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;
```

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;