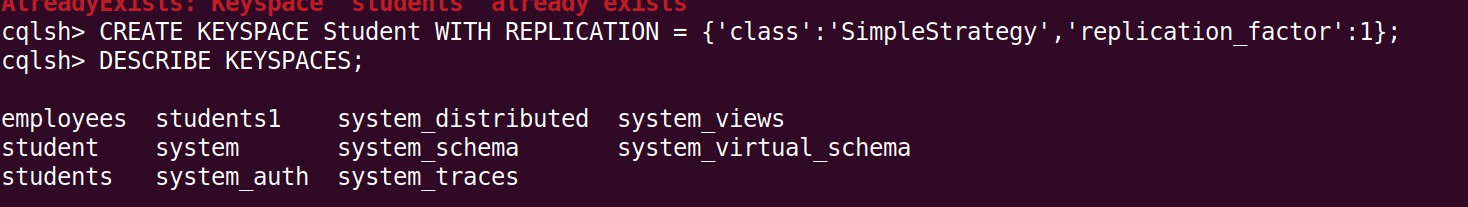
# Working with Cassandra Create KeySpace :

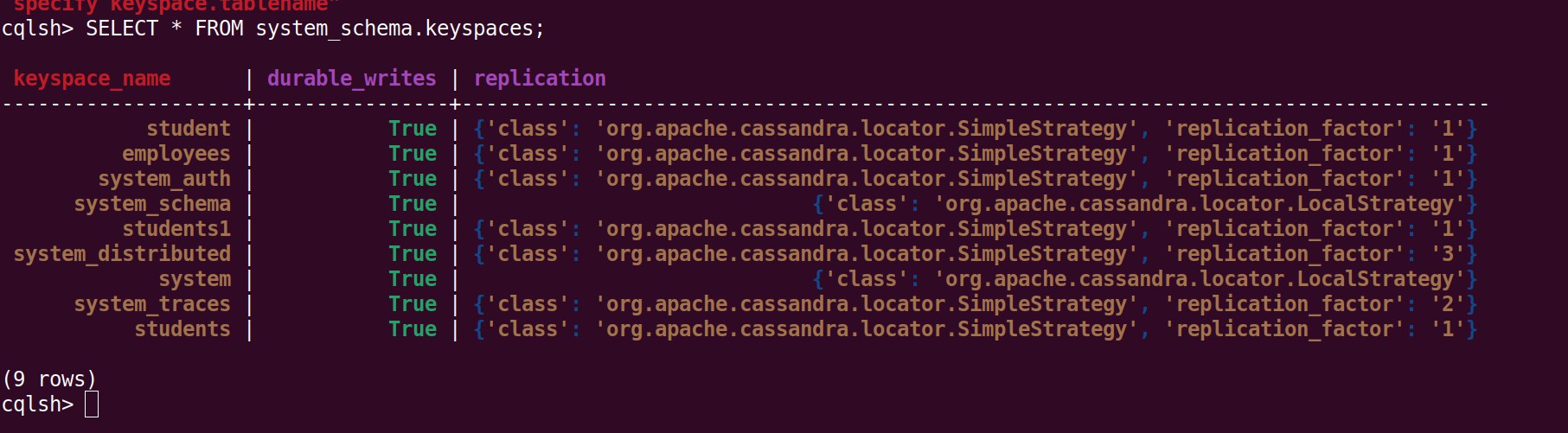
CREATE KEYSPACE Student WITH REPLICATION =

{'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 “Student”:

USE Student;

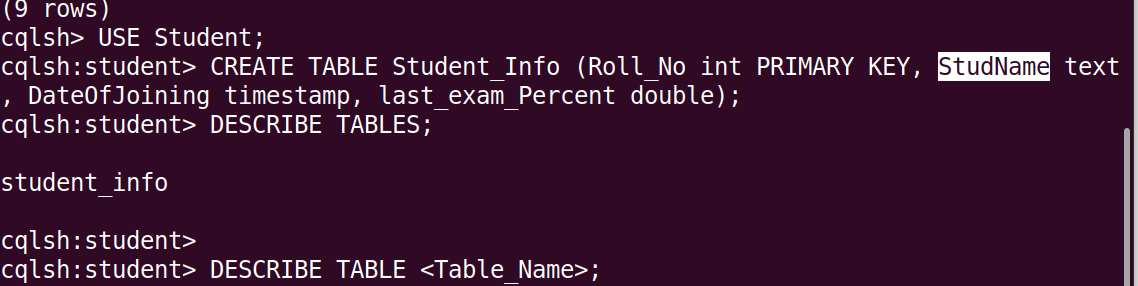
# To create table (column family) by name Student\_Info:

CREATE TABLE Student\_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 Student\_Info(Roll\_No, StudName, DateOfJoining, last\_exam\_Percent) VALUES (1,'Asha','2012-03-12',79.9)

INSERT INTO Student\_Info(Roll\_No, StudName, DateOfJoining, last\_exam\_Percent) VALUES (2,'Krian','2012-03-12',89.9)

INSERT INTO Student\_Info(Roll\_No, StudName, DateOfJoining, last\_exam\_Percent) VALUES (3,'Tarun','2012-03-12',78.9)

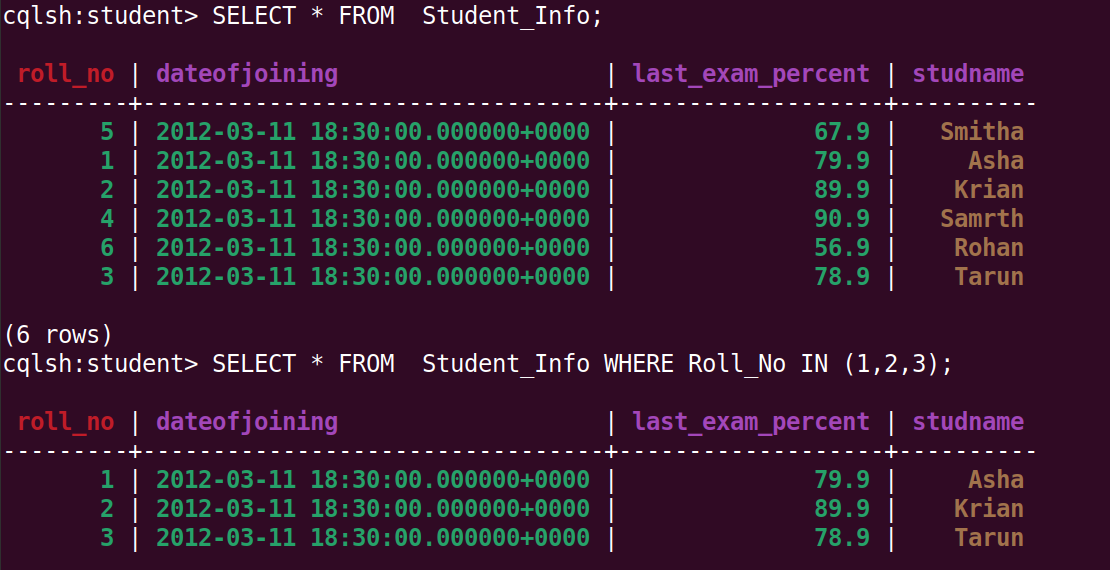
INSERT INTO Student\_Info(Roll\_No, StudName, DateOfJoining, last\_exam\_Percent) VALUES (4,'Samrth','2012-03-12',90.9)

INSERT INTO Student\_Info(Roll\_No, StudName, DateOfJoining, last\_exam\_Percent) VALUES (5,'Smitha','2012-03-12',67.9)

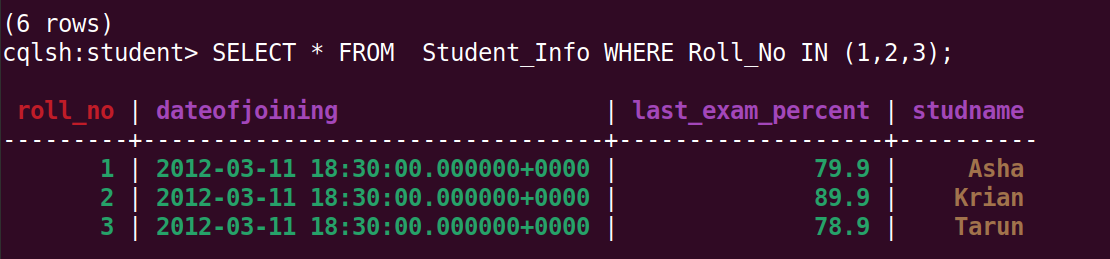
INSERT INTO Student\_Info(Roll\_No, StudName, DateOfJoining, last\_exam\_Percent) VALUES (6,'Rohan','2012-03-12',56.9)

APPLY BATCH;

# View data from the table “Student\_Info”

SELECT \* FROM Student\_Info;

# View data from the table “Student\_Info” where Rollo column either has a value 1 or 2 or 3

SELECT \* FROM Student\_Info WHERE Roll\_No IN (1,2,3);

# To execute a non primary key - will throw an error

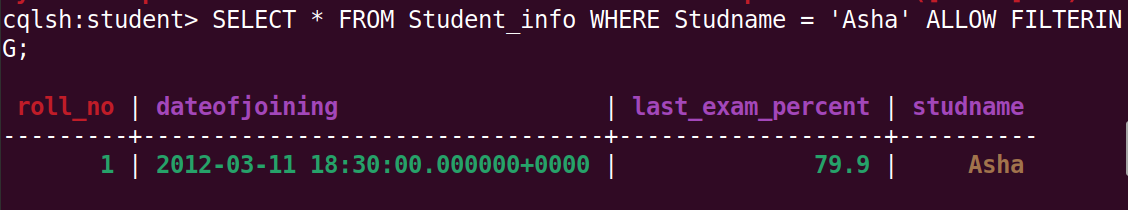
select \* from Student\_info where Studname= 'Asha';

# So create an INDEX on the Column as below:

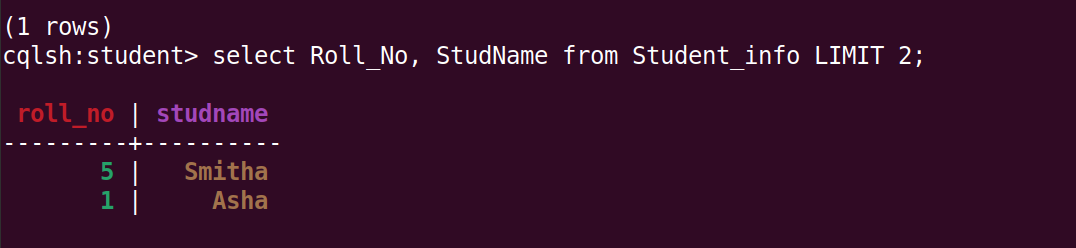
**To create an INDEX on StudName Column of the Student\_Info column family**

CREATE INDEX ON Student\_Info ( StudName);

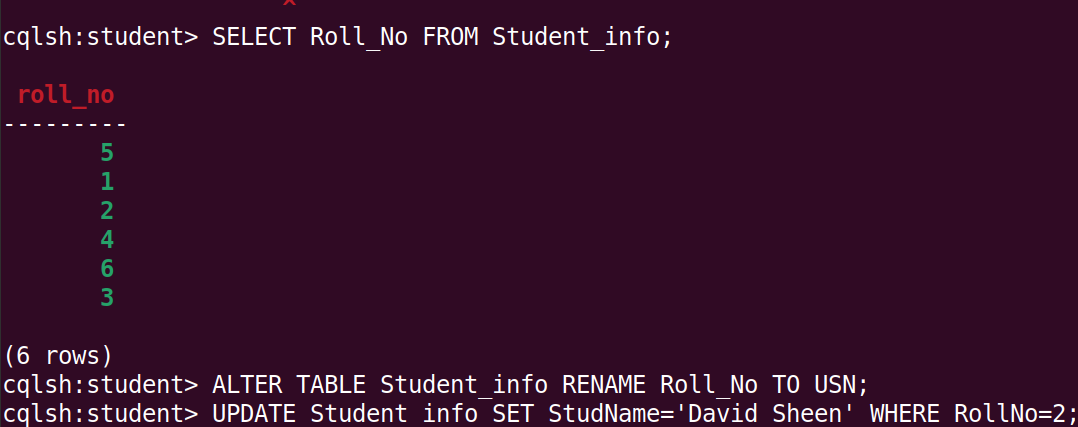
# Now execute the query based on the INDEXED Column:

select \* from Student\_info where Studname= 'Asha';

# To specify the number of rows retured in the output

select Roll\_No, StudName from Student\_info LIMIT 2;

# Alias for Column:

Select Roll\_No as “USN” from Student\_info;

**UPDATE**

UPDATE Student\_info SET StudName='David Sheen' WHERE RollNo=2; Lets try to update the primary key

UPDATE Student\_info SET Roll\_No=6 WHERE Roll\_No=3;

DELETE

DELETE LastExamPercent FROM Student\_info WHERE USN=2;

Delete a Row

DELETE FROM student\_info WHERE USN=2;

Set Collection

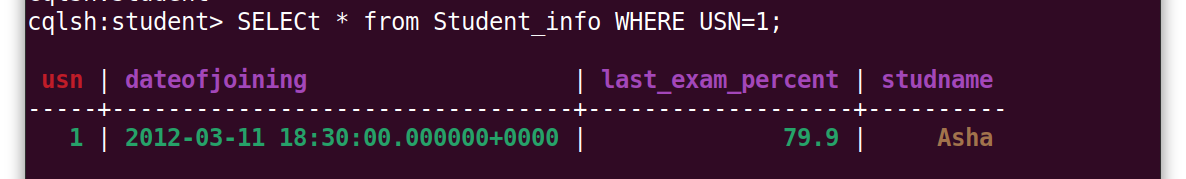
A column of type set consists of unordered unique values. However, when the column is queried, it returns, it returns the values in sorted order. For example, for text values, it sorts in alphabetical order.

ALTER TABLE Student\_info ADD hobbies set<text> List Collection

When the order of elements matter, one should go for a list collection. ALTER TABLE Student\_info ADD language list<text>;

UPDATE Student\_info

SET hobbies=hobbies+{'Chess,Table Tennis'} WHERE USN=1;

SELECt \* from Student\_info WHERE USN=1;

UPDATE Student\_info

SET langusge=language+['Hindi,English'] WHERE USN=1;

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

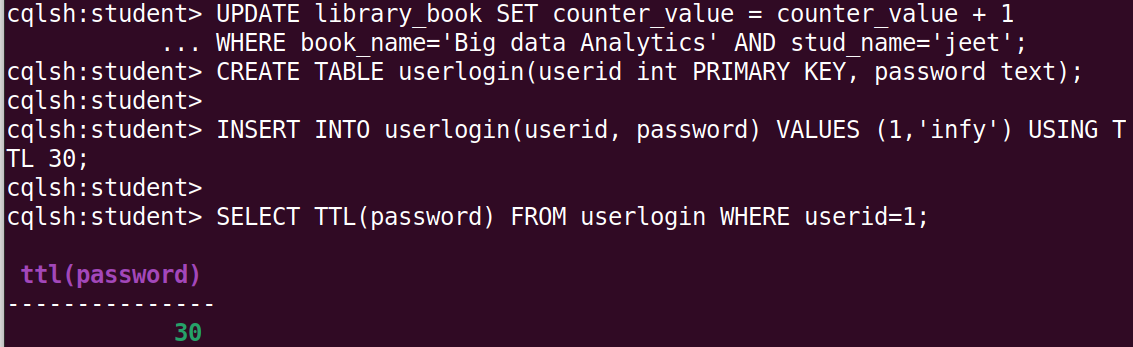
**USING A COUNTER**

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';

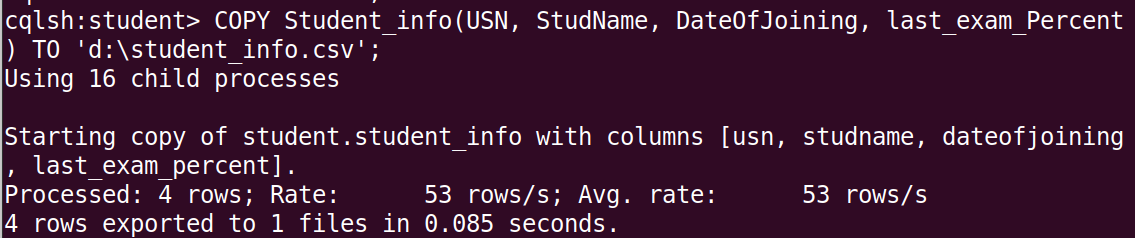


**TIME TO LIVE**

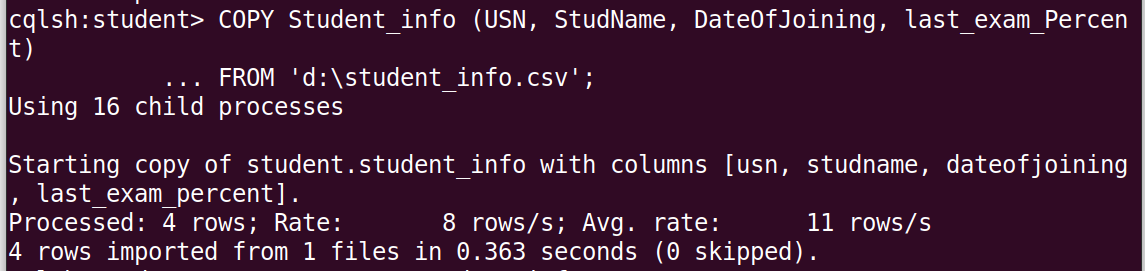
CREATE TABLE userlogin(userid int PRIMARY KEY, password text); INSERT INTO userlogin(userid, password) VALUES (1,'infy') USING TTL 30; SELECT TTL(password) FROM userlogin WHERE userid=1;

**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;**