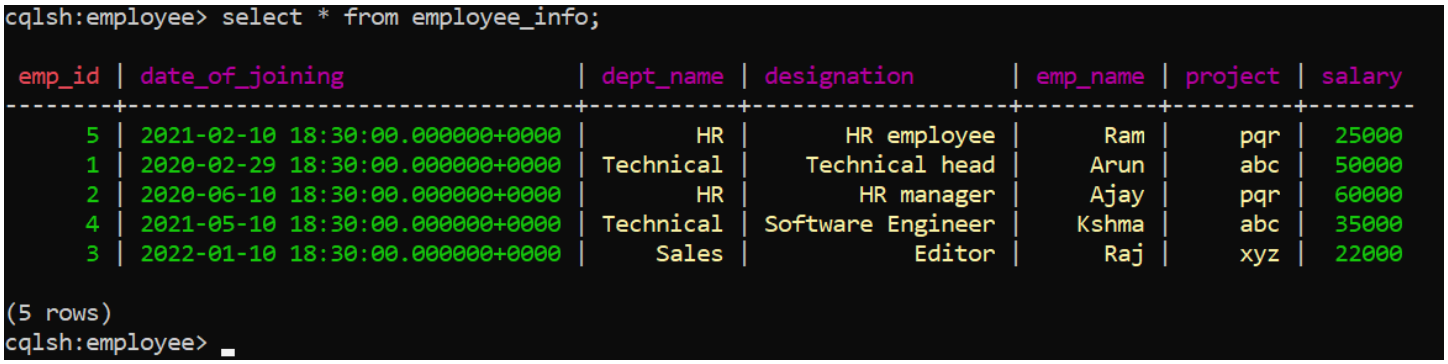
6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

```
cqlsh:employee> alter table employee_info
           ... add project text;
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | designation       | emp_name | project | salary
--------+---------------------------------+-----------+-------------------+----------+---------+--------
      5 | 2021-02-10 18:30:00.000000+0000 |        HR |       HR employee |      Ram |    null |  25000
      1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |    null |  50000
```

```
     2 | 2020-06-10 18:30:00.000000+0000 |       HR |       HR manager |     Ajay |    null |  60000
     4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |    null |  35000
     3 | 2022-01-10 18:30:00.000000+0000 |     Sales |          Editor |      Raj |    null |  22000

(5 rows)
```

7. Update the altered table to add project names.

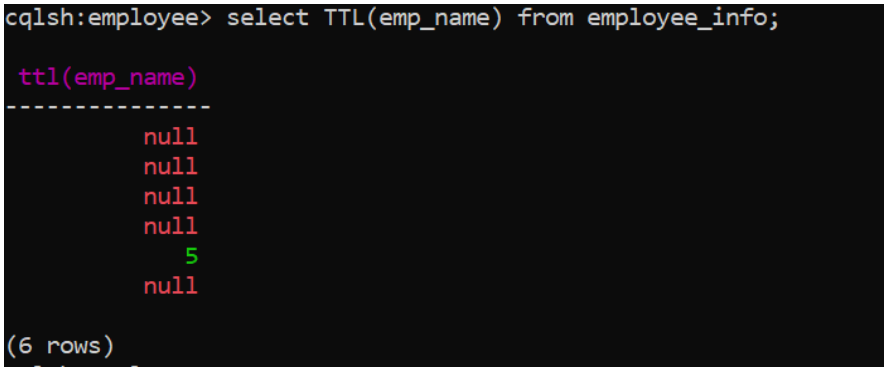
```
cqlsh:employee> begin batch
           ... update employee_info set project = 'xyz' where emp_id = 3
           ... update employee_info set project = 'pqr' where emp_id = 5
           ... update employee_info set project = 'pqr' where emp_id = 2
           ... update employee_info set project = 'abc' where emp_id = 1
           ... update employee_info set project = 'abc' where emp_id = 4
           ... apply batch;
cqlsh:employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | designation       | emp_name | project | salary
--------+---------------------------------+-----------+-------------------+----------+---------+--------
     5 | 2021-02-10 18:30:00.000000+0000 |       HR |       HR employee |      Ram |     pqr |  25000
     1 | 2020-02-29 18:30:00.000000+0000 | Technical |    Technical head |     Arun |     abc |  50000
     2 | 2020-06-10 18:30:00.000000+0000 |       HR |       HR manager |     Ajay |     pqr |  60000
     4 | 2021-05-10 18:30:00.000000+0000 | Technical | Software Engineer |    Kshma |     abc |  35000
     3 | 2022-01-10 18:30:00.000000+0000 |     Sales |          Editor |      Raj |     xyz |  22000

(5 rows)
```



8 Create a TTL of 15 seconds to display the values of Employee

```
cqlsh:employee> insert into employee_info(emp_id,
date_of_joining,dept_name,designation,emp_name,project,salary)  values(6, '2021-02-28','HR','HR
employee','Anvi','xyz',20000) using TTL 15;
cqlsh:employee> select TTL(emp_name) from employee_info;
```

```
 ttl(emp_name)
---------------
          null
          null
          null
          null
             5
          null

(6 rows)
```



LAB PROGRAM 3: Library database using Cassandra

1 Create a key space by name Library

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

```
cqlsh> describe keyspace library;

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

```
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,
        ... 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 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';
```

3. Insert the values into the table in batch

```
cqlsh:library> update library_info set counter_value=counter_value+1 where stud_id=1 and stud_name = 'Raj' and book_name='BDA' and book_id=200 and date_of_issue='2022-04-30';
```

cqlsh:library> update library_info set counter_value=counter_value+1 where stud_id=2 and stud_name = 'Ravi' and book_name='ADA' and book_id=100 and date_of_issue='2022-04-30';
cqlsh:library> update library_info set counter_value=counter_value+1 where stud_id=1 and stud_name = 'Raj' and book_name='BDA' and book_id=200 and date_of_issue='2022-05-30';
cqlsh:library> select * from library_info;

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

 stud_id | stud_name | book_name | book_id | date_of_issue                     | counter_value
---------+-----------+-----------+---------+-----------------------------------+---------------
       1 |       Raj |       BDA |     200 | 2022-04-29 18:30:00.000000+0000 |             1
       1 |       Raj |       BDA |     200 | 2022-05-29 18:30:00.000000+0000 |             1
       2 |      Ravi |       ADA |     100 | 2022-04-29 18:30:00.000000+0000 |             1

(3 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=1 and stud_name = 'Raj' and book_name='BDA' and book_id=200 and date_of_issue='2022-04-30';
cqlsh:library> select * from library_info;

```
 stud_id | stud_name | book_name | book_id | date_of_issue                     | counter_value
---------+-----------+-----------+---------+-----------------------------------+---------------
       1 |       Raj |       BDA |     200 | 2022-04-29 18:30:00.000000+0000 |             2
       1 |       Raj |       BDA |     200 | 2022-05-29 18:30:00.000000+0000 |             1
       2 |      Ravi |       ADA |     100 | 2022-04-29 18:30:00.000000+0000 |             1
```

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

 stud_id | stud_name | book_name | book_id | date_of_issue                     | counter_value
---------+-----------+-----------+---------+-----------------------------------+---------------
       1 |       Raj |       BDA |     200 | 2022-04-29 18:30:00.000000+0000 |             2
       1 |       Raj |       BDA |     200 | 2022-05-29 18:30:00.000000+0000 |             1
       2 |      Ravi |       ADA |     100 | 2022-04-29 18:30:00.000000+0000 |             1

(3 rows)
```

5. Write a query to show that a student with id 1 has taken a book “BDA” 2 times.
cqlsh:library> select counter_value from library_info where stud_id = 1;

```
 counter_value
---------------
             2
             1
```

```
cqlsh:library> select counter_value from library_info where stud_id = 1;

 counter_value
---------------
             2
             1

(2 rows)
```

6. Export the created column to a csv file

```
cqlsh:lab2_library> copy library_info(stud_id,stud_name,book_id,date_of_issue,counter_value)to 'lib.csv';
Using 7 child processes

Starting copy of lab2_library.library_info with columns [stud_id, stud_name, book_id, date_of_issue, counter_v
alue].
Processed: 2 rows; Rate:       9 rows/s; Avg. rate:       9 rows/s
2 rows exported to 1 files in 0.250 seconds.
```

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

cqlsh:library>truncate library_info;

cqlsh:library>copy library_info(stud_id,stud_name,book_id,date_of_issue,counter_value) from ‘lib.csv’;

LAB PROGRAM 4: Execution of HDFS Commands for interaction with Hadoop Environment.

```
bmsce@bmsce-Precision-T1700:~$ sudo su hduser
[sudo] password for bmsce:
hduser@bmsce-Precision-T1700:/home/bmsce$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: namenode running as process 6691. Stop it first.
hduser@localhost's password:
localhost: datanode running as process 6951. Stop it first.
Starting secondary namenodes [0.0.0.0]
hduser@0.0.0.0's password:
0.0.0.0: secondarynamenode running as process 7329. Stop it first.
starting yarn daemons
resourcemanager running as process 7490. Stop it first.
hduser@localhost's password:
localhost: nodemanager running as process 8817. Stop it first.
hduser@bmsce-Precision-T1700:/home/bmsce$ jps
7329 SecondaryNameNode
8817 NodeManager
7490 ResourceManager
6691 NameNode
6951 DataNode
10188 Jps

hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir prathiksha
hduser@bmsce-Precision-T1700:/home/bmsce$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x   - hduser supergroup          0 2022-05-31 09:42 /prathiksha
drwxrwxr-x   - hduser supergroup          0 2019-08-01 16:19 /tmp
drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user

hduser@bmsce-Precision-T1700:/$ cd ~/Desktop
hduser@bmsce-Precision-T1700:~/Desktop$ vi abc.txt
hduser@bmsce-Precision-T1700:~/Desktop$ cd ..
hduser@bmsce-Precision-T1700:~$ hdfs dfs -put ~/Desktop/abc.txt /prathiksha/first.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /prathiksha
Found 1 items
-rw-r--r--   1 hduser supergroup         13 2022-05-31 10:01 /prathiksha/first.txt


hduser@bmsce-Precision-T1700:~$ vi ~/Desktop/welcome.txt
```

```
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal ~/Desktop/welcome.txt /prathiksha/welcome.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /prathiksha
Found 2 items
-rw-r--r--   1 hduser supergroup         13 2022-05-31 10:01 /prathiksha/first.txt
-rw-r--r--   1 hduser supergroup         24 2022-05-31 10:06 /prathiksha/welcome.txt


duser@bmsce-Precision-T1700:~$ hdfs dfs -get /prathiksha/welcome.txt ~/Downloads/first.txt
hduser@bmsce-Precision-T1700:~$ cat ~/Downloads/first.txt
hi hello how you doing?

hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /prathiksha/first.txt ~/Downloads/123.txt
hduser@bmsce-Precision-T1700:~$ cat ~/Downloads/123.txt
abc def ghi
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /prathiksha/first.txt
abc def ghi

hduser@bmsce-Precision-T1700:~$ hdfs dfs -mkdir /ABC
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cp /prathiksha /ABC
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /ABC
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2022-05-31 10:16 /ABC/prathiksha
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cp /prathiksha /DEF
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /DEF
Found 2 items
-rw-r--r--   1 hduser supergroup         13 2022-05-31 10:17 /DEF/first.txt
-rw-r--r--   1 hduser supergroup         24 2022-05-31 10:17 /DEF/welcome.txt

hduser@bmsce-Precision-T1700:~$ hdfs dfs -mv /prathiksha /GHI
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 5 items
drwxr-xr-x   - hduser supergroup          0 2022-05-31 10:16 /ABC
drwxr-xr-x   - hduser supergroup          0 2022-05-31 10:17 /DEF
drwxr-xr-x   - hduser supergroup          0 2022-05-31 10:06 /GHI
drwxrwxr-x   - hduser supergroup          0 2019-08-01 16:19 /tmp
drwxr-xr-x   - hduser supergroup          0 2019-08-01 16:03 /user
```