1BM18CS089
SAHANA L

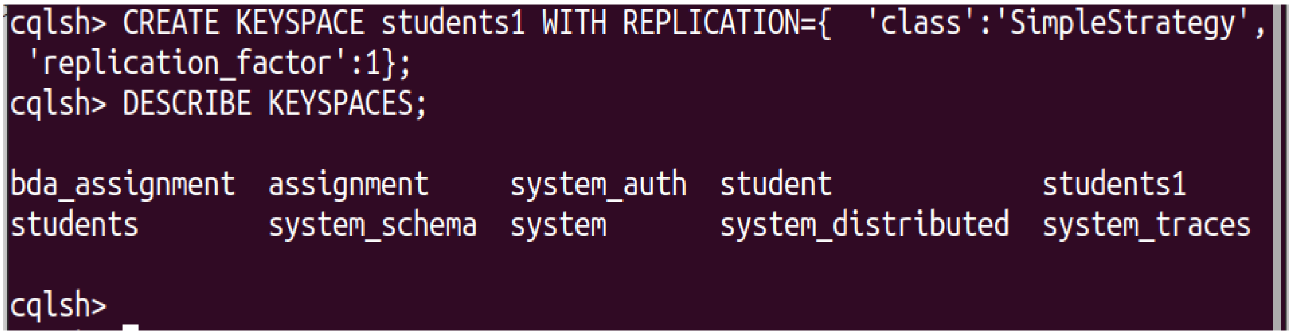
Execute the following commands in Cassandra CQL Shell:

1. Command to create Keyspace:

cqlsh>CREATE KEYSPACE students1 WITH REPLICATION={ 'class':'SimpleStrategy', 'replication_factor':1};

2**. Command to describe the existing Keyspaces:**

cqlsh> DESCRIBE KEYSPACES;



```
cqlsh> CREATE KEYSPACE students1 WITH REPLICATION={  'class':'SimpleStrategy',
 'replication_factor':1};
cqlsh> DESCRIBE KEYSPACES;

bda_assignment  assignment     system_auth  student             students1
students        system_schema  system       system_distributed  system_traces

cqlsh> 
```

3. Command to get more details on existing keyspaces such as keyspace name, durable writes, strategy class, strategy options etc.

cqlsh>SELECT * FROM system.schema_keyspaces;

4. Command to use keyspace students:

cqlsh>USE students1;

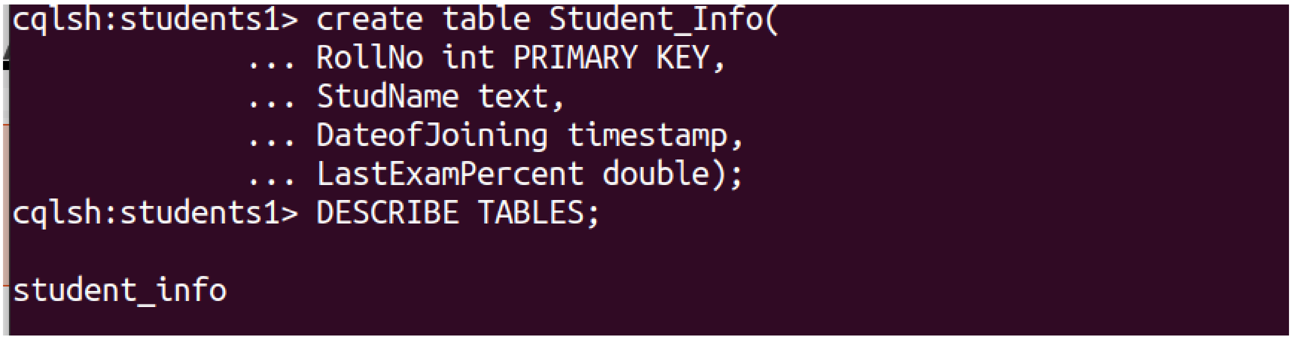
5. Command to create a column family or table by the name “Student_Info”.

cqlsh:students1>create table Student_Info(
RollNo int PRIMARY KEY,
StudName text,
DateofJoining timestamp,
LastExamPercent double);

1BM18CS089
SAHANA L

6. **Command to look up the names of all tables in current keyspace, or in all keyspaces if there is no current space:**

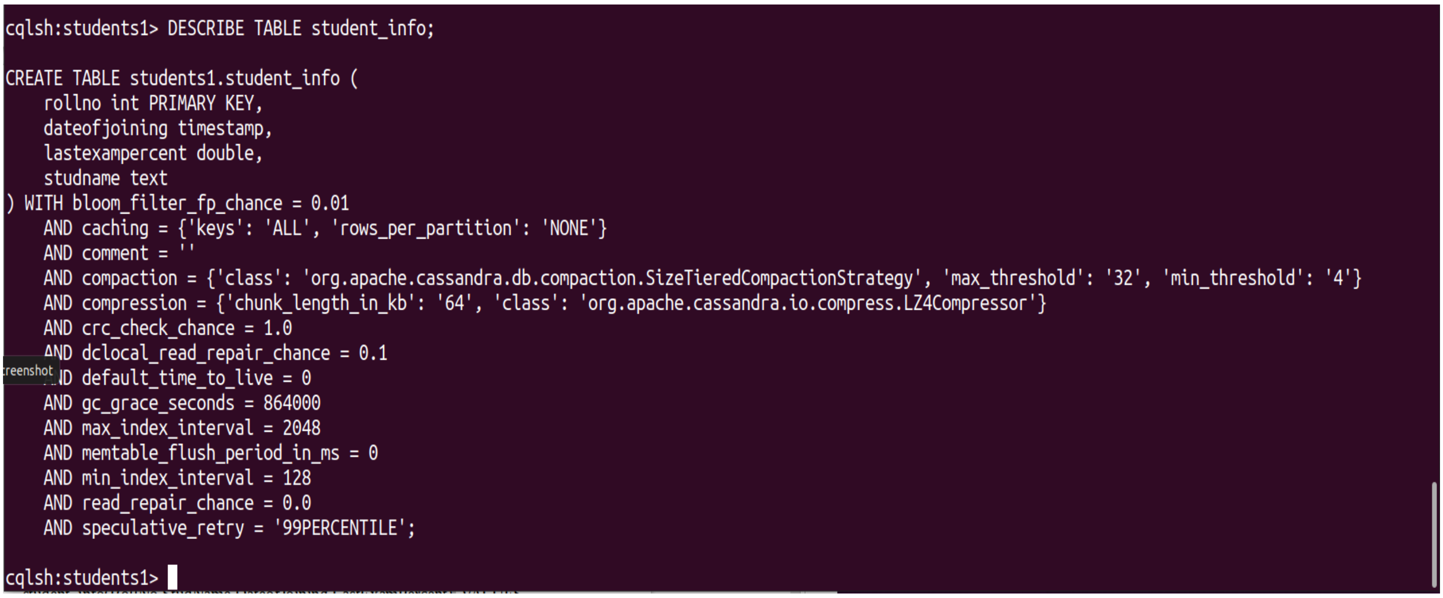
cqlsh:students1> DESCRIBE TABLES;



```
cqlsh:students1> create table Student_Info(
             ... RollNo int PRIMARY KEY,
             ... StudName text,
             ... DateofJoining timestamp,
             ... LastExamPercent double);
cqlsh:students1> DESCRIBE TABLES;

student_info
```

7. Command to describe the table Student_info:

cqlsh:students1> DESCRIBE TABLE student_info;



```
cqlsh:students1> DESCRIBE TABLE student_info;

CREATE TABLE students1.student_info (
    rollno int PRIMARY KEY,
    dateofjoining timestamp,
    lastexampercent 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
     ND 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:students1> 
```

1BM18CS089
SAHANA L

CRUD Operations

1. To insert data into the column family “student_info”.

BEGIN BATCH

INSERT INTO

student_info(RollNo,StudName,DateofJoining,LastExamPercent) VALUES (1,'Michael Storm','2012-03-29', 69.6)

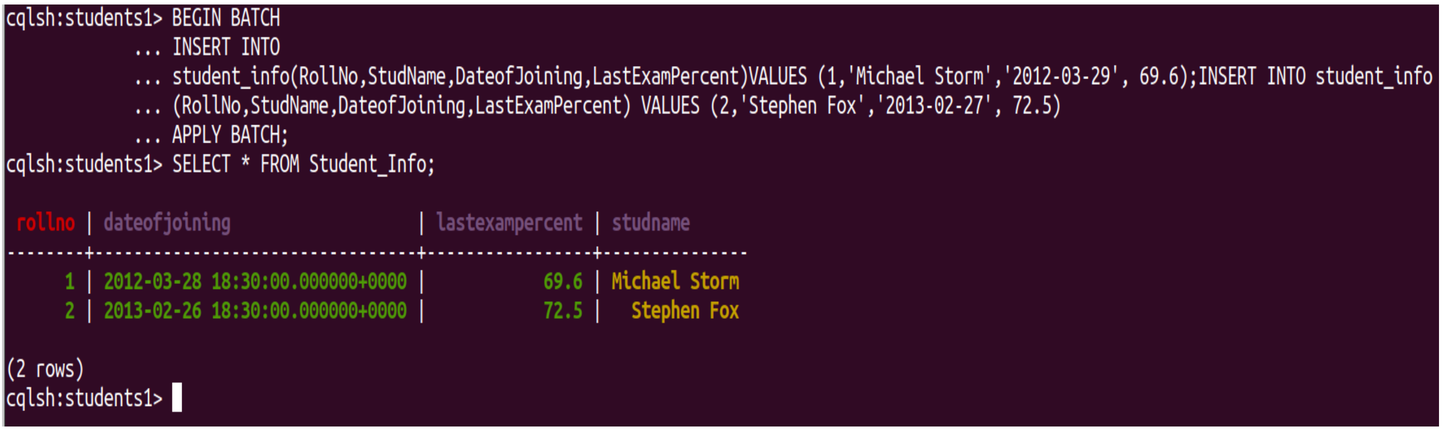
INSERT INTO student_info

(RollNo,StudName,DateofJoining,LastExamPercent) VALUES (2,'Stephen Fox','2013-02-27', 72.5)

APPLY BATCH;

2. To view the data from the table “Student_Info”.

SELECT * FROM Student_Info;



```
cqlsh:students1> BEGIN BATCH
             ... INSERT INTO
             ... student_info(RollNo,StudName,DateofJoining,LastExamPercent)VALUES (1,'Michael Storm','2012-03-29', 69.6);INSERT INTO student_info
             ... (RollNo,StudName,DateofJoining,LastExamPercent) VALUES (2,'Stephen Fox','2013-02-27', 72.5)
             ... APPLY BATCH;
cqlsh:students1> SELECT * FROM Student_Info;

 rollno | dateofjoining                   | lastexampercent | studname
--------+---------------------------------+-----------------+---------------
      1 | 2012-03-28 18:30:00.000000+0000 |            69.6 | Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |            72.5 |   Stephen Fox

(2 rows)
cqlsh:students1> 
```

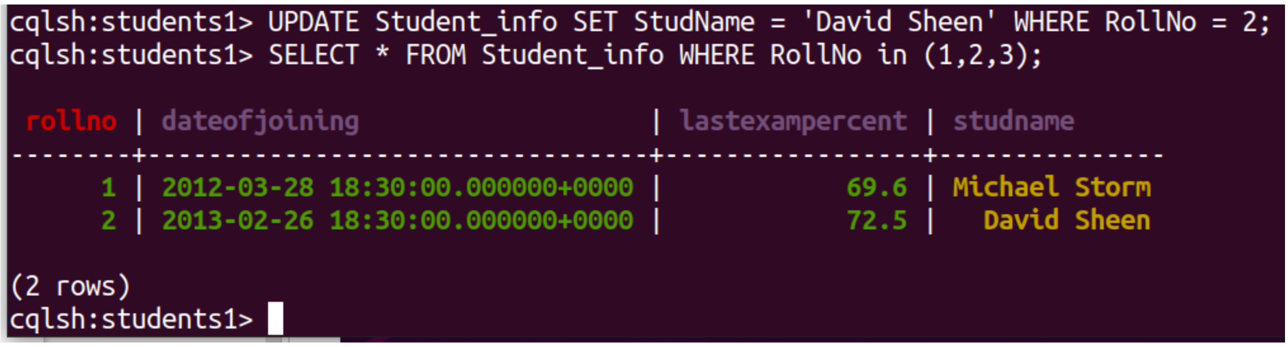
3. To update the value held in the “StudName” column of the “student_info” column family to “David Sheen” for the record where the RollNo column has value = 2.

UPDATE Student_info SET StudName = 'David Sheen' WHERE RollNo = 2;

1BM18CS089
SAHANA L

4. To view only those records where the RollNo column either has a value 1 or 2 or 3.

SELECT * FROM Student_info WHERE RollNo in (1,2,3);



```
cqlsh:students1> UPDATE Student_info SET StudName = 'David Sheen' WHERE RollNo = 2;
cqlsh:students1> SELECT * FROM Student_info WHERE RollNo in (1,2,3);

 rollno | dateofjoining                   | lastexampercent | studname
--------+---------------------------------+-----------------+--------------
      1 | 2012-03-28 18:30:00.000000+0000 |            69.6 | Michael Storm
      2 | 2013-02-26 18:30:00.000000+0000 |            72.5 |   David Sheen

(2 rows)
cqlsh:students1>
```

5. To create an index on the “studname” column of the “student_info” column family use the following statement.

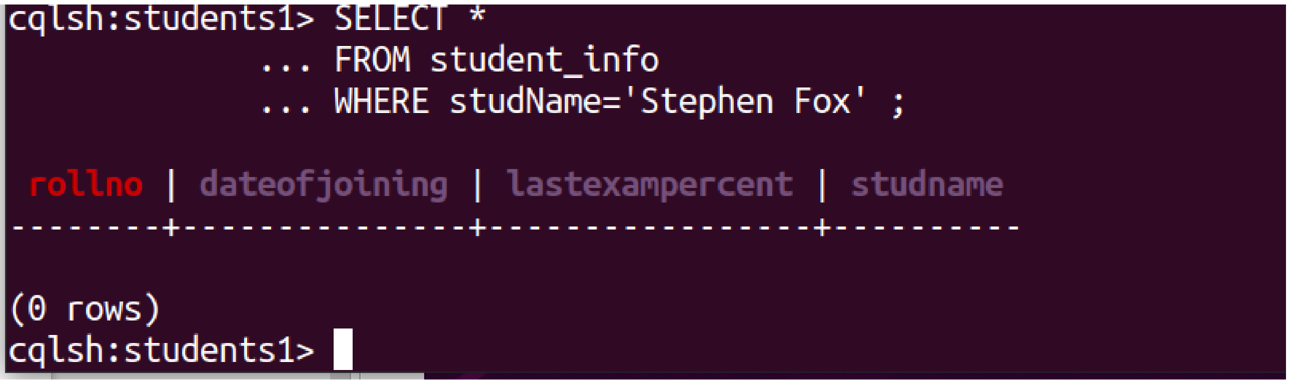
CREATE INDEX ON Student_Info(studname);

6. To execute the query using the index defined on "studname" column.

SELECT *

FROM student_info

WHERE studName='Stephen Fox' ;



```
cqlsh:students1> SELECT *
             ... FROM student_info
             ... WHERE studName='Stephen Fox' ;

 rollno | dateofjoining | lastexampercent | studname
--------+---------------+-----------------+----------

(0 rows)
cqlsh:students1>
```

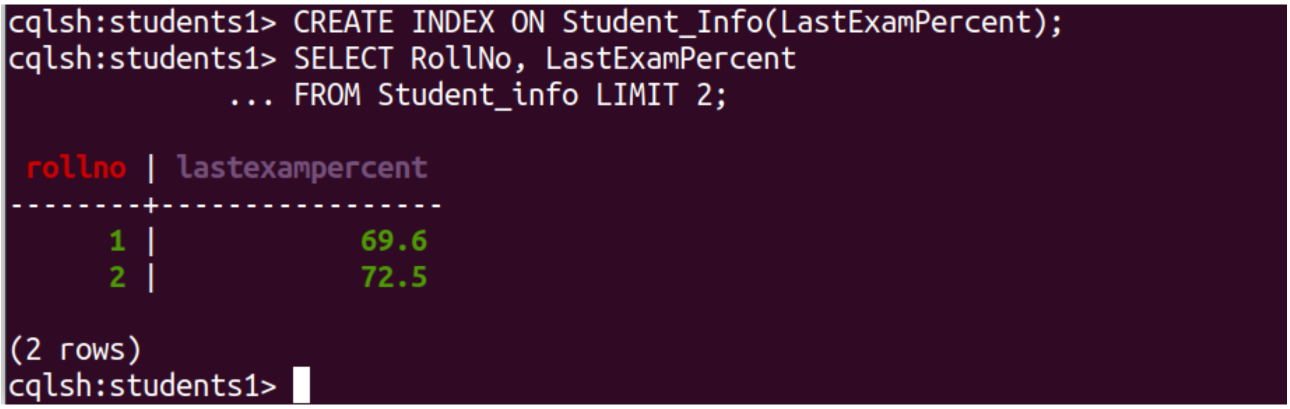
7. To create index on the "LastExamPercent" column of the "Student_Info" column family.

CREATE INDEX ON Student_Info(LastExamPercent);

8. To specify the number of rows returned in the output using limit.

SELECT RollNo, LastExamPercent

FROM Student_info LIMIT 2;

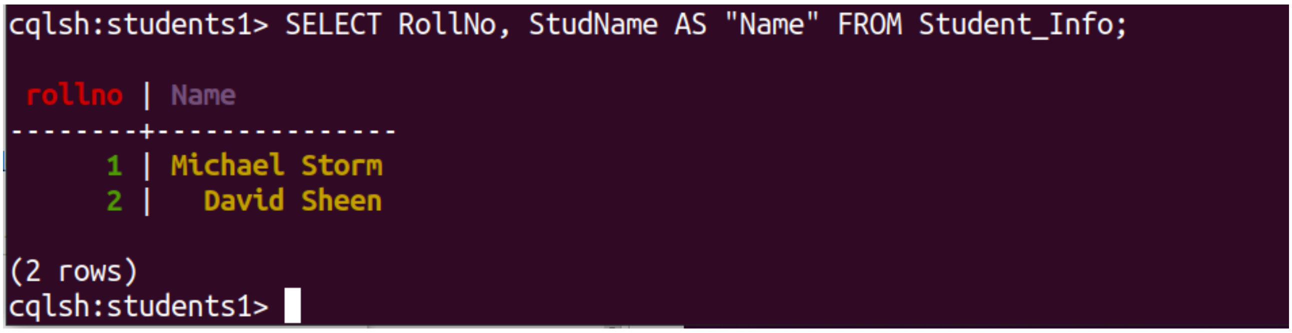


```
cqlsh:students1> CREATE INDEX ON Student_Info(LastExamPercent);
cqlsh:students1> SELECT RollNo, LastExamPercent
             ... FROM Student_info LIMIT 2;

 rollno | lastexampercent
--------+-----------------
      1 |            69.6
      2 |            72.5

(2 rows)
cqlsh:students1> 
```

9. To use column alias for the column ‘StudName’ in the Student_Info table.

SELECT RollNo, StudName AS "Name"

FROM Student_Info;



```
cqlsh:students1> SELECT RollNo, StudName AS "Name" FROM Student_Info;

 rollno | Name
--------+---------------
      1 | Michael Storm
      2 |   David Sheen

(2 rows)
cqlsh:students1> 
```

10. To update more than one column of a row of Cassandra table.

UPDATE Student_Info

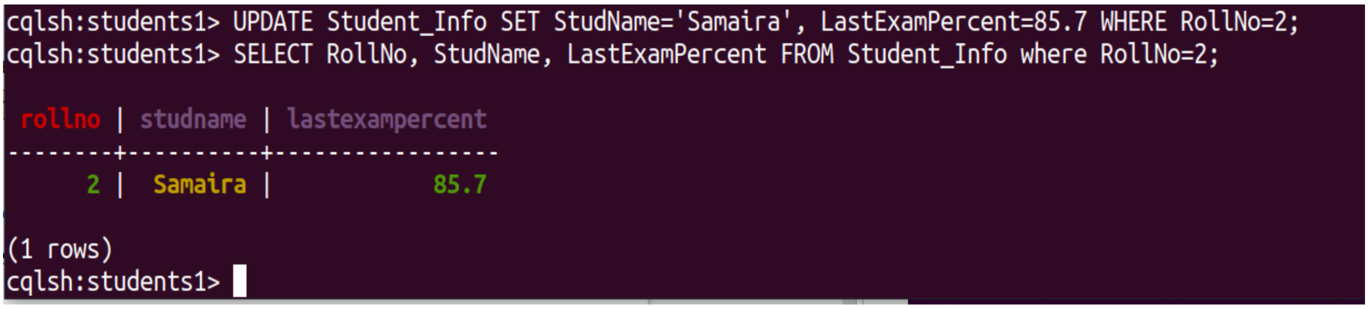
SET StudName=”Samaira”, LastExamPercent=85.7

WHERE RollNo=2;

After the Update:

SELECT RollNo, StudName, LastExamPercent

FROM Student_Info where RollNo=2;



```
cqlsh:students1> UPDATE Student_Info SET StudName='Samaira', LastExamPercent=85.7 WHERE RollNo=2;
cqlsh:students1> SELECT RollNo, StudName, LastExamPercent FROM Student_Info where RollNo=2;

 rollno | studname | lastexampercent
--------+----------+----------------
      2 |  Samaira |            85.7

(1 rows)
cqlsh:students1>
```

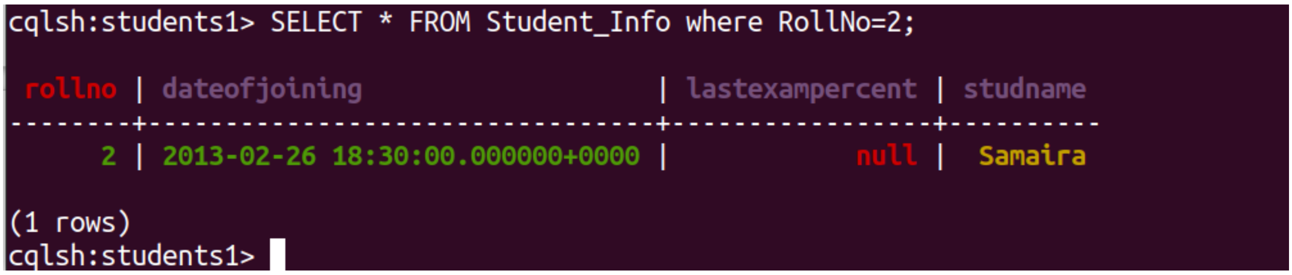
11.To delete the column “LastExamPercent” from the “student_info” table for the record where the RollNo = 2.

DELETE LastExamPercent FROM Student_Info WHERE RollNo=2;

After the Delete:

SELECT *

FROM Student_Info where RollNo=2;



```
cqlsh:students1> SELECT * FROM Student_Info where RollNo=2;

 rollno | dateofjoining                   | lastexampercent | studname
--------+---------------------------------+-----------------+----------
      2 | 2013-02-26 18:30:00.000000+0000 |            null |  Samaira

(1 rows)
cqlsh:students1>
```

12. To delete a row from the table Student_Info.

DELETE FROM Student_Info

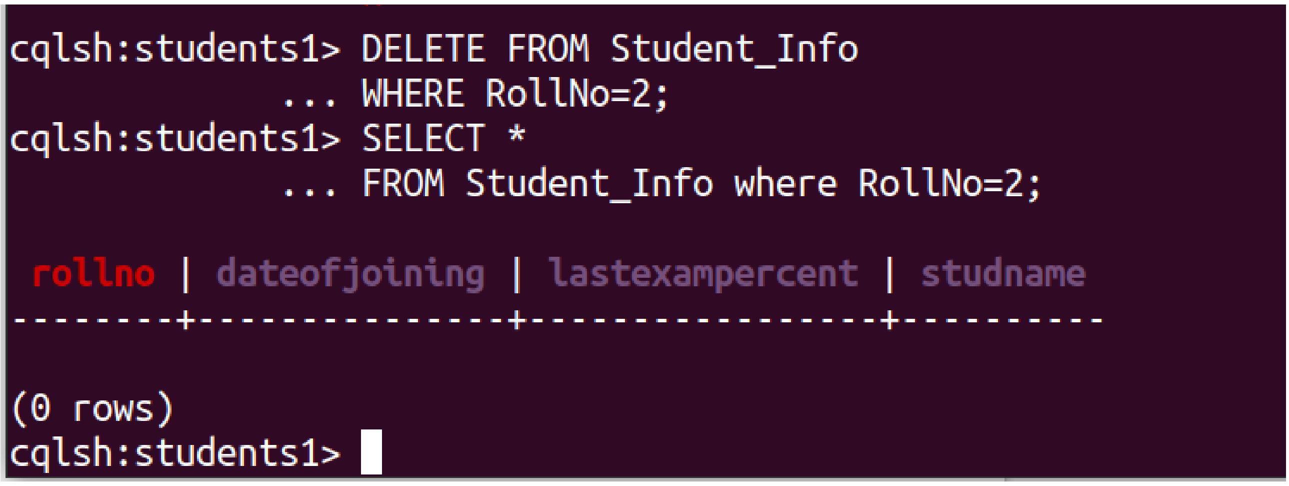
WHERE RollNo=2;

After the Delete:

SELECT *

FROM StudentInfo where RollNo=2;

1BM18CS089
SAHANA L



```
cqlsh:students1> DELETE FROM Student_Info
             ... WHERE RollNo=2;
cqlsh:students1> SELECT *
             ... FROM Student_Info where RollNo=2;

 rollno | dateofjoining | lastexampercent | studname
--------+---------------+-----------------+----------

(0 rows)
cqlsh:students1>
```

Project Table

1. To create Project Table in students keyspace.
students1> CREATE TABLE PROJECT_DETAILS(

... PROJECT_ID INT,

... PROJECT_NAME TEXT,

... STUD_NAME TEXT,

... RATING DOUBLE,

... DURATION INT,

... PRIMARY KEY(PROJECT_ID, PROJECT_NAME));

2. To Insert data
BEGIN BATCH

INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)

... VALUES(1,'MS data migration','David Sheen',3.5,720)

... INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)

... VALUES(1,'MS Data Warehouse','David Sheen',3.9,1440)

... INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)

... VALUES(2,'SAP Reporting','Stephen Fox',4.2,3000)

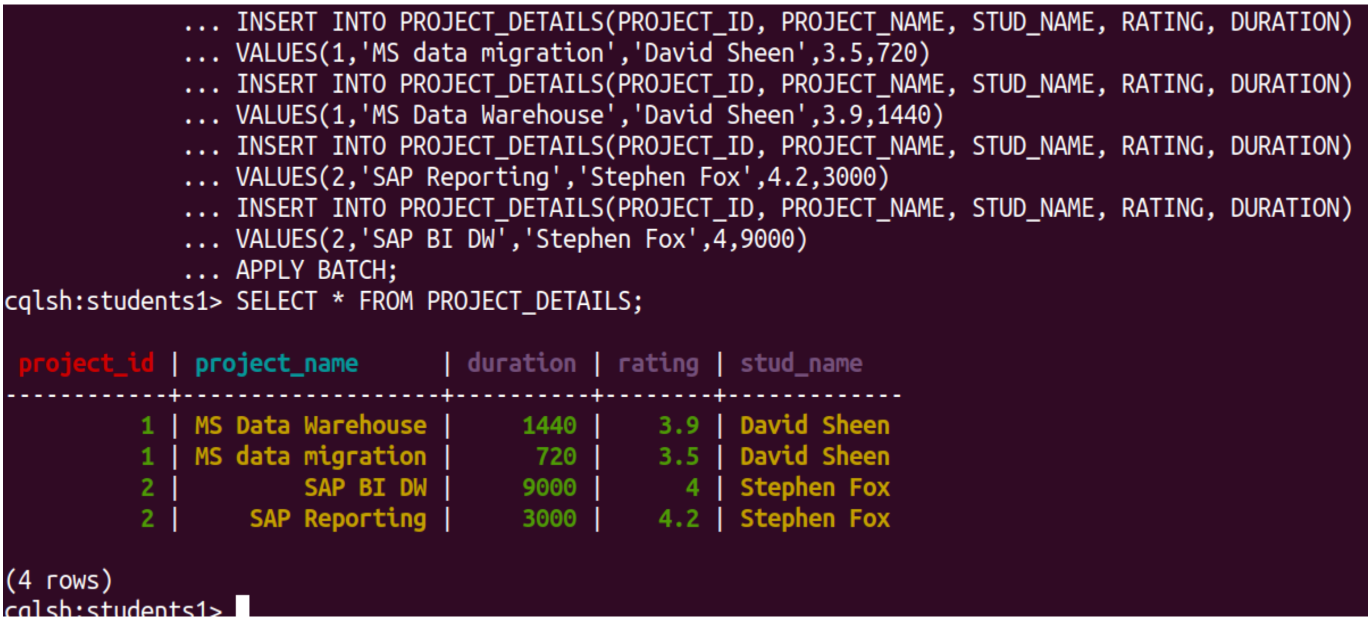
... INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)

... VALUES(2,'SAP BI DW','Stephen Fox',4,9000)

... APPLY BATCH;

3. To view all rows of project_details table.

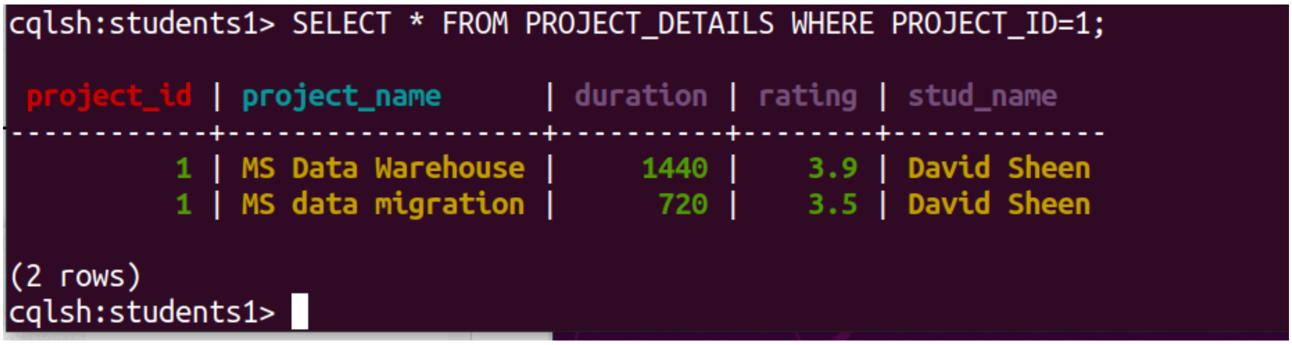
SELECT * FROM PROJECT_DETAILS



```
             ... INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)
             ... VALUES(1,'MS data migration','David Sheen',3.5,720)
             ... INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)
             ... VALUES(1,'MS Data Warehouse','David Sheen',3.9,1440)
             ... INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)
             ... VALUES(2,'SAP Reporting','Stephen Fox',4.2,3000)
             ... INSERT INTO PROJECT_DETAILS(PROJECT_ID, PROJECT_NAME, STUD_NAME, RATING, DURATION)
             ... VALUES(2,'SAP BI DW','Stephen Fox',4,9000)
             ... APPLY BATCH;
cqlsh:students1> SELECT * FROM PROJECT_DETAILS;

 project_id | project_name      | duration | rating | stud_name
------------+-------------------+----------+--------+-------------
          1 | MS Data Warehouse |     1440 |    3.9 | David Sheen
          1 | MS data migration |      720 |    3.5 | David Sheen
          2 |         SAP BI DW |     9000 |      4 | Stephen Fox
          2 |     SAP Reporting |     3000 |    4.2 | Stephen Fox

(4 rows)
cqlsh:students1>
```

4. To view row/record from the “project_details” table wherein the project_id=1

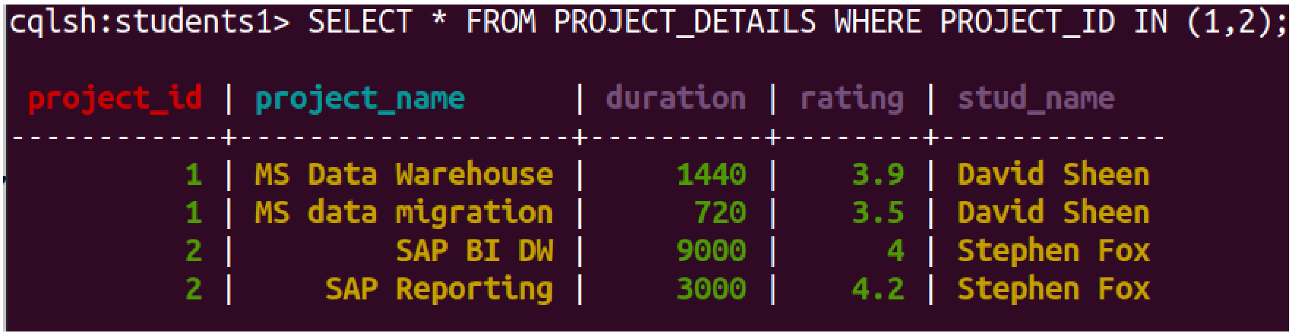
SELECT * FROM PROJECT_DETAILS WHERE PROJECT_ID=1;



```
cqlsh:students1> SELECT * FROM PROJECT_DETAILS WHERE PROJECT_ID=1;

 project_id | project_name       | duration | rating | stud_name
------------+--------------------+----------+--------+-------------
          1 | MS Data Warehouse |     1440 |    3.9 | David Sheen
          1 | MS data migration |      720 |    3.5 | David Sheen

(2 rows)
cqlsh:students1>
```

5. **To sort order the rows/records of the "project_details" in descending order of project_name.**

SELECT * FROM PROJECT_DETAILS WHERE PROJECT_ID IN (1,2);



```
cqlsh:students1> SELECT * FROM PROJECT_DETAILS WHERE PROJECT_ID IN (1,2);

 project_id | project_name       | duration | rating | stud_name
------------+--------------------+----------+--------+-------------
          1 | MS Data Warehouse |     1440 |    3.9 | David Sheen
          1 | MS data migration |      720 |    3.5 | David Sheen
          2 |         SAP BI DW |     9000 |      4 | Stephen Fox
          2 |     SAP Reporting |     3000 |    4.2 | Stephen Fox
```

SELECT * FROM PROJECT_DETAILS WHERE PROJECT_ID IN (1,2) ORDER BY PROJECT_NAME DESC;