Working with Cassandra

Create KeySpace :

```
CREATE KEYSPACE Students WITH REPLICATION =
{'class':'SimpleStrategy','replication_factor':1};
```

Describe the existing Keyspaces:

```
DESCRIBE KEYSPACES;
```

For More details on existing keyspaces:

```
SELECT * FROM system.schema_keyspaces;
```

use the keyspace "Students":

```
USE Students;
```

To create table (column family) by name Student_Info:

```
CREATE TABLE Students_Info (Roll_No int PRIMARY KEY, StudName text,
DateOfJoining timestamp, last_exam_Percent double);
```

Lookup the names of all tables in the current keyspaces

```
DESCRIBE TABLES;
```

Describe the table information

```
DESCRIBE TABLE <Table_Name>;
```

CRUD

Insert :

```
BEGIN BATCH
INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining, last_exam_Percent)
VALUES (1,'Asha','2012-03-12',79.9)
INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining, last_exam_Percent)
VALUES (1,'Krian','2012-03-12',89.9)
INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining, last_exam_Percent)
VALUES (1,'Tarun','2012-03-12',78.9)
INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining, last_exam_Percent)
VALUES (1,'Samrth','2012-03-12',90.9)
INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining, last_exam_Percent)
VALUES (1,'Smitha','2012-03-12',67.9)
INSERT INTO Students_Info(Roll_No, StudName, DateOfJoining, last_exam_Percent)
VALUES (1,'Rohan','2012-03-12',56.9)
APPLY BATCH;
```

View data from the table "Students_Info"

```
SELECT * FROM  Students_Info;
```

View data from the table "Students_Info" where RoolNo column either has a value 1 or 2 or 3

```
SELECT * FROM  Students_Info WHERE Roll_No IN (1,2,3);
```

To execute a non primary key - will throw an error

```
select * from students_info where Studname= 'Asha';
```

So create an INDEX on the Column as below:
To create an INDEX on StudName Column of the Students_Info column family

```
CREATE INDEX ON  Students_Info ( StudName);
```

Now execute the query based on the INDEXED Column:

```
select * from students_info where Studname= 'Asha';
```

To specify the number of rows retured in the output

```
select Roll_No, StudName from students_info LIMIT 2;
```

Alias for Column:

```
Select Roll_No as "USN" from Students_info;
```

UPDATE

```
UPDATE students_info SET StudName='David Sheen' WHERE RollNo=2;
```

Lets try to update the primary key

```
UPDATE students_info SET rollno=6 WHERE rollno=3;
```

DELETE

```
DELETE LastExamPercent FROM students_info WHERE RollNo=2;
```

Delete a Row

```
DELETE FROM student_info WHERE RollNo=2;
```

Set Collection

A column of type set consists of unordered unique values. However, when the column is queried, it returns, it returns the values in sorted order. For example, for text values, it sorts in alphabetical order.

```
ALTER TABLE students_info ADD hobbies set<text>
```

List Collection

When the order of elements matter, one should go for a list collection.

```
ALTER TABLE students_info ADD language list<text>;

UPDATE students_info
        SET hobbies=hobbies+{'Chess,Table Tennis'}
                WHERE RollNo=1;
```

Dear all,
Today's lab program to be executed.

```
SELECt * from students_info WHERE RollNo=1;


UPDATE students_info
        SET langusge=language+['Hindi,English']
            WHERE RollNo=1;
```

Note: You can remove an element from a set using the subtraction(-) operator.

USING A COUNTER

A counter is a special column that is changed in increments. For example, we may need a counter column to count the number of times a particular book is issued from the library bythe student.

```
CREATE TABLE library_book(counter_value counter, book_name varchar, stud_name varchar,
PRIMARY KEY(book_name,stud_name));
```

Load data into the counter column

```
UPDATE library_book SET counetr value=couner_vale+1 WHERE book_name='Big data Analytics'
AND stud_name='jeet';
```

TIME TO LIVE

```
CREATE TABLE userlogin(userid int PRIMARY KEY, password text);

INSERT INTO userlogin(userid, password) VALUES (1,'infy') USING TTL 30;

SELECT TTL (password) FROM userlogin WHERE userid=1;
```

IMPORT and EXPORT

Export to CSV

```
COPY elearninglists(id,course_order, course_id,courseowner,title) TO 'd:\elearninglists.csv';
```

Import from CSV

```
COPY elearninglists(id,course_order, course_id,courseowner,title) FROM
'd:\elearninglists.csv';
```

Import FROM STDIN

```
COPY persons(id,fname,lnmae)FROM STDIN;
```

Export to STDOUT

```
COPY elearninglists(id,course_order, course_id,courseowner,title) TO STDOUT;
```

1BM22CS158_BDA_Lab4_cassandra

1. Create Keyspace
2. Use the Keyspace
3. Create Table
4. **Inserting Data into Table**
5. **View Table Data**

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Students WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 1};
AlreadyExists: Keyspace 'students' already exists
cqlsh> CREATE KEYSPACE school_data WITH REPLICATION = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE school_data;
cqlsh:school_data>
cqlsh:school_data> CREATE TABLE student_info (
               ...   Roll_No int PRIMARY KEY,
               ...   StudName text,
               ...   DateOfJoining timestamp,
               ...   last_exam_percent double
               ... );
cqlsh:school_data> INSERT INTO student_info (Roll_No, StudName, DateOfJoining, last_exam_percent)
               ... VALUES (1, 'Asha', '2012-03-12', 79.9);
cqlsh:school_data>
cqlsh:school_data> INSERT INTO student_info (Roll_No, StudName, DateOfJoining, last_exam_percent)
               ... VALUES (2, 'Kiran', '2012-03-12', 89.9);
cqlsh:school_data>
cqlsh:school_data> INSERT INTO student_info (Roll_No, StudName, DateOfJoining, last_exam_percent)
               ... VALUES (3, 'Tarun', '2012-03-12', 78.0);
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2012-03-11 18:30:00.000000+0000 |              79.9 |     Asha
       2 | 2012-03-11 18:30:00.000000+0000 |              89.9 |    Kiran
       3 | 2012-03-11 18:30:00.000000+0000 |                78 |    Tarun

(3 rows)
cqlsh:school_data>
```

6. **Create Index**
7. **Select Data Using Index**

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> USE school_data;
cqlsh:school_data> CREATE INDEX ON student_info (StudName);
cqlsh:school_data> SELECT * FROM student_info WHERE StudName = 'Asha';

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2012-03-11 18:30:00.000000+0000 |              79.9 |     Asha

(1 rows)
cqlsh:school_data> UPDATE student_info SET StudName = 'David Sheen' WHERE Roll_No = 2;
```

8. Update Data
9. Delete Data
10. Add a Set or List Collection
11. Update Collections

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> USE school_data;
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+-------------
       1 | 2012-03-11 18:30:00.000000+0000 |              79.9 |        Asha
       2 | 2012-03-11 18:30:00.000000+0000 |              89.9 | David Sheen
       3 | 2012-03-11 18:30:00.000000+0000 |                78 |       Tarun

(3 rows)
cqlsh:school_data> DELETE FROM student_info WHERE Roll_No = 2;
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | last_exam_percent | studname
---------+---------------------------------+-------------------+----------
       1 | 2012-03-11 18:30:00.000000+0000 |              79.9 |     Asha
       3 | 2012-03-11 18:30:00.000000+0000 |                78 |    Tarun

(2 rows)
cqlsh:school_data> ALTER TABLE student_info ADD hobbies SET<text>;

cqlsh:school_data> ALTER TABLE student_info ADD languages LIST<text>;
cqlsh:school_data>
cqlsh:school_data> UPDATE student_info SET hobbies = hobbies + {'Chess', 'Table Tennis'} WHERE Roll_No = 1;
cqlsh:school_data> UPDATE student_info SET languages = languages + {'English', 'French'} WHERE Roll_No = 1;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid set literal for languages of type list<text>
cqlsh:school_data> UPDATE student_info SET languages = languages || ['English', 'French'] WHERE Roll_No = 1;
Invalid syntax at char 47
  UPDATE student_info SET languages = languages || ['English', 'French'] WHERE Roll_No = 1;
                                               ^
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | hobbies                  | languages | last_exam_percent | studname
---------+---------------------------------+--------------------------+-----------+-------------------+----------
       1 | 2012-03-11 18:30:00.000000+0000 | {'Chess', 'Table Tennis'} |      null |              79.9 |     Asha
       3 | 2012-03-11 18:30:00.000000+0000 |                     null |      null |                78 |    Tarun

(2 rows)
cqlsh:school_data>
```

```
(2 rows)
cqlsh:school_data> ALTER TABLE student_info ADD hobbies SET<text>;

cqlsh:school_data> ALTER TABLE student_info ADD languages LIST<text>;
cqlsh:school_data>
cqlsh:school_data> UPDATE student_info SET hobbies = hobbies + {'Chess', 'Table Tennis'} WHERE Roll_No = 1;
cqlsh:school_data> UPDATE student_info SET languages = languages + {'English', 'French'} WHERE Roll_No = 1;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid set literal for languages of type list<text>
cqlsh:school_data> UPDATE student_info SET languages = languages || ['English', 'French'] WHERE Roll_No = 1;
Invalid syntax at char 47
  UPDATE student_info SET languages = languages || ['English', 'French'] WHERE Roll_No = 1;
                                               ^
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | hobbies                  | languages | last_exam_percent | studname
---------+---------------------------------+--------------------------+-----------+-------------------+----------
       1 | 2012-03-11 18:30:00.000000+0000 | {'Chess', 'Table Tennis'} |      null |              79.9 |     Asha
       3 | 2012-03-11 18:30:00.000000+0000 |                     null |      null |                78 |    Tarun

(2 rows)
cqlsh:school_data> UPDATE student_info SET languages = languages + ['English', 'French'] WHERE Roll_No = 1;
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | hobbies                  | languages             | last_exam_percent | studn
---------+---------------------------------+--------------------------+-----------------------+-------------------+------
       1 | 2012-03-11 18:30:00.000000+0000 | {'Chess', 'Table Tennis'} | ['English', 'French'] |              79.9 |     A
       3 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |                78 |    Ta

(2 rows)
cqlsh:school_data>
```

14. Batch Insert Operations

```
cqlsh:school_data> BEGIN BATCH
               ... INSERT INTO student_info (Roll_No, StudName, DateOfJoining, last_exam_percent) VALUES (4, 'Samrth', '2012-03-12', 90.9);
               ... INSERT INTO student_info (Roll_No, StudName, DateOfJoining, last_exam_percent) VALUES (5, 'Smitha', '2012-03-12', 67.9);
               ... INSERT INTO student_info (Roll_No, StudName, DateOfJoining, last_exam_percent) VALUES (6, 'Rohen', '2012-03-12', 56.9);
               ... APPLY BATCH;
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | hobbies                  | languages             | last_exam_percent | studname
---------+---------------------------------+--------------------------+-----------------------+-------------------+----------
       5 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |              67.9 |   Smitha
       1 | 2012-03-11 18:30:00.000000+0000 | {'Chess', 'Table Tennis'} | ['English', 'French'] |              79.9 |     Asha
       4 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |              90.9 |   Samrth
       6 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |              56.9 |    Rohen
       3 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |                78 |    Tarun

(5 rows)
cqlsh:school_data>
```

13. Describe Keyspaces and Tables

```
cqlsh:school_data> DESCRIBE KEYSPACES;

company     newstudents  students   students9          system_schema
education   school_data  students1  system             system_traces
employee    stud         students3  system_auth        system_views
newstudent  student      students8  system_distributed  system_virtual_schema

cqlsh:school_data> DESCRIBE TABLES;

student_info

cqlsh:school_data> DESCRIBE TABLE student_info;

CREATE TABLE school_data.student_info (
    roll_no int PRIMARY KEY,
    dateofjoining timestamp,
    last_exam_percent double,
    studname text,
    hobbies set<text>,
    languages list<text>
) 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 memtable = 'default'
    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';

CREATE INDEX student_info_studname_idx ON school_data.student_info (studname);
cqlsh:school_data> SELECT * FROM student_info;

 roll_no | dateofjoining                   | hobbies                  | languages             | last_exam_percent | studname
---------+---------------------------------+--------------------------+-----------------------+-------------------+----------
       5 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |              67.9 |   Smitha
       1 | 2012-03-11 18:30:00.000000+0000 | {'Chess', 'Table Tennis'} | ['English', 'French'] |              79.9 |     Asha
       4 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |              90.9 |   Samrth
       6 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |              56.9 |    Rohen
       3 | 2012-03-11 18:30:00.000000+0000 |                     null |                  null |                78 |    Tarun

(5 rows)
cqlsh:school_data>
```

2. Exporting and Importing Data
Export to CSV:

Import from CSV: