NAMRATHA V
1BM17CS151
C3

BDA LAB 6

NAMRATHA V
1BM17CS151
C3

Outputs

I. Perform the following DB operations using Cassandra.

1. Create a keyspace by name Employee

```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.8 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh> CREATE KEYSPACE Employee WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES;

system_schema  system   system_distributed  system_traces
system_auth    student  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

```
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE Employee_Info (Emp_Id int PRIMARY KEY, Emp_Name text,Designation text, DateOfJoining timest
amp, Salary double,Dept_Name text);
cqlsh:employee> DESCRIBE TABLES;

employee_info
```

3. Insert the values into the table in batch

```
cqlsh:employee> BEGIN BATCH INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name)
 VALUES(120,'Nam','Manager','2020-08-01',1000000,'Development');INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation
 , DateOfJoining ,Salary ,Dept_Name) VALUES(121,'Amy','SE','2020-10-18',60000,'Development');INSERT INTO Employee_Info(E
mp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name) VALUES(122,'Penny','SDET','2020-01-08',50000,'R&D');IN
SERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name) VALUES(123,'Shelly','Data Ana
lyst','2020-10-18',40000,'R&D');INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_N
ame) VALUES(124,'Leo','Manager','2019-08-18',1000000,'HR');APPLY BATCH;
cqlsh:employee> SELECT * FROM employee_info;

 emp_id | dateofjoining                   | dept_name   | designation  | emp_name | salary
--------+---------------------------------+-------------+--------------+----------+--------
    120 | 2020-07-31 18:30:00.000000+0000 | Development |      Manager |      Nam |  1e+06
    123 | 2020-10-17 18:30:00.000000+0000 |         R&D | Data Analyst |   Shelly |  40000
    122 | 2020-01-07 18:30:00.000000+0000 |         R&D |         SDET |    Penny |  50000
    121 | 2020-10-17 18:30:00.000000+0000 | Development |           SE |      Amy |  60000
    124 | 2019-08-17 18:30:00.000000+0000 |          HR |      Manager |      Leo |  1e+06

(5 rows)
```

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

```
cqlsh:employee> UPDATE Employee_Info SET Emp_Name = 'Raj' , Dept_Name='R&D' WHERE Emp_Id=121;
cqlsh:employee> SELECT * FROM employee_info;

 emp_id | dateofjoining                   | dept_name   | designation  | emp_name | salary
--------+---------------------------------+-------------+--------------+----------+--------
    120 | 2020-07-31 18:30:00.000000+0000 | Development |      Manager |      Nam |  1e+06
    123 | 2020-10-17 18:30:00.000000+0000 |         R&D | Data Analyst |   Shelly |  40000
    122 | 2020-01-07 18:30:00.000000+0000 |         R&D |         SDET |    Penny |  50000
    121 | 2020-10-17 18:30:00.000000+0000 |         R&D |           SE |      Raj |  60000
    124 | 2019-08-17 18:30:00.000000+0000 |          HR |      Manager |      Leo |  1e+06
```

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.

```
cqlsh:employee> ALTER TABLE employee_info ADD Project VARCHAR;
cqlsh:employee> DESCRIBE TABLE employee_info;

CREATE TABLE employee.employee_info (
    emp_id int PRIMARY KEY,
    dateofjoining timestamp,
    dept_name text,
    designation text,
    emp_name text,
    project text,
    salary double
```

6. Update the altered table to add project names.

```
cqlsh:employee> UPDATE employee_info SET project='EDM' WHERE emp_id=120;
cqlsh:employee> UPDATE employee_info SET project='Alexa' WHERE emp_id=121;
cqlsh:employee> UPDATE employee_info SET project='Health Monitoring System' WHERE emp_id=122;
cqlsh:employee> UPDATE employee_info SET project='Prediction App' WHERE emp_id=123;
cqlsh:employee> UPDATE employee_info SET project='Stock Management' WHERE emp_id=120;
cqlsh:employee> SELECT * FROM employee_info;

 emp_id | dateofjoining                   | dept_name   | designation  | emp_name | project                 | salary
--------+---------------------------------+-------------+--------------+----------+-------------------------+--------
    120 | 2020-07-31 18:30:00.000000+0000 | Development |      Manager |      Nam |        Stock Management |  1e+06
    123 | 2020-10-17 18:30:00.000000+0000 |         R&D | Data Analyst |   Shelly |          Prediction App |  40000
    122 | 2020-01-07 18:30:00.000000+0000 |         R&D |         SDET |    Penny | Health Monitoring System |  50000
    121 | 2020-10-17 18:30:00.000000+0000 |         R&D |           SE |      Raj |                   Alexa |  60000
    124 | 2019-08-17 18:30:00.000000+0000 |          HR |      Manager |      Leo |                    null |  1e+06
```

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

```
cqlsh:employee> INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name) VALUES(125,
'Joe','Software Developer','2020-10-01',60000,'Development') USING TTL 15;
cqlsh:employee> SELECT TTL(designation) FROM employee_Info where Emp_id=125;

 ttl(designation)
------------------
               13
```

II. Perform the following DB operations using Cassandra.

1. Create a keyspace by name Library

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

system_schema  system   student             employee
system_auth    library  system_distributed  system_traces
```

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

```
cqlsh> USE Library;
cqlsh:library> CREATE TABLE Library_Info (Stud_id int,Counter_value counter, Stud_Name text,Book_Name text,Book_Id int,D
oi timestamp,PRIMARY KEY(Stud_id,Stud_Name,Book_Name,Book_id,doi));
cqlsh:library> DESCRIBE TABLES;

library_info
```

3. Insert the values into the table in batch

```
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=111 and Stud_Name='Nam' AND Book_Na
me='BDA' and Book_id=121 and Doi='2020-11-05' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=112 and Stud_Name='Amy' AND Book_Na
me='BDA' and Book_id=122 and Doi='2020-10-05' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=113 and Stud_Name='Penny' AND Book_
Name='DSR' and Book_id=131 and Doi='2020-11-05' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=114 and Stud_Name='Shelly' AND Book
_Name='SQM' and Book_id=141 and Doi='2020-11-03' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=115 and Stud_Name='Leo' AND Book_Na
me='DSR' and Book_id=132 and Doi='2020-11-04'
```

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

```
cqlsh:library> SELECT * FROM  Library_Info;

 stud_id | stud_name | book_name | book_id | doi                             | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     114 |    Shelly |       SQM |     141 | 2020-11-02 18:30:00.000000+0000 |             1
     111 |       Nam |       BDA |     121 | 2020-11-04 18:30:00.000000+0000 |             1
     113 |     Penny |       DSR |     131 | 2020-11-04 18:30:00.000000+0000 |             1
     112 |       Amy |       BDA |     122 | 2020-10-04 18:30:00.000000+0000 |             1
     115 |       Leo |       DSR |     132 | 2020-11-03 18:30:00.000000+0000 |             1
```

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

```
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=112 and Stud_Name='Amy' AND Book_Na
me='BDA' and Book_id=122 and Doi='2020-10-05' ;
cqlsh:library> SELECT * FROM  Library_Info;

 stud_id | stud_name | book_name | book_id | doi                             | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     114 |    Shelly |       SQM |     141 | 2020-11-02 18:30:00.000000+0000 |             1
     111 |       Nam |       BDA |     121 | 2020-11-04 18:30:00.000000+0000 |             1
     113 |     Penny |       DSR |     131 | 2020-11-04 18:30:00.000000+0000 |             1
     112 |       Amy |       BDA |     122 | 2020-10-04 18:30:00.000000+0000 |             2
     115 |       Leo |       DSR |     132 | 2020-11-03 18:30:00.000000+0000 |             1
```

5. Export the created column to a csv file

```
cqlsh:library> COPY Library_Info(Stud_id,Counter_value,Stud_Name,Book_Name,Book_id,doi) TO 'C:\Users\lenovo\Desktop\BDA\
LAB\LAB 6\libraryInfo.csv';
Using 3 child processes

Starting copy of library.library_info with columns [stud_id, counter_value, stud_name, book_name, book_id, doi].
Processed: 5 rows; Rate:      20 rows/s; Avg. rate:       1 rows/s
5 rows exported to 1 files in 3.812 seconds.
```

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

```
cqlsh:library> COPY Library_Info(Stud_id,Counter_value,Stud_Name,Book_Name,Book_id,doi) FROM 'C:\Users\lenovo\Desktop\BD
A\LAB\LAB 6\libraryInfo.csv';
Using 3 child processes
```