Big Data Analytics Lab Report

Ankit kesar
1BM18CS150

i. Create a key space with name students.
ii. Create a column family with name student_info.
iii. Insert the values into the table in batch.

```
bnsce@bnsce-Precision-T1700:~$ cd cassandra/apache-cassandra-3.11.0/bin
bnsce@bnsce-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.
cqlsh> DESCRIBE KEYSPACES;

system_schema  system   system_distributed  system_traces
system_auth    student  employee

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

students       system_auth  student             employee
system_schema  system       system_distributed  system_traces

cqlsh> USE students;
cqlsh:students> CREATE TABLE students_info (Roll_No int PRIMARY KEY, StudName text, DateOfJoining timestamp, last_exam_Percent double);
cqlsh:students> DESCRIBE TABLES;

students_info

cqlsh:students> DESCRIBE TABLE students_info;

CREATE TABLE students.students_info (
    roll_no int PRIMARY KEY,
    dateofjoining timestamp,
    last_exam_percent 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 students_info(Roll_No, StudName, DateOfJoining, last_exam_Percent) VALUES(1,'John','2021-03-25',89.98) INSERT INTO students_info(Roll_No, StudN
ame, DateOfJoining, last_exam_Percent) VALUES(2,'JohnDoe','2021-01-25',97.98) INSERT INTO students_info(Roll_No, StudName, DateOfJoining, last_exam_Percent) VALUES(2,'Smith','2021-01-
07',76.98) INSERT INTO students_info(Roll_No, StudName, DateOfJoining, last_exam_Percent) VALUES(4,'Smith','2021-01-06',64.86)APPLY BATCH;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |     John
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       4 | 2021-01-05 18:30:00.000000+0000 |             64.86 |    Smith

(3 rows)
cqlsh:students>
cqlsh:students> INSERT INTO students_info(Roll_No, StudName, DateOfJoining, last_exam_Percent) VALUES(3,'Iyer','2021-01-07',76.98)
           ... ;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |     John
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       4 | 2021-01-05 18:30:00.000000+0000 |             64.86 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 |             76.98 |     Iyer

(4 rows)
cqlsh:students> UPDATE TABLE Students_info SET studname='John Doe' WHERE Roll_No = 4;
SyntaxException: line 1:7 no viable alternative at input 'TABLE' (UPDATE [TABLE]...)
cqlsh:students> UPDATE Students_info SET studname='John Doe' WHERE Roll_No = 4;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |     John
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       4 | 2021-01-05 18:30:00.000000+0000 |             64.86 | John Doe
       3 | 2021-01-06 18:30:00.000000+0000 |             76.98 |     Iyer

(4 rows)
cqlsh:students>
?           CAPTURE      CONSISTENCY  DEBUG        DESCRIBE     GRANT        LIST         REVOKE       SHOW         TRUNCATE     exit
ALTER       CLEAR        COPY         DELETE       DROP         HELP         LOGIN        SELECT       SOURCE       UPDATE       quit
BEGIN       CLS          CREATE       DESC         EXPAND       INSERT       PAGING       SERIAL       TRACING      USE
cqlsh:students> SELECT Roll_No as "USN" from Students_info;
```

Big Data Analytics Lab Report

```
cqlsh:students> SELECT Roll_No as "USN" from Students_info;

 USN
-----
   1
   2
   4
   3

(4 rows)
cqlsh:students> ?

Documented shell commands:
==========================
CAPTURE  CLS          COPY  DESCRIBE  EXPAND  LOGIN   SERIAL  SOURCE   UNICODE
CLEAR    CONSISTENCY  DESC  EXIT      HELP    PAGING  SHOW    TRACING

CQL help topics:
================
AGGREGATES               CREATE_KEYSPACE          DROP_TRIGGER      TEXT
ALTER_KEYSPACE           CREATE_MATERIALIZED_VIEW  DROP_TYPE         TIME
ALTER_MATERIALIZED_VIEW  CREATE_ROLE              DROP_USER         TIMESTAMP
ALTER_TABLE              CREATE_TABLE             FUNCTIONS         TRUNCATE
ALTER_TYPE               CREATE_TRIGGER           GRANT             TYPES
ALTER_USER               CREATE_TYPE              INSERT            UPDATE
APPLY                    CREATE_USER              INSERT_JSON       USE
ASCII                    DATE                     INT               UUID
BATCH                    DELETE                   JSON
BEGIN                    DROP_AGGREGATE           KEYWORDS
BLOB                     DROP_COLUMNFAMILY        LIST_PERMISSIONS
BOOLEAN                  DROP_FUNCTION            LIST_ROLES
COUNTER                  DROP_INDEX               LIST_USERS
CREATE_AGGREGATE         DROP_KEYSPACE            PERMISSIONS
CREATE_COLUMNFAMILY      DROP_MATERIALIZED_VIEW   REVOKE
CREATE_FUNCTION          DROP_ROLE                SELECT
CREATE_INDEX             DROP_TABLE               SELECT_JSON

cqlsh:students> SELECT * FROM students_info WHERE Roll_No in (1,2,4);

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |     John
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       4 | 2021-01-05 18:30:00.000000+0000 |             64.86 | John Doe

(3 rows)
```

```
cqlsh:students> SELECT * FROM students_info WHERE StudName = 'John';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you
want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
cqlsh:students> CREATE INDEX ON Students_info(StudName);
cqlsh:students> SELECT * FROM students_info WHERE StudName = 'John';

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |     John

(1 rows)
cqlsh:students> SELECT * FROM students_info WHERE StudName = 'Smith';

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith

(1 rows)
cqlsh:students> UPDATE students_info SET Studname = 'David' WHERE studname = 'John';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Some partition key parts are missing: roll_no"
cqlsh:students> UPDATE students_info SET Studname = 'David' WHERE Roll_No = 1;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       4 | 2021-01-05 18:30:00.000000+0000 |             64.86 | John Doe
       3 | 2021-01-06 18:30:00.000000+0000 |             76.98 |     Iyer

(4 rows)
cqlsh:students> DELETE FROM Students_info WHERE Roll_No = 4;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 |             76.98 |     Iyer

(3 rows)
cqlsh:students> DELETE last_exam_percent FROM students_info WHERE Roll_no = 3;
```

Big Data Analytics Lab Report

```
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 |             76.98 |     Iyer

(3 rows)
cqlsh:students> DELETE last_exam_percent FROM students_info WHERE Roll_no = 3;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 |              null |     Iyer

(3 rows)
cqlsh:students> UPDATE students_info SET last_exam_percent  = 85.16 WHERE Roll_No = 3;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |             97.98 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 |             85.16 |     Iyer

(3 rows)
cqlsh:students> ALTER TABLE Students_info ADD hobbies set<text>;
cqlsh:students> DESCRIBE TABLE students_info;

CREATE TABLE students.students_info (
    roll_no int PRIMARY KEY,
    dateofjoining timestamp,
    hobbies set<text>,
    last_exam_percent 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
```

```
cqlsh:students> UPDATE students_info SET language+['Hindi,English'] WHERE Roll_no=1;
SyntaxException: line 1:33 no viable alternative at input '+' (UPDATE students_info SET language[+]...)
cqlsh:students> UPDATE students_info SET language=language+['Hindi,English'] WHERE Roll_no=1;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | hobbies                | language          | last_exam_percent | studname
---------+---------------------------------+------------------------+-------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |     {'Chess,Cricket'} | ['Hindi,English'] |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |  {'Football,Reading'} |              null |             97.98 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 | {'Hockey,Travelling'} |              null |             85.16 |     Iyer

(3 rows)
cqlsh:students> UPDATE students_info SET language=language+['Kannada,Marathi'] WHERE Roll_no=2;
cqlsh:students> UPDATE students_info SET language=language+['Kannada,English,French'] WHERE Roll_no=3;
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | hobbies                | language                       | last_exam_percent | studname
---------+---------------------------------+------------------------+--------------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |     {'Chess,Cricket'} |          ['Hindi,English'] |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |  {'Football,Reading'} |        ['Kannada,Marathi'] |             97.98 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 | {'Hockey,Travelling'} | ['Kannada,English,French'] |             85.16 |     Iyer

(3 rows)
cqlsh:students> CREATE TABLE library_book(counter_value counter, book_name varchar,stud_name varchar, PRIMARY KEY(book_name,stud_name));
cqlsh:students> DESCRIBE library_book;

CREATE TABLE students.library_book (
    book_name text,
    stud_name text,
    counter_value counter,
    PRIMARY KEY (book_name, stud_name)
) WITH CLUSTERING ORDER BY (stud_name 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';
```

Big Data Analytics Lab Report

```
cqlsh:students> INSERT INTO library_book(counter_value, book_name, stud_name) VALUES(0, 'BDA','John');
InvalidRequest: Error from server: code=2200 [Invalid query] message="INSERT statements are not allowed on counter tables, use UPDATE instead"
cqlsh:students> UPDATE library_book SET counter_value=counter_value+1 WHERE book_name='BDA' AND stud_name='John';
cqlsh:students> SELECT * FROM library_book;

 book_name | stud_name | counter_value
-----------+-----------+---------------
       BDA |      John |             1

(1 rows)
cqlsh:students> CREATE TABLE userlogin(userid int PRIMARY KEY, password text);
cqlsh:students> DESCRIBE TABLE userlogin;

CREATE TABLE students.userlogin (
    userid int PRIMARY KEY,
    password 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> INSERT INTO userlogin(userid, password) VALUES(1,'infi') USING TTL 30;
cqlsh:students> SELECT * FROM userlogin;

 userid | password
--------+----------
      1 |     infi

(1 rows)
cqlsh:students> SELECT TTL(password) FROM userlogin WHERE userid = 1;

 ttl(password)
---------------

(0 rows)
```

```
cqlsh:students> INSERT INTO userlogin(userid, password) VALUES(1,'infi') USING TTL 30;
cqlsh:students> SELECT * FROM userlogin;

 userid | password
--------+----------
      1 |     infi

(1 rows)
cqlsh:students> SELECT TTL(password) FROM userlogin WHERE userid = 1;

 ttl(password)
---------------

(0 rows)
cqlsh:students> SELECT * FROM userlogin;

 userid | password
--------+----------

(0 rows)
cqlsh:students> SELECT * FROM students_info;

 roll_no | dateofjoining                   | hobbies               | language                    | last_exam_percent | studname
---------+---------------------------------+-----------------------+-----------------------------+-------------------+----------
       1 | 2021-03-24 18:30:00.000000+0000 |     {'Chess,Cricket'} |          ['Hindi,English'] |             89.98 |    David
       2 | 2021-01-24 18:30:00.000000+0000 |  {'Football,Reading'} |        ['Kannada,Marathi'] |             97.98 |    Smith
       3 | 2021-01-06 18:30:00.000000+0000 | {'Hockey,Travelling'} | ['Kannada,English,French'] |             85.16 |     Iyer

(3 rows)
```

Big Data Analytics Lab Report

2. Perform the following DB operations using Cassandra.

i. Create a keyspace by name Employee.

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

iii. Insert the values into the table in batch.

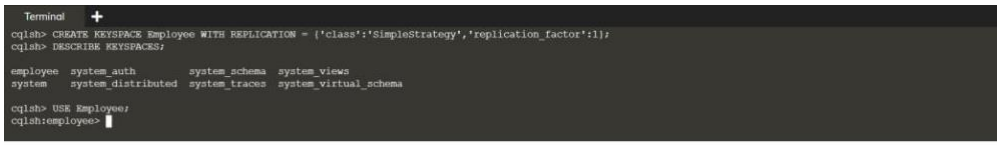
iv. Update Employee name and Department of Emp-Id 121

v. Sort the details of Employee records based on salary

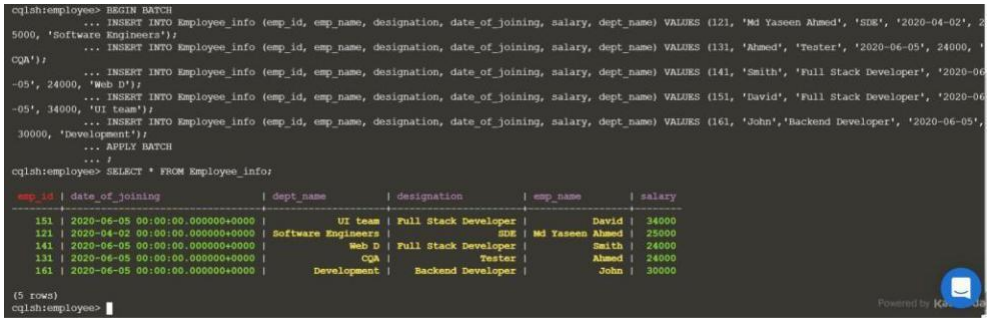
vi. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

vii. Update the altered table to add project names.

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



Big Data Analytics Lab Report



```
cqlsh:employee> BEGIN BATCH
           ... INSERT INTO Employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (121, 'Md Yaseen Ahmed', 'SDE', '2020-04-02', 2
5000, 'Software Engineers');
           ... INSERT INTO Employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (131, 'Ahmed', 'Tester', '2020-06-05', 24000, '
CQA');
           ... INSERT INTO Employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (141, 'Smith', 'Full Stack Developer', '2020-06
-05', 24000, 'Web D');
           ... INSERT INTO Employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (151, 'David', 'Full Stack Developer', '2020-06
-05', 34000, 'UI team');
           ... INSERT INTO Employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (161, 'John','Backend Developer', '2020-06-05',
 30000, 'Development');
           ... APPLY BATCH
           ... ;
cqlsh:employee> SELECT * FROM Employee_info;

 emp_id | date_of_joining                 | dept_name          | designation          | emp_name        | salary
--------+---------------------------------+--------------------+----------------------+-----------------+--------
    151 | 2020-06-05 00:00:00.000000+0000 |            UI team | Full Stack Developer |           David |  34000
    121 | 2020-04-02 00:00:00.000000+0000 | Software Engineers |                  SDE | Md Yaseen Ahmed |  25000
    141 | 2020-06-05 00:00:00.000000+0000 |             Web D | Full Stack Developer |           Smith |  24000
    131 | 2020-06-05 00:00:00.000000+0000 |                CQA |               Tester |           Ahmed |  24000
    161 | 2020-06-05 00:00:00.000000+0000 |        Development |    Backend Developer |            John |  30000

(5 rows)
cqlsh:employee>
```

```
cqlsh:employee> UPDATE Employee_info SET emp_name='Yaseen Ahmed MD', designation='Full Stack Developer' WHERE emp_id=121;
cqlsh:employee> SELECT * FROM Employee_info;

 emp_id | date_of_joining                 | dept_name          | designation          | emp_name        | salary
--------+---------------------------------+--------------------+----------------------+-----------------+--------
    151 | 2020-06-05 00:00:00.000000+0000 |            UI team | Full Stack Developer |           David |  34000
    121 | 2020-04-02 00:00:00.000000+0000 | Software Engineers | Full Stack Developer | Yaseen Ahmed MD |  25000
    141 | 2020-06-05 00:00:00.000000+0000 |             Web D | Full Stack Developer |           Smith |  24000
    131 | 2020-06-05 00:00:00.000000+0000 |                CQA |               Tester |           Ahmed |  24000
    161 | 2020-06-05 00:00:00.000000+0000 |        Development |    Backend Developer |            John |  30000

(5 rows)
cqlsh:employee>
```

```
cqlsh:employee> ALTER TABLE Employee_info ADD projects set<text>;
cqlsh:employee> UPDATE Employee_info set projects = projects + {'Ecommerce','LMS'} WHERE emp_id = 121;
cqlsh:employee> SELECT * FROM Employee_info;

 emp_id | date_of_joining                 | dept_name          | designation          | emp_name        | projects                  | salary
--------+---------------------------------+--------------------+----------------------+-----------------+---------------------------+--------
    151 | 2020-06-05 00:00:00.000000+0000 |            UI team | Full Stack Developer |           David |                      null |  34000
    121 | 2020-04-02 00:00:00.000000+0000 | Software Engineers | Full Stack Developer | Yaseen Ahmed MD | {'Ecommerce', 'LMS'} |  25000
    141 | 2020-06-05 00:00:00.000000+0000 |             Web D | Full Stack Developer |           Smith |                      null |  24000
    131 | 2020-06-05 00:00:00.000000+0000 |                CQA |               Tester |           Ahmed |                      null |  24000
    161 | 2020-06-05 00:00:00.000000+0000 |        Development |    Backend Developer |            John |                      null |  30000

(5 rows)
cqlsh:employee>
```

```
cqlsh:employee> INSERT INTO Employee_info (emp_id, emp_name, designation, date_of_joining, salary, dept_name) VALUES (171, 'Shikhar', 'UI Designer', '2020-07-10',24
500, 'UI team') using TTL 15;
cqlsh:employee> SELECT * FROM Employee_info;

 emp_id | date_of_joining                 | dept_name          | designation          | emp_name        | projects                  | salary
--------+---------------------------------+--------------------+----------------------+-----------------+---------------------------+--------
    151 | 2020-06-05 00:00:00.000000+0000 |            UI team | Full Stack Developer |           David |                      null |  34000
    121 | 2020-04-02 00:00:00.000000+0000 | Software Engineers | Full Stack Developer | Yaseen Ahmed MD | {'Ecommerce', 'LMS'} |  25000
    141 | 2020-06-05 00:00:00.000000+0000 |             Web D | Full Stack Developer |           Smith |                      null |  24000
    131 | 2020-06-05 00:00:00.000000+0000 |                CQA |               Tester |           Ahmed |                      null |  24000
    171 | 2020-07-10 00:00:00.000000+0000 |            UI team |          UI Designer |         Shikhar |                      null |  24500
    161 | 2020-06-05 00:00:00.000000+0000 |        Development |    Backend Developer |            John |                      null |  30000

(6 rows)
cqlsh:employee> SELECT * FROM Employee_info;

 emp_id | date_of_joining                 | dept_name          | designation          | emp_name        | projects                  | salary
--------+---------------------------------+--------------------+----------------------+-----------------+---------------------------+--------
    151 | 2020-06-05 00:00:00.000000+0000 |            UI team | Full Stack Developer |           David |                      null |  34000
    121 | 2020-04-02 00:00:00.000000+0000 | Software Engineers | Full Stack Developer | Yaseen Ahmed MD | {'Ecommerce', 'LMS'} |  25000
    141 | 2020-06-05 00:00:00.000000+0000 |             Web D | Full Stack Developer |           Smith |                      null |  24000
    131 | 2020-06-05 00:00:00.000000+0000 |                CQA |               Tester |           Ahmed |                      null |  24000
    161 | 2020-06-05 00:00:00.000000+0000 |        Development |    Backend Developer |            John |                      null |  30000

(5 rows)
cqlsh:employee>
```

Big Data Analytics Lab Report

3. Perform the following DB operations using Cassandra.

i. Create a keyspace by name Library.

ii. 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.

iii. Insert the values into the table in batch.

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

v. Write a query to show that a student with id 112 has taken a book "BDA" 2 times.

vi. Export the created column to a csv file.

vii. 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> DESCRIBE KEYSPACES;

library  system_auth         system_schema  system_views
system   system_distributed  system_traces  system_virtual_schema

cqlsh> USE Library;
cqlsh: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 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:library>
```

Big Data Analytics Lab Report

```
cqlsh:library> UPDATE Library_info set counter_value = counter_value + 1 WHERE Stud_id = 101 AND Stud_name = 'Md Yaseen' AND Book_name = 'BDA' AND Book_id = 1001 AND date_
of_issue = '2021-01-26';
cqlsh:library> UPDATE Library_info set counter_value = counter_value + 1 WHERE Stud_id = 102 AND Stud_name = 'Arbaz Ahmed' AND Book_name = 'ML' AND Book_id = 1002 AND date
_of_issue = '2021-02-03';
cqlsh:library> UPDATE Library_info set counter_value = counter_value + 1 WHERE Stud_id = 103 AND Stud_name = 'David' AND Book_name = 'OOMD' AND Book_id = 1003 AND date_of_
issue = '2021-03-13';
cqlsh:library> UPDATE Library_info set counter_value = counter_value + 1 WHERE Stud_id = 104 AND Stud_name = 'Smith' AND Book_name = 'CNS' AND Book_id = 1004 AND date_of_i
ssue = '2021-02-13';
cqlsh:library> UPDATE Library_info set counter_value = counter_value + 1 WHERE Stud_id = 105 AND Stud_name = 'John' AND Book_name = 'ME' AND Book_id = 1005 AND date_of_iss
ue = '2021-01-06';
cqlsh:library> SELECT * FROM Library_info;

 stud_id | stud_name   | book_name | book_id | date_of_issue                   | counter_value
---------+-------------+-----------+---------+---------------------------------+---------------
     105 |        John |        ME |    1005 | 2021-01-06 00:00:00.000000+0000 |             1
     104 |       Smith |       CNS |    1004 | 2021-02-13 00:00:00.000000+0000 |             1
     102 | Arbaz Ahmed |        ML |    1002 | 2021-02-03 00:00:00.000000+0000 |             1
     101 |   Md Yaseen |       BDA |    1001 | 2021-01-26 00:00:00.000000+0000 |             1
     103 |       David |      OOMD |    1003 | 2021-03-13 00:00:00.000000+0000 |             1

(5 rows)
```

```
cqlsh:library> UPDATE Library_info SET counter_value = counter_value + 1 WHERE Stud_id = 101 AND Stud_name = 'Md Yaseen' AND Book_name = 'BDA' AND Book_id = 1001 AND date_
of_issue = '2021-01-30';
cqlsh:library> SELECT * FROM Library_info WHERE Stud_id = 101 AND Stud_name = 'Md yaseen' AND Book_name = 'BDA' AND book_id = 1001;
 stud_id | stud_name | book_name | book_id | date_of_issue | counter_value
---------+-----------+-----------+---------+---------------+---------------

(0 rows)
cqlsh:library> SELECT * FROM Library_info WHERE Stud_id = 101;

 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     101 | Md Yaseen |       BDA |    1001 | 2021-01-26 00:00:00.000000+0000 |             1
     101 | Md Yaseen |       BDA |    1001 | 2021-01-30 00:00:00.000000+0000 |             1

(2 rows)
cqlsh:library> UPDATE Library_info SET counter_value = counter_value + 1 WHERE Stud_id = 101 AND Stud_name = 'Md Yaseen' AND Book_name = 'BDA' AND Book_id = 1001 AND date_
of_issue = '2021-01-30';
cqlsh:library> SELECT * FROM Library_info WHERE Stud_id = 101;
 stud_id | stud_name | book_name | book_id | date_of_issue                   | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     101 | Md Yaseen |       BDA |    1001 | 2021-01-26 00:00:00.000000+0000 |             1
     101 | Md Yaseen |       BDA |    1001 | 2021-01-30 00:00:00.000000+0000 |             2

(2 rows)
```

Big Data Analytics Lab Report

1. Create a new collection use Student

2. Insert a value

```
db.Student.insert({ "Name" :
"Akash",
  "RollNo:" : 1,
  "Age" : 21,
  "ContactNo" : "7894561230",
  "EmailId": "akasha@gmail.com"
})
```

3. Insert multiple values at once

```
var MyStudents = [
  {
    "Name" : "Akshay",
    "RollNo:" : 2,
    "Age" : 22,
    "ContactNo" : "8945612370",
    "EmailId": "akshay@gmail.com"
  },
  {
    "Name" : "Anand",
    "RollNo:" : 3,
    "Age" : 21,
    "ContactNo" : "1234567890",
    "EmailId" : "anand@gmail.com"
  },
  {
    "Name" : "Ayesha",
    "RollNo:" : 4,
    "Age" : 20,
    "ContactNo" : "5289631470",
    "EmailId" : "ayesha@gmail.com"
  },
  {
    "Name" : "Vinay",
    "RollNo:" : 5,
    "Age" : 18,
    "ContactNo" : "4561237890",
```

```
      "EmailId" : "vinay@gmail.com"
   },
]

db.Student.insert(MyStudents);
```

4. Print all current values

```
db.getCollection('Student').find({}).forEach(printjson)

{
    "_id" : ObjectId("606ad5a6e581cc0b904470a5"),
    "Name" : "Akash",
    "RollNo:" : 1,
    "Age" : 21,
    "ContactNo" : "7894561230",
    "EmailId": "akasha@gmail.com"
}
{
   "_id" : ObjectId("606ad60fe581cc0b904470a6"),
   "Name" : "Akshay",
     "RollNo:" : 2,
     "Age" : 22,
     "ContactNo" : "8945612370",
     "EmailId": "akshay@gmail.com"
}
{
   "_id" : ObjectId("606ad60fe581cc0b904470a7"),
   "Name" : "Anand",
     "RollNo:" : 3,
     "Age" : 21,
     "ContactNo" : "1234567890",
     "EmailId" : "anand@gmail.com"
}
{
   "_id" : ObjectId("606ad60fe581cc0b904470a8"),
   "Name" : "Ayesha",
     "RollNo:" : 4,
     "Age" : 20,
     "ContactNo" : "5289631470",
     "EmailId" : "ayesha@gmail.com"
}
```

```
{
   "_id" : ObjectId("606ad60fe581cc0b904470a9"),
   "Name" : "Vinay",
     "RollNo:" : 10,
     "Age" : 18,
     "ContactNo" : "4561237890",
     "EmailId" : "vinay@gmail.com"
}
```


5. Update RollNo of a student

```
db.Student.update( {"RollNo:" :
10},
{$set: { "EmailId" : "updated@gmail.com"}});

db.getCollection('Student').find({"RollNo:":10}).forEach(printjson)
```


```
{
   "_id" : ObjectId("606ad60fe581cc0b904470a9"),
   "Name" : "Vinay",
     "RollNo:" : 10,
     "Age" : 18,
     "ContactNo" : "4561237890",
     "EmailId" : "updated@gmail.com"
}
```


6. Update Name of a student

```
db.Student.update( {"Name" :
"Akshay"},
{$set: { "Name" : "Avanthika"}});

db.getCollection('Student').find({"Name" : "Avanthika"}).forEach(printjson)
```


```
{
   "_id" : ObjectId("606ad5a6e581cc0b904470a5"),
   "Name" : "Avanthika",
   "RollNo:" : 2,
     "Age" : 22,
     "ContactNo" : "8945612370",
     "EmailId": "akshay@gmail.com"
}
```

7. Export to json
mongoexport --db testdb --collection Student --
out C:\Users\ Desktop\Student.json

{"_id" : ObjectId("606ad5a6e581cc0b904470a5"),"Name" : "Akash","RollNo:" :
1,"Age" : 21,"ContactNo" : "7894561230","EmailId": "akasha@gmail.com"}

{"_id" : ObjectId("606ad5a6e581cc0b904470a5"),"Name" : "Avanthika","RollNo:" :
2,"Age" : 22,"ContactNo" : "8945612370","EmailId": "akshay@gmail.com"}

{"_id" : ObjectId("606ad60fe581cc0b904470a7"),"Name" : "Anand","RollNo:" :
3,"Age" : 21,"ContactNo" : "1234567890","EmailId" : "anand@gmail.com"}

{"_id" : ObjectId("606ad60fe581cc0b904470a8"),"Name" : "Ayesha","RollNo:" :
4,"Age" : 20,"ContactNo" : "5289631470","EmailId" : "ayesha@gmail.com"}

{"_id" : ObjectId("606ad60fe581cc0b904470a9"),"Name" : "Vinay","RollNo:" :
10,"Age" : 18,"ContactNo" : "4561237890","EmailId" : "updated@gmail.com"}

8. Drop Student
db.getCollection('Student').drop()

9. Import from exported file mongoimport
--db testdb --collection Student
C:\Users \Desktop\Student.json