B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

Lab Record

Big-Data Analytics

Submitted in partial fulfillment for the 6th Semester Laboratory

Bachelor of Technology
in
Computer Science and Engineering

Submitted by:

P Sai Deekshith
1BM18S148

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
Mar-June 2021

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Big-Data Analytics (20CS6PEBDA) laboratory has been carried out by P Sai Deekshith (1BM18CS148) during the 6th Semester Mar-June-2021.

Signature of the Faculty Incharge:

Soumya V
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

Sl No.	Programs
1	Perform the following DB operations using Cassandra Employee.
2	Perform the following DB operations using Cassandra Library.
3	MongoDB - CRUD Demonstration.

BDA lab – 1

Date - 22/03/2021
Experiment-1
Questions:

Perform the following DB operations using Cassandra.

1. Command to create Keyspace.
2. Command to describe the existing Keyspaces.
3. Command to get more details on existing keyspaces such as keyspace name, durable writes, strategy class, strategy options etc.
4. Command to use keyspace students.
5. Command to create a column family or table by the name "Student_Info".
6. Command to look up the names of all tables in current keyspace, or in all keyspaces if there is no current space:
7. Command to describe the table Student_info.
8. To insert data into the column family "student_info".
9. To view the data from the table "Student_Info".
10. To update the value held in the "StudName" column of the "student_info" column family to "David Sheen" for the record where the RollNo column has value = 2.
11. To view only those records where the RollNo column either has a value 1 or 2 or 3.
12. To create an index on the "studname" column of the "student_info" column family use the following statement.
13. To execute the query using the index defined on "studname" column. 14. To create index on the "LastExamPercent" column of the "Student_Info" column family.
15. To specify the number of rows returned in the output using limit. 16. To use column alias for the column 'StudName' in the Student_Info table.
17. To update more than one column of a row of Cassandra table.
18. To delete the column "LastExamPercent" from the "student_info" table for the record where the RollNo = 2.

19. To delete a row from the table Student_Info.

<u>Commands:</u>

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

cqlsh> DESCRIBE KEYSPACES;

students   system_auth system_distributed stocks

system_schema system   system_traces cqlsh>

SELECT * FROM system_schema.keyspaces;

keyspace_name | durable_writes | replication
----------------------+----------------+--
     system_auth | True | {'class': 'org.apache.cassandra.locator.SimpleStrategy',
'replication_factor': '1'}
     system_schema | True | {'class':
'org.apache.cassandra.locator.LocalStrategy'} stocks | True | {'class':
        'org.apache.cassandra.locator.SimpleStrategy',
'replication_factor': '1'} system_distributed | True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor':
'3'}
          system | True | {'class':
'org.apache.cassandra.locator.LocalStrategy'} system_traces | True | {'class':
    'org.apache.cassandra.locator.SimpleStrategy',
'replication_factor': '2'} students | True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}

(7 rows) cqlsh> USE students;
cqlsh:students> create table
Student_Info(
        ... RollNo int PRIMARY KEY,
        ... StudName text,
        ... DateofJoining timestamp,
        ... LastExamPercent double);

```
cqlsh:students> DESCRIBE TABLES; student_info

cqlsh:students> DESCRIBE TABLE student_info;

CREATE TABLE students.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    lastexampercent double, studname text
) WITH 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:students> BEGIN BATCH
        ... INSERT INTO
        ... student_info(RollNo,StudName,DateofJoining,LastExamPercent)
        ... VALUES (1,'Michael Storm','2012-03-29', 69.6)
        ... INSERT INTO student_info
        ... (RollNo,StudName,DateofJoining,LastExamPercent) VALUES
        ... (2,'Stephen Fox','2013-02-27', 72.5)
        ... APPLY BATCH;
cqlsh:students> SELECT * FROM student_info;
```

```
rollno | dateofjoining              | lastexampercent | studname
--------+----------------------------+-----------------+----------------
      1 | 2012-03-28 18:30:00.000000+0000 |           69.6 | Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |           72.5 | Stephen Fox
```

(2 rows) cqlsh:students> UPDATE student_info SET StudName='David Sheen' WHERE RollNo=2; cqlsh:students> SELECT * FROM student_info;

```
rollno | dateofjoining              | lastexampercent | studname
--------+----------------------------+-----------------+----------------
      1 | 2012-03-28 18:30:00.000000+0000 |           69.6 | Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |           72.5 | David Sheen
```

(2 rows) cqlsh:students> select * from student_info where rollno in (1,2,3);

```
rollno | dateofjoining              | lastexampercent | studname
--------+----------------------------+-----------------+----------------
      1 | 2012-03-28 18:30:00.000000+0000 |           69.6 | Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |           72.5 | David Sheen
```

(2 rows)

cqlsh:students> create index on student_info(studname); cqlsh:students> select * from student_info where studname='Michael Storm';

```
rollno | dateofjoining              | lastexampercent | studname
--------+----------------------------+-----------------+----------------
      1 | 2012-03-28 18:30:00.000000+0000 |           69.6 | Michael Storm
```

(1 rows)
cqlsh:students> create index on student_info(lastexampercent); cqlsh:students> select rollno,lastexampercent from student_info limit 1;

```
rollno | lastexampercent
--------+-----------------
      1 |            69.6
```

(1   rows) cqlsh:students> select rollno,studname as "name"from

student_info;

rollno | name
----------+------------------
    1 | Michael Storm
    2 | David Sheen

(2   rows) cqlsh:students> update student_info set
studname='samaira',lastexampercent=85.7 where rollno=2; cqlsh:students> Select
* from student_info;

rollno | dateofjoining | lastexampercent | studname
----------+----------------------------+-----------------+--------------
    1 | 2012-03-28 18:30:00.000000+0000 | 69.6 | Michael Storm
    2 | 2013-02-26 18:30:00.000000+0000 | 85.7 | samaira

(2 rows)

cqlsh:students> begin batch insert into
student_info(rollno,studname,dateofjoining,lastexampercent) values(3,'Mahira
Raj','2012-04-22',76.7) insert into student_info
(rollno,studname,dateofjoining,lastexampercent)
values(4,'Prita Malhotra','2013-03-21',82.5) apply batch;
cqlsh:students> select * from student_info;

rollno | dateofjoining | lastexampercent | studname
----------+----------------------------+-----------------+--------------
    1 | 2012-03-28 18:30:00.000000+0000 | 69.6 | Michael Storm
    2 | 2013-02-26 18:30:00.000000+0000 | 85.7 | samaira
    4 | 2013-03-20 18:30:00.000000+0000 | 82.5 | Prita Malhotra
    3 | 2012-04-21 18:30:00.000000+0000 | 76.7 | Mahira Raj

(4 rows)

cqlsh:students> update student_info set studname='Samaira Arora' where rollno=2;
cqlsh:students> select * from student_info;

```
rollno | dateofjoining              | lastexampercent | studname
-------+----------------------------+-----------------+---------------
     1 | 2012-03-28 18:30:00.000000+0000 |            69.6 | Michael Storm
     2 | 2013-02-26 18:30:00.000000+0000 |            85.7 | Samaira Arora
     4 | 2013-03-20 18:30:00.000000+0000 |            82.5 | Prita Malhotra
     3 | 2012-04-21 18:30:00.000000+0000 |            76.7 |     Mahira Raj

(4 rows)
```

cqlsh:students> delete lastexampercent from student_info where rollno=2;
cqlsh:students> select * from student_info;

```
rollno | dateofjoining              | lastexampercent | studname
-------+----------------------------+-----------------+---------------
     1 | 2012-03-28 18:30:00.000000+0000 |            69.6 | Michael Storm
     2 | 2013-02-26 18:30:00.000000+0000 |            null | Samaira Arora
     4 | 2013-03-20 18:30:00.000000+0000 |            82.5 | Prita Malhotra
     3 | 2012-04-21 18:30:00.000000+0000 |            76.7 |     Mahira Raj

(4 rows)
```

cqlsh:students> delete from student_info where rollno=2;
cqlsh:students> select * from student_info;

```
rollno | dateofjoining              | lastexampercent | studname
-------+----------------------------+-----------------+---------------
     1 | 2012-03-28 18:30:00.000000+0000 |            69.6 | Michael Storm
     4 | 2013-03-20 18:30:00.000000+0000 |            82.5 | Prita Malhotra
     3 | 2012-04-21 18:30:00.000000+0000 |            76.7 |     Mahira Raj

(3 rows)
```
Screenshots:

```
cqlsh> CREATE KEYSPACE students WITH REPLICATION={
   ...    ... 'class':'SimpleStrategy',
   ...    ... 'replication_factor':1};
SyntaxException: line 2:3 mismatched input '.' expecting '}' (...KEYSPACE students WITH REPLICATION={   [.]...)
cqlsh> CREATE KEYSPACE students WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
cqlsh>  DESCRIBE KEYSPACES;

students        system_auth   system_distributed   stocks
system_schema   system        system_traces

cqlsh> SELECT * FROM system_schema.keyspaces;

 keyspace_name     | durable_writes | replication
-------------------+----------------+------------------------------------------------------------------------------------
        system_auth |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
      system_schema |           True |                      {'class': 'org.apache.cassandra.locator.LocalStrategy'}
             stocks |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
  system_distributed |          True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
             system |           True |                      {'class': 'org.apache.cassandra.locator.LocalStrategy'}
      system_traces |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}
           students |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}

(7 rows)
cqlsh> USE students;
cqlsh:students> create table Student_Info(
          ... RollNo int PRIMARY  KEY,
          ... StudName text,
          ... DateofJoining timestamp,
          ... LastExamPercent double);
cqlsh:students> DESCRIBE TABLES;

student_info
```

```
cqlsh:students> DESCRIBE TABLE student_info;

CREATE TABLE students.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    lastexampercent double,
    studname text
) WITH 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:students> BEGIN BATCH
          ... INSERT INTO
          ... student_info(RollNo,StudName,DateofJoining,LastExamPercent)
          ... VALUES (1,'Michael Storm','2012-03-29', 69.6)
          ... INSERT INTO student_info
          ... (RollNo,StudName,DateofJoining,LastExamPercent) VALUES
          ... (2,'Stephen Fox','2013-02-27', 72.5)
          ... APPLY BATCH;
cqlsh:students> SELECT * FROM student_info;

 rollno | dateofjoining                   | lastexampercent | studname
--------+---------------------------------+-----------------+---------------
      1 | 2012-03-28 18:30:00.000000+0000 |            69.6 | Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |            72.5 |   Stephen Fox

(2 rows)
cqlsh:students> UPDATE student_info SET  StudName='David Sheen' WHERE RollNo=2;
cqlsh:students> SELECT * FROM student_info;

 rollno | dateofjoining                   | lastexampercent | studname
--------+---------------------------------+-----------------+---------------
      1 | 2012-03-28 18:30:00.000000+0000 |            69.6 | Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |            72.5 |   David Sheen

(2 rows)
```

```
cqlsh:students> delete lastexampercent from student_info where rollno=2;
cqlsh:students> select * from student_info;

 rollno | dateofjoining                   | lastexampercent | studname
--------+---------------------------------+-----------------+----------------
      1 | 2012-03-28 18:30:00.000000+0000 |            69.6 |   Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |            null |   Samaira Arora
      4 | 2013-03-20 18:30:00.000000+0000 |            82.5 |  Prita Malhotra
      3 | 2012-04-21 18:30:00.000000+0000 |            76.7 |      Mahira Raj

(4 rows)
cqlsh:students> delete from student_info where rollno=2;
cqlsh:students> select * from student_info;

 rollno | dateofjoining                   | lastexampercent | studname
--------+---------------------------------+-----------------+----------------
      1 | 2012-03-28 18:30:00.000000+0000 |            69.6 |   Michael Storm
      4 | 2013-03-20 18:30:00.000000+0000 |            82.5 |  Prita Malhotra
      3 | 2012-04-21 18:30:00.000000+0000 |            76.7 |      Mahira Raj

(3 rows)
cqlsh:students>
```

Experiment-2
Questions:
Perform the following DB operations using **Cassandra**.
1. To create Project Table in students keyspace.
2. To Insert data.
3. To view all rows of project_details table.
4. To view row/record from the "project_details" table wherein the project_id=1
5. To sort order the rows/records of the "project_details" in descending order of project_name.

Commands:

cqlsh:students> create table project_details(
    ... project_id int,
    ... project_name text,
    ... stud_name text,
    ... rating double,
    ... duration int,
    ... primary key(project_id,project_name));

cqlsh:students> begin batch
    ... insert into project_details(project_id,project_name,stud_name,rating,duration)
    ... values(1,'MS data migration','Mahira Raj',3.5,720)
    ... apply batch; cqlsh:students> select
* from project_details;

project_id	project_name	duration	rating	stud_name
1	MS data migration	720	3.5	Mahira Raj

(1 rows) cqlsh:students> begin batch insert into project_details(project_id,project_name,stud_name,rating,duration) values(1,'MS data warehouse','Mahira Raj',3.9,1440) apply batch;

cqlsh:students> begin batch insert into project_details(project_id,project_name,stud_name,rating,duration) values(2,'Sap reporting','Michael Storm',4.2,3000) apply batch;
cqlsh:students> begin batch insert into

project_details(project_id,project_name,stud_name,rating,duration) values(2,'Sap BI DW','Michael Storm',4,9000) apply batch; cqlsh:students> select * from project_details;

```
project_id | project_name        | duration | rating | stud_name
-----------+---------------------+----------+--------+------------
         1 | MS data migration   |      720 |    3.5 |   Mahira Raj
         1 | MS data warehouse   | 1440     |        |    3.9 |    Mahira Raj
         2 |        Sap BI DW    |     9000 | 4 | Michael Storm
         2 |     Sap reporting   |     3000 |    4.2 | Michael Storm
```

(4 rows) cqlsh:students> select * from project_details where project_id=1;

```
project_id | project_name        | duration | rating | stud_name
-----------+---------------------+----------+--------+------------
         1 | MS data migration   |      720 |    3.5 | Mahira Raj
         1 | MS data warehouse   |     1440 |    3.9 | Mahira Raj
```

(2 rows)

cqlsh:students> begin batch insert into project_details(project_id,project_name,stud_name,rating,duration) values(1,'MS data warehouse','Shrushti Das',3.1,3000) apply batch; cqlsh:students> paging off Disabled Query paging. cqlsh:students> select * from project_details where project_id in (1,2) order by project_name desc;

```
project_id | project_name        | duration | rating | stud_name
-----------+---------------------+----------+--------+------------
         2 |     Sap reporting   |     3000 |    4.2 | Michael Storm 2
           |        Sap BI DW    |     9000 |      4 | Michael Storm
         1 | MS data warehouse   |     3000 |    3.1 | Shrushti Das
         1 | MS data migration   |      720 |    3.5 |   Mahira Raj
```

(4 rows)
Screenshots:

```
cqlsh:students> create table project_details(
           ... project_id int,
           ... project_name text,
           ... stud_name text,
           ... rating double,
           ... duration int,
           ... primary key(project_id,project_name));
cqlsh:students> begin batch
           ... insert into project_details(project_id,project_name,stud_name,rating,duration)
           ... values(1,'MS data migration','Mahira Raj',3.5,720)
           ... apply batch;
cqlsh:students> select * from project_details;

 project_id | project_name      | duration | rating | stud_name
------------+-------------------+----------+--------+------------
          1 | MS data migration |      720 |    3.5 | Mahira Raj

(1 rows)
cqlsh:students> begin batch insert into project_details(project_id,project_name,stud_name,rating,duration) values(1,'MS data warehouse','Mahira Raj',3.9,1440) apply batch;
cqlsh:students> begin batch insert into project_details(project_id,project_name,stud_name,rating,duration) values(2,'Sap reporting','Michael Storm',4.2,3000) apply batch;
cqlsh:students> begin batch insert into project_details(project_id,project_name,stud_name,rating,duration) values(2,'Sap BI DW','Michael Storm',4,9000) apply batch;
cqlsh:students> select * from project_details;

 project_id | project_name      | duration | rating | stud_name
------------+-------------------+----------+--------+---------------
          1 | MS data migration |      720 |    3.5 |    Mahira Raj
          1 | MS data warehouse |     1440 |    3.9 |    Mahira Raj
          2 |         Sap BI DW |     9000 |      4 | Michael Storm
          2 |     Sap reporting |     3000 |    4.2 | Michael Storm

(4 rows)
cqlsh:students> select * from project_details where project_id=1;

 project_id | project_name      | duration | rating | stud_name
------------+-------------------+----------+--------+------------
          1 | MS data migration |      720 |    3.5 | Mahira Raj
          1 | MS data warehouse |     1440 |    3.9 | Mahira Raj

(2 rows)
```

```
cqlsh:students> begin batch insert into project_details(project_id,project_name,stud_name,rating,duration) values(1,'MS data warehouse','Shrushti Das',3.1,3000) apply batch;
cqlsh:students> paging off
Disabled Query paging.
cqlsh:students> select * from project_details where project_id in (1,2) order by project_name desc;

 project_id | project_name      | duration | rating | stud_name
------------+-------------------+----------+--------+---------------
          2 |     Sap reporting |     3000 |    4.2 | Michael Storm
          2 |         Sap BI DW |     9000 |      4 | Michael Storm
          1 | MS data warehouse |     3000 |    3.1 |   Shrushti Das
          1 | MS data migration |      720 |    3.5 |    Mahira Raj

(4 rows)
cqlsh:students>
```

BDA lab -2
Date - 29/03/2020
Experiment-1

<u>Questions:</u>
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.

<u>Commands:</u>

cqlsh> create keyspace "Employee" with replication={

   .. 'class':'SimpleStrategy','replication_factor':1};
cqlsh> describe keyspaces

students   system_auth system_distributed system_traces
system_schema system   "Employee"   stocks

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 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: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	2020-05-24 18:30:00.000000+0000	Sales	Manager	John	65000
122	2020-04-15 18:30:00.000000+0000	IT	Software Tester	Jane	70000
121	2021-03-15 18:30:00.000000+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	2020-05-24 18:30:00.000000+0000	Sales	Manager	John	65000
122	2020-04-15 18:30:00.000000+0000	IT	Software Tester	Jane	70000
121	2021-03-15 18:30:00.000000+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 | 2020-05-24 18:30:00.000000+0000 | Sales | Manager | John | null | 65000
   122 | 2020-04-15 18:30:00.000000+0000 | IT | Software Tester | Jane | null | 70000
   121 | 2021-03-15 18:30:00.000000+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 | 2020-05-24 18:30:00.000000+0000 | Sales | Manager | John | {'ad management sys', 'sales improvement proj'} | 65000
   122 | 2020-04-15 18:30:00.000000+0000 | IT | Software Tester | Jane | {'company website testing'} | 70000
   121 | 2021-03-15 18:30:00.000000+0000 | Software | Software Developer | Rosy | {'Employee management app', 'company website'} | 80000 (3

rows)

cqlsh:Employee> INSERT INTO employee_info(Emp_Id,Emp_Name,Designation,Date_Of_joining,Salary,Dept_Name,proje cts) VALUES(124,'Joe','Intern','2021-03-20',25000,'IT',{'LMS'}) USING TTL 15;
cqlsh:Employee> select ttl(emp_name) from employee_info where ... emp_id=124 and salary=25000 allow filtering;

ttl(emp_name)

        7

(1 rows) cqlsh:Employee> select * from employee_info;

emp_id | date_of_joining | dept_name | designation | emp_name | projects | salary

---------+-------------------------------+-----------+--------------------+----------+--+----------

   123 | 2020-05-24 18:30:00.000000+0000 | Sales | Manager | John | {'ad management sys', 'sales improvement proj'} | 65000
   122 | 2020-04-15 18:30:00.000000+0000 | IT | Software Tester | Jane | {'company website testing'} | 70000
   121 | 2021-03-15 18:30:00.000000+0000 | Software | Software Developer | Rosy | {'Employee management app', 'company website'} | 80000

(3 rows)
cqlsh:Employee>

Screenshots:

```
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 | 2020-05-24 18:30:00.000000+0000 |     Sales |            Manager |     John |     null |  65000
    122 | 2020-04-15 18:30:00.000000+0000 |        IT |    Software Tester |     Jane |     null |  70000
    121 | 2021-03-15 18:30:00.000000+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 | 2020-05-24 18:30:00.000000+0000 |     Sales |            Manager |     John | {'ad management sys', 'sales improvement proj'} |  65000
    122 | 2020-04-15 18:30:00.000000+0000 |        IT |    Software Tester |     Jane |                     {'company website testing'} |  70000
    121 | 2021-03-15 18:30:00.000000+0000 |  Software | Software Developer |     Rosy |  {'Employee management app', 'company website'} |  80000

(3 rows)
```

```
cqlsh:Employee> 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;
cqlsh:Employee> select ttl(emp_name) from employee_info where
           ... emp_id=124 and salary=25000 allow filtering;

 ttl(emp_name)
---------------
             7

(1 rows)
cqlsh:Employee> select * from employee_info;

 emp_id | date_of_joining                 | dept_name | designation        | emp_name | projects                                       | salary
--------+---------------------------------+-----------+--------------------+----------+------------------------------------------------+--------
    123 | 2020-05-24 18:30:00.000000+0000 |     Sales |            Manager |     John | {'ad management sys', 'sales improvement proj'} |  65000
    122 | 2020-04-15 18:30:00.000000+0000 |        IT |    Software Tester |     Jane |                     {'company website testing'} |  70000
    121 | 2021-03-15 18:30:00.000000+0000 |  Software | Software Developer |     Rosy |  {'Employee management app', 'company website'} |  80000

(3 rows)
cqlsh:Employee>
```

Experiment-2
Questions:
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
3. Display the details of the table created and increase the value of the counter 4.
Write a query to show that a student with id 112 has taken a book "BDA" 2 times.
5. Export the created column to a csv file
6. Import a given csv dataset from local file system into Cassandra column family.

Commands:

cqlsh> CREATE KEYSPACE "Library" WITH REPLICATION = {
'class':'SimpleStrategy', 'replication_factor':1}; cqlsh> USE
"Library"; cqlsh:Library> DESCRIBE
KEYSPACES;

students   system_auth "Library"   "Employee"   stocks system_schema
system   system_distributed system_traces

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='Anusree' 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='Priya' and Book_name='OOMD' and Book_id=112 and Date_Of_Issue='2021-02-12';
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> select * from library_info;

stud_id | stud_name | book_name | book_id | date_of_issue | counter_value

```
 ........+........-+.......-+.......+.................................+.........-
    1 |  Anusree |      BDA |    111 | 2021-03-14 18:30:00.000000+0000 |                1
    2 |    Priya |     OOMD |    112 | 2021-02-11 18:30:00.000000+0000 |                1
  112 |    Aswin |      BDA |   1123 | 2021-01-17 18:30:00.000000+0000 |                1
```

(3 rows)
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> select * from library_info;

```
stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
........+.........-+.........-+.......+.................................+.........-
    1 |  Anusree |      BDA |    111 | 2021-03-14 18:30:00.000000+0000 |                1
    2 |    Priya |     OOMD |    112 | 2021-02-11 18:30:00.000000+0000 |                1
  112 |    Aswin |      BDA |   1123 | 2021-01-17 18:30:00.000000+0000 |                2
```

(3 rows)
cqlsh:Library> COPY Library_Info(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_value) TO 'd:\libraryInfo.csv';
Using 7 child processes
Starting copy of Library.library_info with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 3 rows; Rate: 1 rows/s; Avg. rate: 1
rows/s 3 rows exported to 1 files in 4.090 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(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_val ue) FROM 'd:\libraryInfo.csv';

Processed: 3 rows; Rate: 0 rows/s; Avg. rate: 1 rows/s
3 rows imported from 1 files in 3.240 seconds (0 skipped).
cqlsh:Library> SELECT * FROM Library_Info_Import;

stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
1	Anusree	BDA	111	2021-03-14 18:30:00.000000+0000	1
2	Priya	OOMD	112	2021-02-11 18:30:00.000000+0000	1
112	Aswin	BDA	1123	2021-01-17 18:30:00.000000+0000	2

(3 rows)

cqlsh:Library>

Screenshots:


```
cqlsh> CREATE KEYSPACE "Library" WITH REPLICATION = { 'class':'SimpleStrategy', 'replication_factor':1};
cqlsh> USE "Library";
cqlsh:Library> DESCRIBE KEYSPACES;

students         system_auth  "Library"            "Employee"      stocks
system_schema    system       system_distributed   system_traces
```



```
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='Anusree' 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='Priya' and Book_name='OOMD' and Book_id=112 and Date_Of_Issue='2021-02-12';
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> select * from library_info
           ... ;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
       1 |   Anusree |       BDA |     111 | 2021-03-14 18:30:00.000000+0000 |             1
       2 |     Priya |      OOMD |     112 | 2021-02-11 18:30:00.000000+0000 |             1
     112 |     Aswin |       BDA |    1123 | 2021-01-17 18:30:00.000000+0000 |             1

(3 rows)
```



```
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> select * from library_info;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
       1 |   Anusree |       BDA |     111 | 2021-03-14 18:30:00.000000+0000 |             1
       2 |     Priya |      OOMD |     112 | 2021-02-11 18:30:00.000000+0000 |             1
     112 |     Aswin |       BDA |    1123 | 2021-01-17 18:30:00.000000+0000 |             2

(3 rows)
cqlsh:Library> COPY Library_Info(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_value) TO 'd:\libraryInfo.csv';
Using 7 child processes

Starting copy of Library.library_info with columns [stud_id, stud_name, book_name, book_id, date_of_issue, counter_value].
Processed: 3 rows; Rate:      1 rows/s; Avg. rate:      1 rows/s
3 rows exported to 1 files in 4.090 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(Stud_Id,Stud_Name,Book_Name,Book_Id,Date_Of_Issue,Counter_value) FROM 'd:\libraryInfo.csv ;

Processed: 3 rows; Rate:       0 rows/s; Avg. rate:       1 rows/s
3 rows imported from 1 files in 3.240 seconds (0 skipped).
cqlsh:Library> SELECT * FROM Library_Info_Import;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
       1 |   Anusree |       BDA |     111 | 2021-03-14 18:30:00.000000+0000 |             1
       2 |     Priya |      OOMD |     112 | 2021-02-11 18:30:00.000000+0000 |             1
     112 |     Aswin |       BDA |    1123 | 2021-01-17 18:30:00.000000+0000 |             2

(3 rows)
cqlsh:Library>
```

BDA Lab-3
Date - 05/04/2020
Questions
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 a 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 the local file system into mongodb collection.

Commands:

use studentdb switched to
db studentdb

db.createCollection("student_details")
{ "ok" : 1 }

db.student_details.insert({'name':'abc','rollno':1,'age':19,'contactno':909090909 0,'
email':'abc@lab.com'}) WriteResult({ "nInserted" : 1 })

db.student_details.insert({'name':'mno','rollno':2,'age':20,'contactno':9999900000,' email':'mno@lab.com'})
WriteResult({ "nInserted" : 1 })

db.student_details.insert({'name':'xyz','rollno':3,'age':21,'contactno':9999911111,' e
mail':'xyz@lab.com'}) WriteResult({ "nInserted" : 1 })

db.student_details.find({})
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19,
"contactno" : 9090909090, "email" : "abc@lab.com" }
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20,
"contactno" : 9999900000, "email" : "mno@lab.com" }
{ "_id" : ObjectId("60a88f8fffecf7c8abe76777"), "name" : "xyz", "rollno" : 3, "age" : 21,

"contactno" : 9999911111, "email" : "xyz@lab.com" }

db.student_details.update({'rollno':3},{$set:{'email':'update@lab.com'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

db.student_details.find({'rollno':3})
{ "_id" : ObjectId("60a88f8fffecf7c8abe76777"), "name" : "xyz", "rollno" : 3, "age" : 21,
"contactno" : 9999911111, "email" : "update@lab.com" }

db.student_details.update({'name':'xyz'},{$set:{'name':'pqr'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
db.student_details.find({'name':'pqr'})
{ "_id" : ObjectId("60a88f8fffecf7c8abe76777"), "name" : "pqr", "rollno" : 3, "age" : 21,
"contactno" : 9999911111, "email" : "update@lab.com" }

mongoexport --db studentdb --collection student_details --out
E:\Desktop\sample.json
2021-05-22T10:43:30.687+0530 connected to: mongodb://localhost/ 2021-
05-22T10:43:31.026+0530 exported 3 records

db.getCollection('student_details').drop()
true

mongoimport --db studentdb --collection student_details --type=json --file=
E:\Desktop\sample.json
2021-05-22T10:46:49.898+0530 connected to: mongodb://localhost/
2021-05-22T10:46:50.044+0530 3 document(s) imported successfully. 0 document(s) failed to
import.

db.student_details.find({})
{ "_id" : ObjectId("60a88f8fffecf7c8abe76777"), "name" : "pqr", "rollno" : 3, "age" : 21,
"contactno" : 9999911111, "email" : "update@lab.com" }
{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19,
"contactno" : 9090909090, "email" : "abc@lab.com" }
{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20,
"contactno" : 9999900000, "email" : "mno@lab.com" }
db.student_details.remove({age:{$gt:20}}) WriteResult({ "nRemoved" : 1 })

db.student_details.find({})

{ "_id" : ObjectId("60a88f32ffecf7c8abe76775"), "name" : "abc", "rollno" : 1, "age" : 19, "contactno" : 9090909090, "email" : "abc@lab.com" }

{ "_id" : ObjectId("60a88f7effecf7c8abe76776"), "name" : "mno", "rollno" : 2, "age" : 20, "contactno" : 9999900000, "email" : "mno@lab.com" }

Screenshots: