**Name:- Samreen Shaikh Roll no:- 64**

**Subject:- Big Data Analytics Date:- 10/01/2025**

**PRACTICAL - 11**

**CASSANDRA KEYSPACE, CASSANDRA TABLE, CASSANDRA CRUD, CQL OPERATIONS**

**1. KEYSPACE OPERATIONS:**

**Q1] Create keyspace[database] tycs with replication simple and factor 1**

**Create Keyspace:**

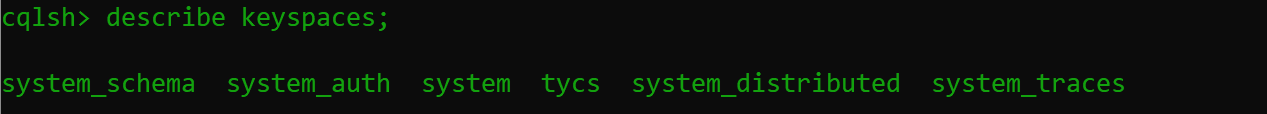
cqlsh> CREATE KEYSPACE tycs WITH replication = {'class':'SimpleStrategy', 'replication\_factor' : 1};



**Q2] Show the structure of keyspaces**

**Verification:**

cqlsh> describe keyspaces;

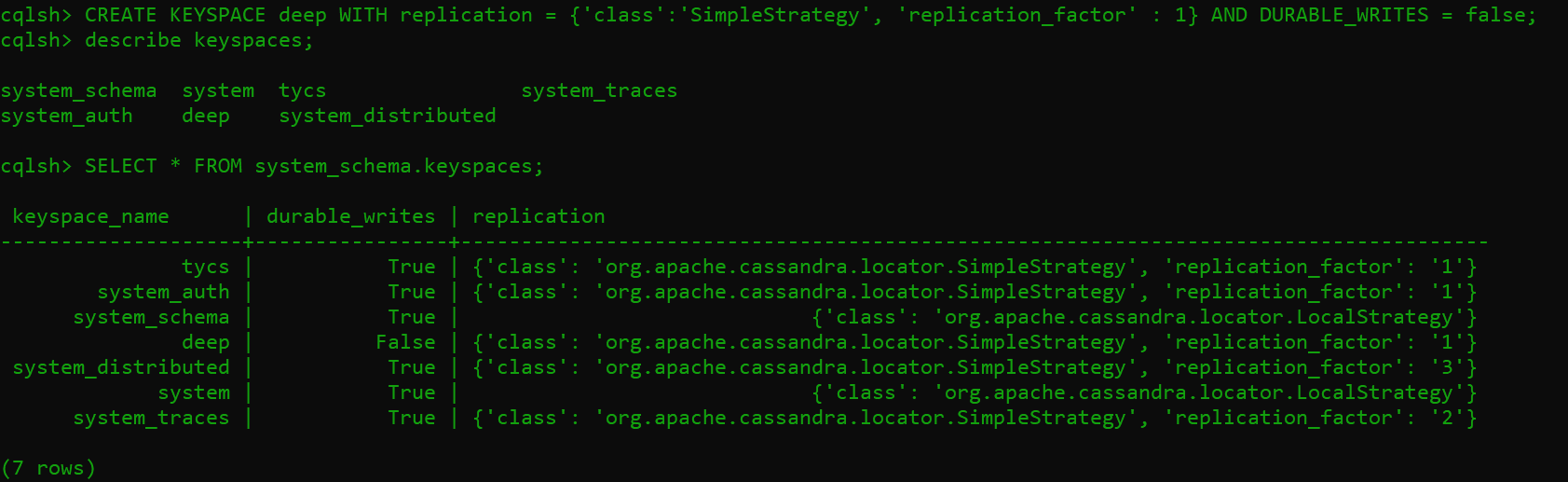


**Q3] Create keyspace[database] deep with replication simple and factor 1 and durable properties as false**

**Durable\_writes:**

By default, the durable\_writes properties of a table is set to true, however it can be set to false.

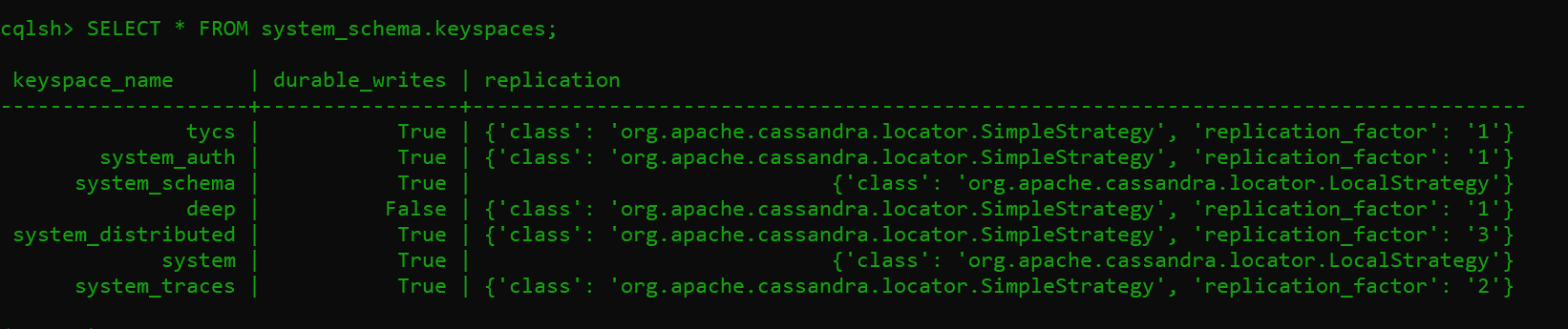
cqlsh> CREATE KEYSPACE deep WITH replication = {'class':'SimpleStrategy', 'replication\_factor' : 1} AND DURABLE\_WRITES = false;



**Q4] Display the durable\_writes of the keyspaces**

**Verification:**

cqlsh> select \* from system\_schema.keyspaces;



**Q5] Use the keyspace tycs**

**Using a Keyspace:**

cqlsh> use tycs;



**Q6] Alter the keyspace deep replication class from Simple to Network**

**Altering a KeySpace:**

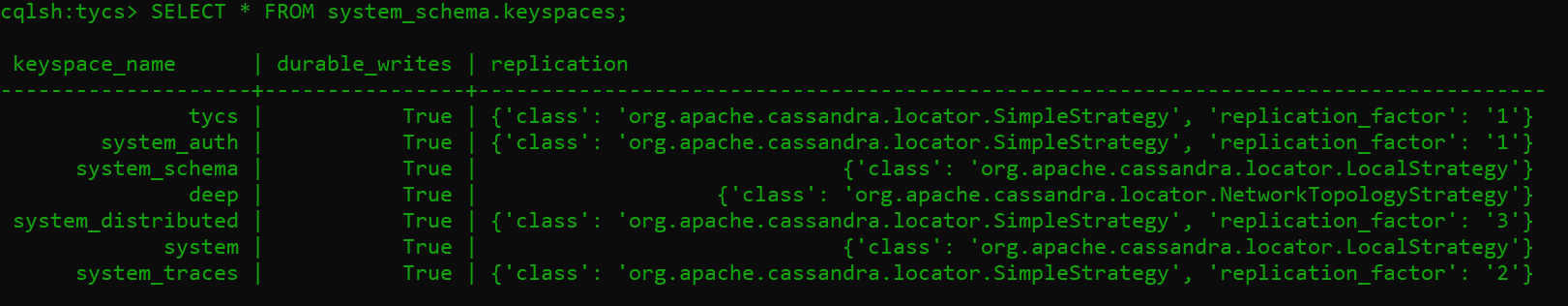
cqlsh:tycs> ALTER KEYSPACE deep WITH replication = {'class':'NetworkTopologyStrategy'} AND DURABLE\_WRITES = true;



**Q7] Display the durable\_writes of deep keyspace.**

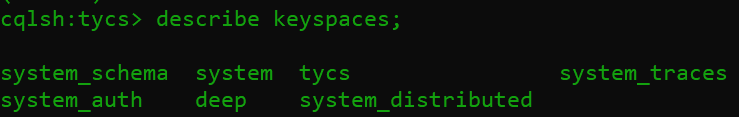
**Verification:**

cqlsh:tycs> SELECT \* FROM system\_schema.keyspaces;



**Q8] Display the keyspaces**

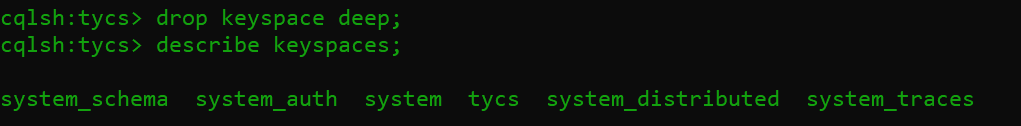
**Dropping a Keyspace:**

cqlsh:tycs> describe keyspaces;****

**Q9] Drop keyspace deep**

cqlsh:tycs> drop keyspace deep;

cqlsh:tycs> describe keyspaces;



**2. TABLE OPERATIONS:**

**Q10] Create table report in keyspace tycs having fiels subject id integer Primary key,subject name text,marks var int,grade text.**

**Creating a Table:**

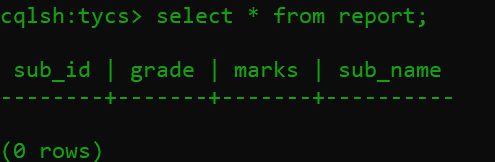
cqlsh:tycs> CREATE TABLE report(sub\_id int PRIMARY KEY,sub\_name text,marks varint,grade text);



**Q11] Dispplay table report**

**Verification:**

cqlsh:tycs> select \* from report;



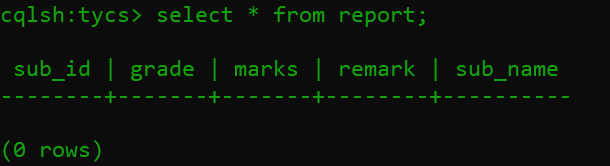
**Q12] Add new field remark in report table**

**Altering a Table:**

cqlsh:tycs> ALTER TABLE report ADD remark text;



**Verification:**

cqlsh:tycs> select \* from report;

**Q13] Drop new field remark from report table**

**Dropping a Column:**

cqlsh:tycs> ALTER TABLE report DROP remark;



**3. CRUD OPERATIONS:**

**Q14] Insert 5 Records in report table**

**Creating Data in Table:**

cqlsh:tycs> ALTER TABLE report DROP remark;

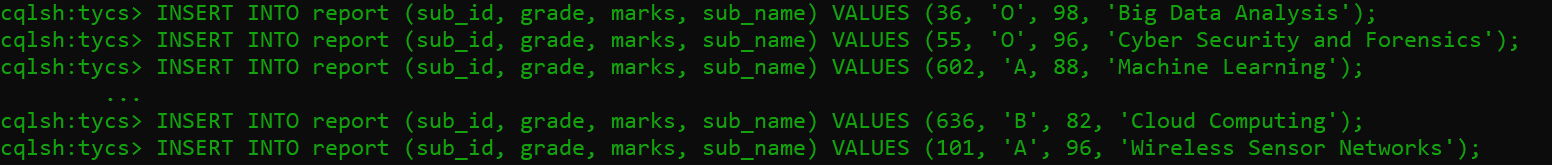
cqlsh:tycs> INSERT INTO report (sub\_id, grade, marks, sub\_name) VALUES (36, 'O', 98, 'Big Data Analysis');

cqlsh:tycs> INSERT INTO report (sub\_id, grade, marks, sub\_name) VALUES (55, 'O', 96, 'Cyber Security and Forensics');

cqlsh:tycs> INSERT INTO report (sub\_id, grade, marks, sub\_name) VALUES (602, 'A, 88, 'Machine Learning');

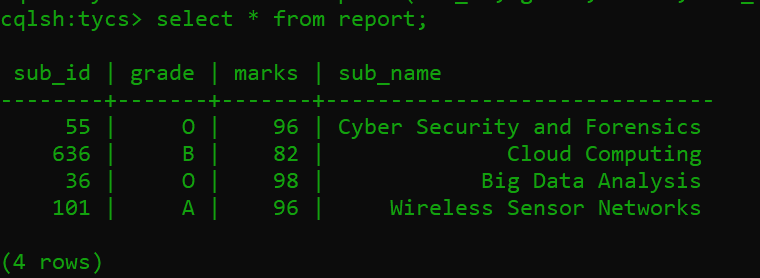
cqlsh:tycs> INSERT INTO report (sub\_id, grade, marks, sub\_name) VALUES (636, 'B', 82, 'Cloud Computing');

cqlsh:tycs> INSERT INTO report (sub\_id, grade, marks, sub\_name) VALUES (101, 'A', 96, 'Wireless Sensor Networks');



**Q15] Display the inserted records**

cqlsh:tycs> select \* from report;



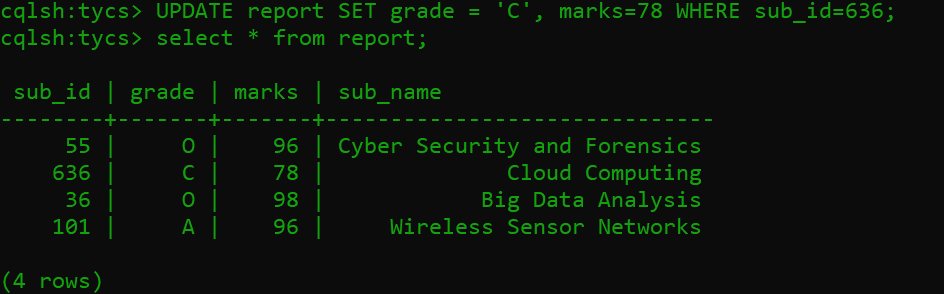
**Q16] Update the record set grade as C and Marks as 78 where subject id is 636**

**Updating Data in a Table:**

cqlsh:tycs> UPDATE report SET grade = 'C', marks=78 WHERE sub\_id=636;

**Verification:**

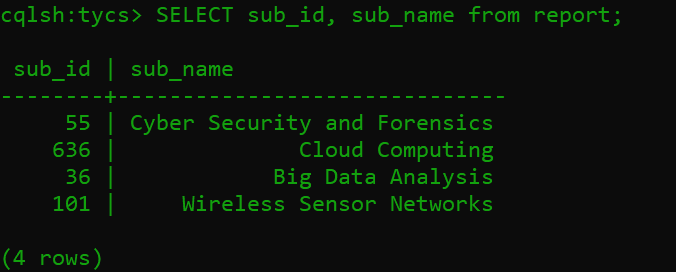
cqlsh:tycs> select \* from report;



**Q17] Display subject id and subject name only of all records**

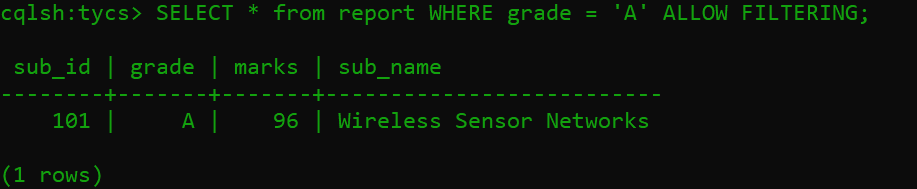
**Reading Required Columns:**

cqlsh:tycs> SELECT sub\_id, sub\_name from report;



**Q18] Display record of only where grade is A by applying filter**

cqlsh:tycs> SELECT \* from report WHERE grade = 'A' ALLOW FILTERING;



**Q19]** **Display only subject id and subject name of only where grade is A by applying filter**

cqlsh:tycs> SELECT sub\_id, sub\_name from report WHERE grade = 'A' ALLOW FILTERING;



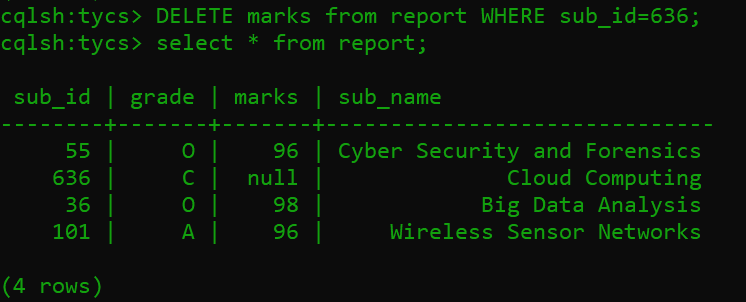
**Q20] Delete the marks of record where subject id is 636**

**Deleting Datafrom a Table:**

cqlsh:tycs> DELETE marks from report WHERE sub\_id=636;

**Verification:**

cqlsh:tycs> select \* from report;



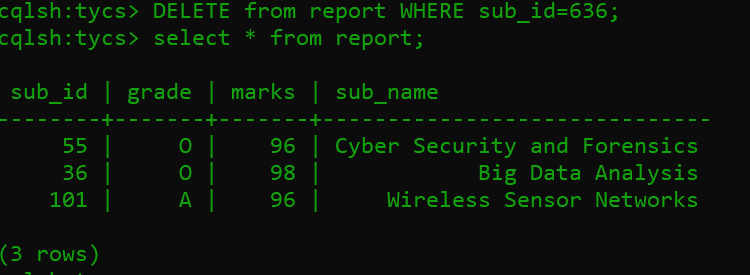
**Q21]** **Delete the record where subject id is 636**

**Deleting an Entire Row:**

cqlsh:tycs> DELETE from report WHERE sub\_id=636;

**Verification**

cqlsh:tycs> select \* from report;



**CQL Collections**

**1. USING LIST**

**Question:**  
How to store a list of emails for each user?

**Query:**

CREATE TABLE users (name TEXT PRIMARY KEY,emails LIST<TEXT>);

****

**Insert Data:**

INSERT INTO users (name, emails) VALUES ('John Doe', ['john@example.com', 'jdoe@example.com']);

**Output:**



**2. USING SET**

**Question:**  
How to store a set of phone numbers for each user?

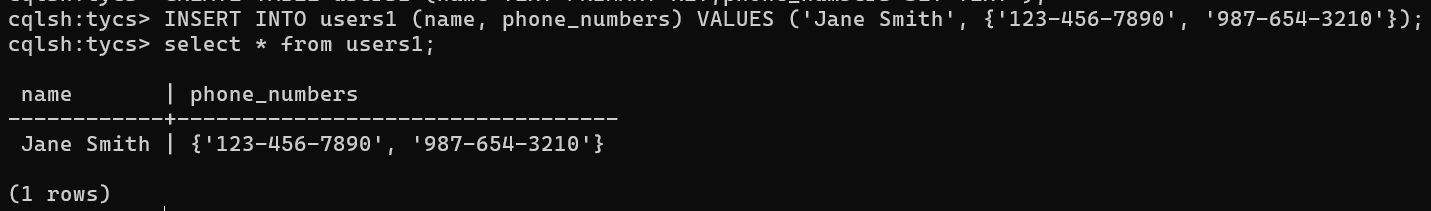
**Query:**

cqlsh:tycs> CREATE TABLE users1 (name TEXT PRIMARY KEY,phone\_numbers SET<TEXT>);

****

**Insert Data:**

INSERT INTO users1 (name, phone\_numbers) VALUES ('Jane Smith', {'123-456-7890', '987-654-3210'});



**3. USING MAP**

**Question:**  
How to store a map of attributes (like address and city) for each user?

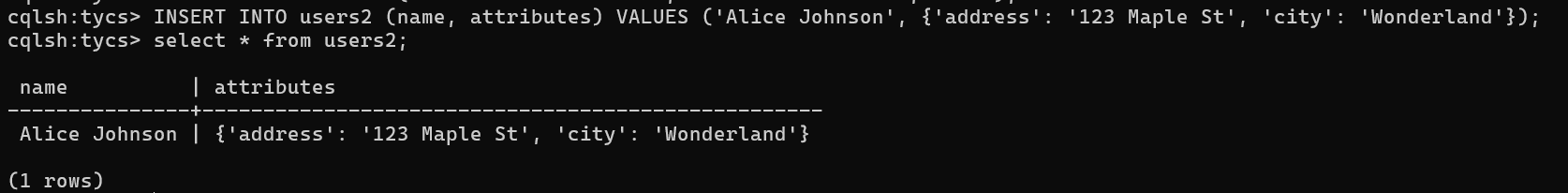
**Query:**

cqlsh:tycs> CREATE TABLE users2 (name TEXT PRIMARY KEY,attributes MAP<TEXT, TEXT>);



**Insert Data:**

INSERT INTO users2 (name, attributes) VALUES ('Alice Johnson', {'address': '123 Maple St', 'city': 'Wonderland'});



**Full Example**

CREATE TABLE user\_info (name TEXT PRIMARY KEY,emails LIST<TEXT>,phone\_numbers SET<TEXT>,attributes MAP<TEXT, TEXT>);



**Inserting data into the combined table:**

INSERT INTO user\_info (name, emails, phone\_numbers, attributes) VALUES ('Bob Brown', ['bob@example.com', 'bbrown@example.com'], {'555-1234', '555-5678'}, {'address': '456 Oak Rd', 'city': 'Smalltown'});

