Exercise: 6

Table and Batch Operations in Cassandra with DataStax Astra :

Aim:

To master table and batch operations in Cassandra, focusing on efficient data manipulation, querying, and updating techniques.

Procedure:

Initialization :

Display all available keyspaces to understand the database schema landscape.

Select a specific keyspace (default_keyspace) for subsequent operations to ensure all actions are scoped within a targeted environment.

Table Setup :

Create a users table with appropriate columns (user_id, fname, lname) to store user information, identifying user_id as the primary key to ensure unique identification.

Data Insertion :

Insert individual records into the users table using single-row insert commands for initial data population.

Employ batch operations to insert multiple records simultaneously, enhancing efficiency for bulk data entry.

Data Retrieval and Display:

Execute queries to fetch and display data from the users table, including:

Retrieving all records.

Selectively displaying specific columns.

Applying conditions to filter results (e.g., by last name).

Ordering results and limiting the number of records retrieved.

Perform aggregation to count records based on common attributes (e.g., last name) and identify unique values.

Data Modification :

Update records based on specific criteria, adjusting fields as required.Utilize conditional updates to modify records under defined conditions, ensuring targeted and precise data manipulation.

Advanced Data Management :

Combine various data manipulation operations (inserts, updates, deletes) within single batch operations to streamline complex modifications.

Extend the users table schema by adding new columns, allowing for richer data representation.

Cleanup and Schema Evolution :

Execute deletion operations for both individual records and specific column values, maintaining data relevance and accuracy.

Employ batch operations for efficient removal of multiple records.

Conduct structural modifications by removing the users table entirely or clearing its contents, facilitating schema evolution and data refreshment.

Create an additional table with a composite primary key, demonstrating advanced schema design for nuanced data relationships.

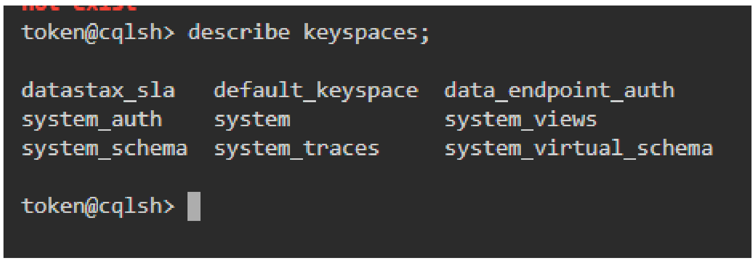
Queries :

1. Display the set of key spaces :

Select * from system.schema_keyspaces;

(OR)

describe keyspaces;



```
token@cqlsh> describe keyspaces;

datastax_sla   default_keyspace  data_endpoint_auth
system_auth    system            system_views
system_schema  system_traces     system_virtual_schema

token@cqlsh>
```

2. Connect with 'default_keyspace' key space :

use default_keyspace;

3. Create a table named users with columns for user_id, fname, and lname.

CREATE TABLE users (user_id int PRIMARY KEY,fname text,lname text);

4. Insert data into the users table one by one

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(1,'john','smith');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(2,'robert','jack');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(3,'daniel','bala');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(4,'priya','sweatha');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(5,'siva','geetha');

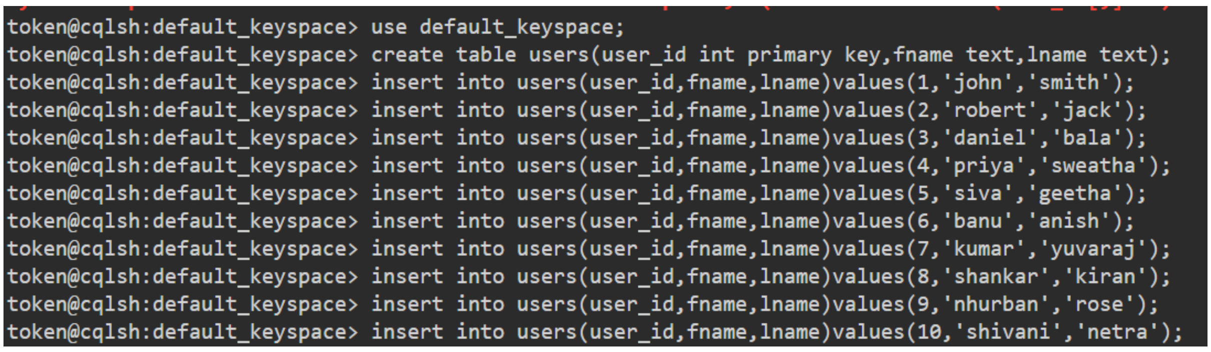
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(6,'banu','anish');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(7,'kumar','yuvaraj');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(8,'shankar','kiran');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(9,'nhurban','rose');

token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(10,'shivani','netra');



```
token@cqlsh:default_keyspace> use default_keyspace;
token@cqlsh:default_keyspace> create table users(user_id int primary key,fname text,lname text);
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(1,'john','smith');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(2,'robert','jack');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(3,'daniel','bala');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(4,'priya','sweatha');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(5,'siva','geetha');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(6,'banu','anish');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(7,'kumar','yuvaraj');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(8,'shankar','kiran');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(9,'nhurban','rose');
token@cqlsh:default_keyspace> insert into users(user_id,fname,lname)values(10,'shivani','netra');
```

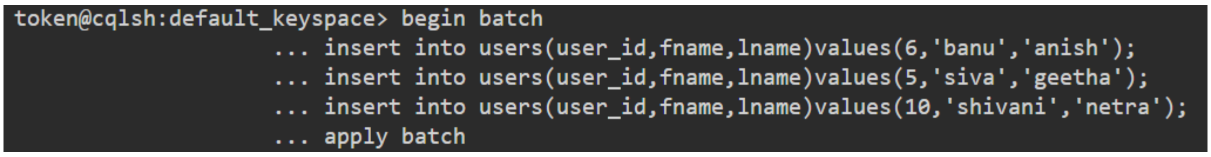
5. Insert multiple rows into the users table in Cassandra using a batch operation.

BEGIN BATCH

INSERT INTO users (user_id, fname, lname) VALUES (1745, 'John', 'Smith');

INSERT INTO users (user_id, fname, lname) VALUES (1746, 'Jane', 'Doe');

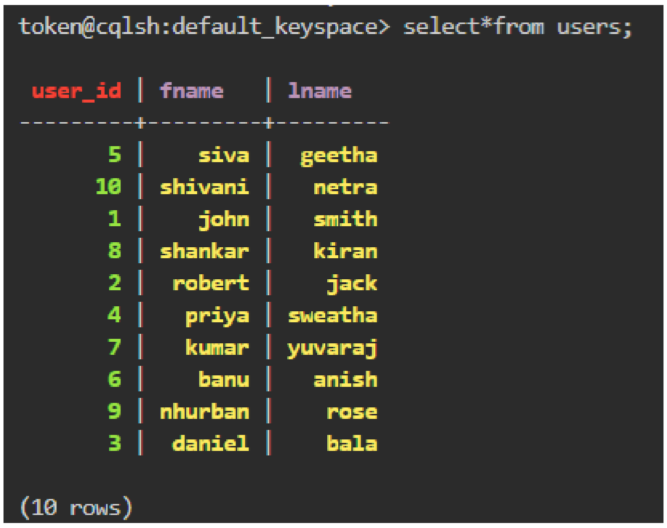
APPLY BATCH;



```
token@cqlsh:default_keyspace> begin batch
                          ... insert into users(user_id,fname,lname)values(6,'banu','anish');
                          ... insert into users(user_id,fname,lname)values(5,'siva','geetha');
                          ... insert into users(user_id,fname,lname)values(10,'shivani','netra');
                          ... apply batch
```

6. Display all data from the users table.

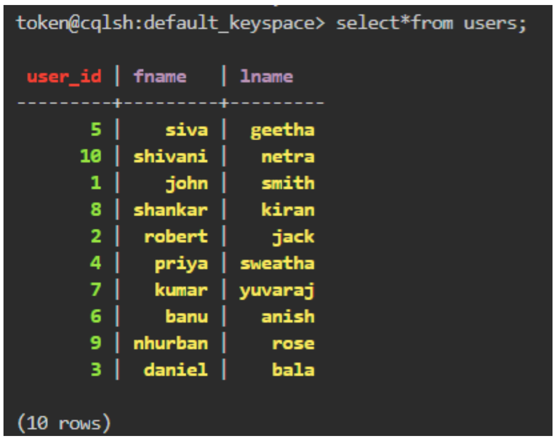
SELECT * FROM users;



```
token@cqlsh:default_keyspace> select*from users;

 user_id | fname   | lname
---------+---------+---------
       5 |    siva |  geetha
      10 | shivani |   netra
       1 |    john |   smith
       8 | shankar |   kiran
       2 |  robert |    jack
       4 |   priya | sweatha
       7 |   kumar | yuvaraj
       6 |    banu |   anish
       9 | nhurban |    rose
       3 |  daniel |    bala

(10 rows)
```

7. Display only the user_id and fname columns from the users table

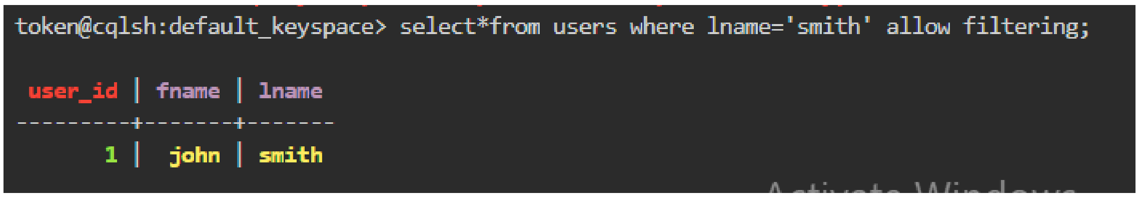
SELECT user_id, fname FROM users;



```
token@cqlsh:default_keyspace> select*from users;

 user_id | fname   | lname
---------+---------+---------
       5 |    siva |  geetha
      10 | shivani |   netra
       1 |    john |   smith
       8 | shankar |   kiran
       2 |  robert |    jack
       4 |   priya | sweatha
       7 |   kumar | yuvaraj
       6 |    banu |   anish
       9 | nhurban |    rose
       3 |  daniel |    bala

(10 rows)
```

8. Display data from the users table where lname is 'Smith'.

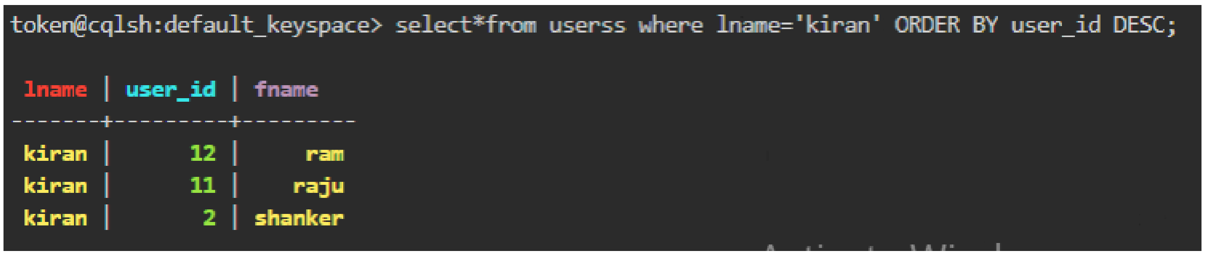
SELECT * FROM users WHERE lname = 'Smith';



```
token@cqlsh:default_keyspace> select*from users where lname='smith' allow filtering;

 user_id | fname | lname
---------+-------+-------
       1 |  john | smith
```

9.Display data from the users table sorted by lname in descending order.

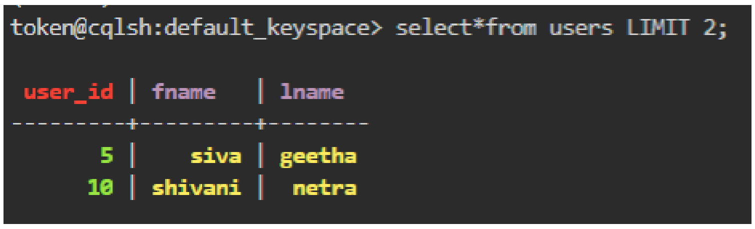
SELECT * FROM users ORDER BY lname DESC;



```
token@cqlsh:default_keyspace> select*from userss where lname='kiran' ORDER BY user_id DESC;

 lname | user_id | fname
-------+---------+---------
 kiran |      12 |     ram
 kiran |      11 |    raju
 kiran |       2 | shanker
```

10.Display the first two rows from the users table.

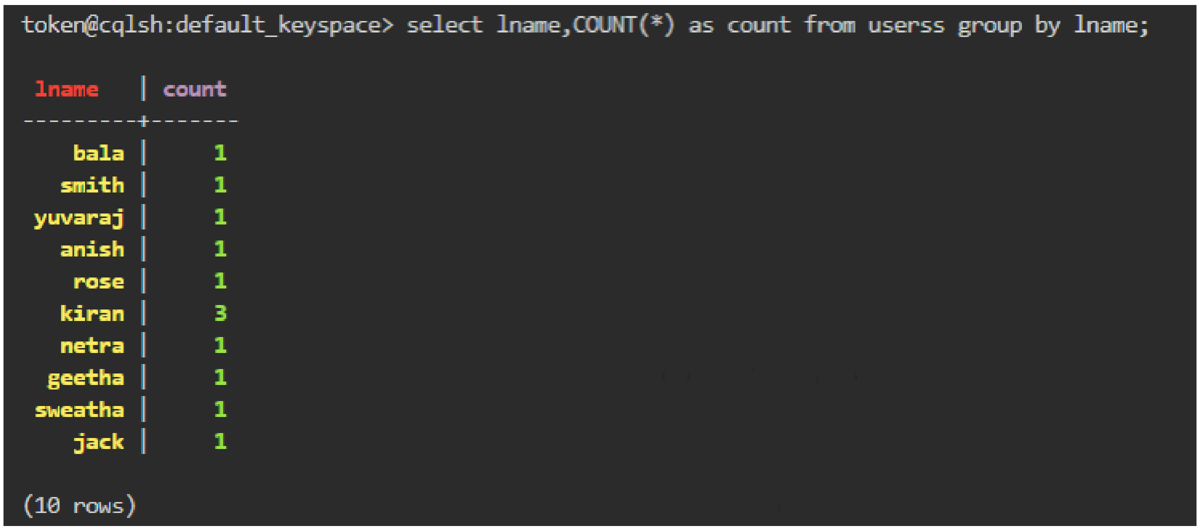
SELECT * FROM users LIMIT 2;



```
token@cqlsh:default_keyspace> select*from users LIMIT 2;

 user_id | fname   | lname
---------+---------+--------
       5 |    siva | geetha
      10 | shivani |  netra
```

11.Count the number of users with the same lname.

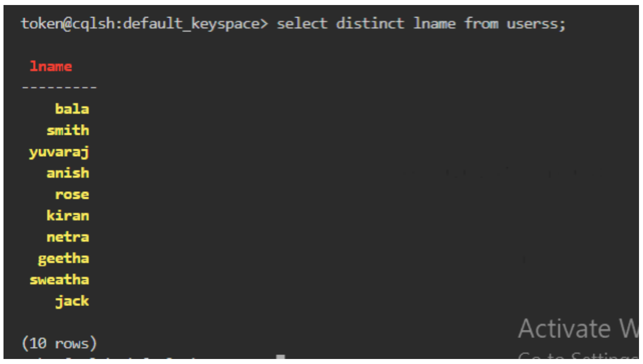
SELECT lname, COUNT(*) as count FROM users GROUP BY lname;



```
token@cqlsh:default_keyspace> select lname,COUNT(*) as count from userss group by lname;

 lname    | count
----------+-------
     bala |     1
    smith |     1
  yuvaraj |     1
    anish |     1
     rose |     1
    kiran |     3
    netra |     1
   geetha |     1
  sweatha |     1
     jack |     1

(10 rows)
```

12. Display unique last names from the users table.

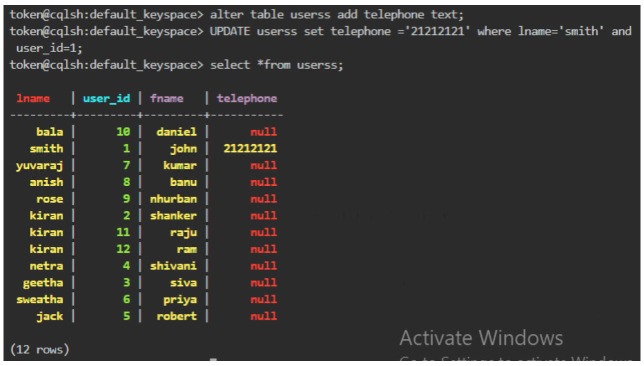
SELECT DISTINCT lname FROM users;



```
token@cqlsh:default_keyspace> select distinct lname from userss;

 lname
---------
    bala
   smith
 yuvaraj
   anish
    rose
   kiran
   netra
  geetha
 sweatha
    jack

(10 rows)
```

13. Update the telephone field for the user with user_id = 1745 to '21212121'.

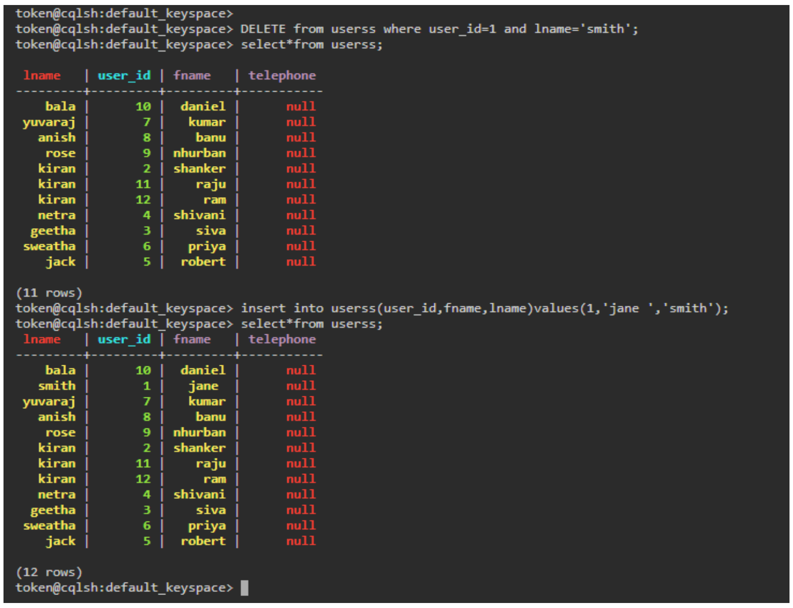
UPDATE users SET telephone = '21212121' WHERE user_id = 1745;



```
token@cqlsh:default_keyspace> alter table userss add telephone text;
token@cqlsh:default_keyspace> UPDATE userss set telephone ='21212121' where lname='smith' and
 user_id=1;
token@cqlsh:default_keyspace> select *from userss;

 lname   | user_id | fname   | telephone
---------+---------+---------+-----------
    bala |      10 |  daniel |      null
   smith |       1 |    john |  21212121
 yuvaraj |       7 |   kumar |      null
   anish |       8 |    banu |      null
    rose |       9 | nhurban |      null
   kiran |       2 | shanker |      null
   kiran |      11 |    raju |      null
   kiran |      12 |     ram |      null
   netra |       4 | shivani |      null
  geetha |       3 |    siva |      null
 sweatha |       6 |   priya |      null
    jack |       5 |  robert |      null

(12 rows)
```

14.Update both fname and lname of the user with user_id 1746.

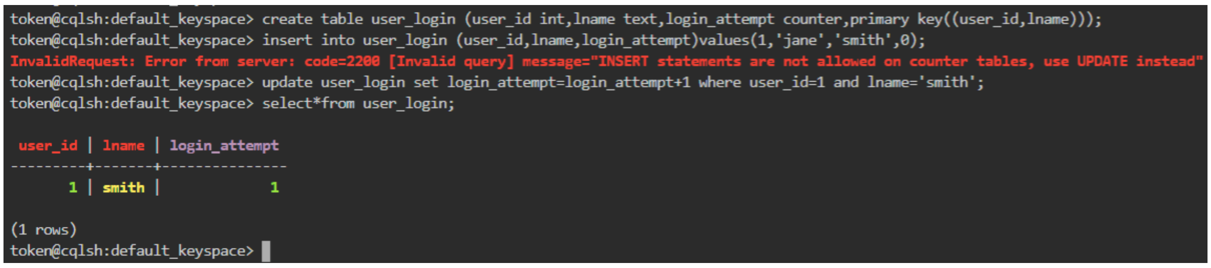
UPDATE users SET fname = 'Jane', lname = 'Smith' WHERE user_id = 1746;



```
token@cqlsh:default_keyspace>
token@cqlsh:default_keyspace> DELETE from userss where user_id=1 and lname='smith';
token@cqlsh:default_keyspace> select*from userss;

 lname   | user_id | fname   | telephone
---------+---------+---------+-----------
    bala |      10 |  daniel |      null
 yuvaraj |       7 |   kumar |      null
   anish |       8 |    banu |      null
    rose |       9 | nhurban |      null
   kiran |       2 | shanker |      null
   kiran |      11 |    raju |      null
   kiran |      12 |     ram |      null
   netra |       4 | shivani |      null
  geetha |       3 |    siva |      null
 sweatha |       6 |   priya |      null
    jack |       5 |  robert |      null

(11 rows)
token@cqlsh:default_keyspace> insert into userss(user_id,fname,lname)values(1,'jane ','smith');
token@cqlsh:default_keyspace> select*from userss;
 lname   | user_id | fname   | telephone
---------+---------+---------+-----------
    bala |      10 |  daniel |      null
   smith |       1 |   jane  |      null
 yuvaraj |       7 |   kumar |      null
   anish |       8 |    banu |      null
    rose |       9 | nhurban |      null
   kiran |       2 | shanker |      null
   kiran |      11 |    raju |      null
   kiran |      12 |     ram |      null
   netra |       4 | shivani |      null
  geetha |       3 |    siva |      null
 sweatha |       6 |   priya |      null
    jack |       5 |  robert |      null

(12 rows)
token@cqlsh:default_keyspace>
```

15. Increment the login_attempts counter for the user with user_id 1747 by 1.

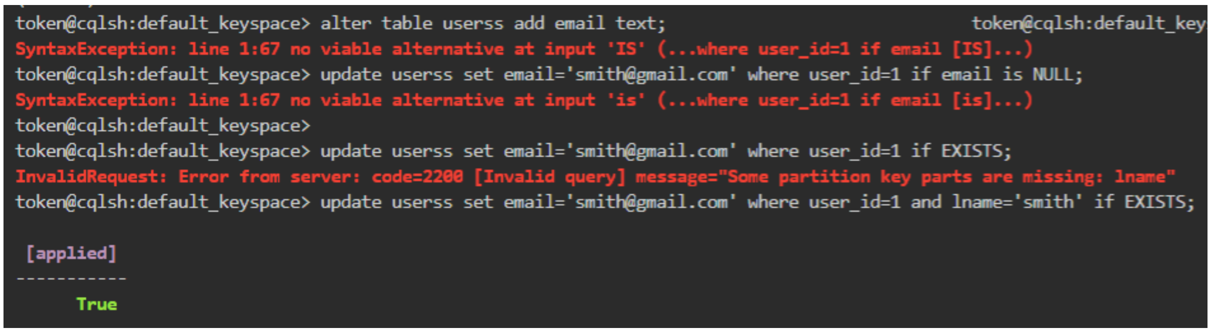
UPDATE users SET login_attempts = login_attempts + 1 WHERE user_id = 1747;



```
token@cqlsh:default_keyspace> create table user_login (user_id int,lname text,login_attempt counter,primary key((user_id,lname)));
token@cqlsh:default_keyspace> insert into user_login (user_id,lname,login_attempt)values(1,'jane','smith',0);
InvalidRequest: Error from server: code=2200 [Invalid query] message="INSERT statements are not allowed on counter tables, use UPDATE instead"
token@cqlsh:default_keyspace> update user_login set login_attempt=login_attempt+1 where user_id=1 and lname='smith';
token@cqlsh:default_keyspace> select*from user_login;

 user_id | lname | login_attempt
---------+-------+---------------
       1 | smith |             1

(1 rows)
token@cqlsh:default_keyspace>
```

16.Update the email of the user with user_id 1748 only if it's currently null.

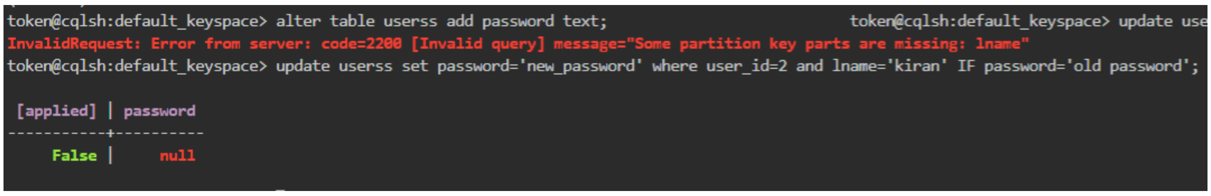
UPDATE users SET email = 'example@example.com' WHERE user_id = 1748 IF email IS NULL;



```
token@cqlsh:default_keyspace> alter table userss add email text;                    token@cqlsh:default_key
SyntaxException: line 1:67 no viable alternative at input 'IS' (...where user_id=1 if email [IS]...)
token@cqlsh:default_keyspace> update userss set email='smith@gmail.com' where user_id=1 if email is NULL;
SyntaxException: line 1:67 no viable alternative at input 'is' (...where user_id=1 if email [is]...)
token@cqlsh:default_keyspace>
token@cqlsh:default_keyspace> update userss set email='smith@gmail.com' where user_id=1 if EXISTS;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Some partition key parts are missing: lname"
token@cqlsh:default_keyspace> update userss set email='smith@gmail.com' where user_id=1 and lname='smith' if EXISTS;

 [applied]
-----------
      True
```

17.Update the password of the user with user_id 1745 only if the current password matches a specific value.

UPDATE users SET password = 'new_password' WHERE user_id = 1745 IF password = 'old_password';



```
token@cqlsh:default_keyspace> alter table userss add password text;                    token@cqlsh:default_keyspace> update use
InvalidRequest: Error from server: code=2200 [Invalid query] message="Some partition key parts are missing: lname"
token@cqlsh:default_keyspace> update userss set password='new_password' where user_id=2 and lname='kiran' IF password='old password';

 [applied] | password
-----------+----------
     False |     null
```

18. Use different types of data modification operations within a single batch operation in Cassandra.

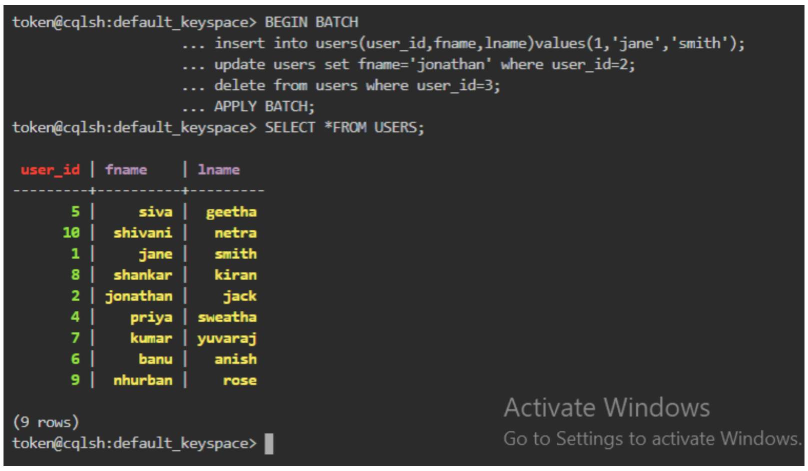
BEGIN BATCH

INSERT INTO users (user_id, fname, lname) VALUES (1745, 'John', 'Smith');

UPDATE users SET fname = 'Jonathan' WHERE user_id = 1746;

DELETE FROM users WHERE user_id = 1747;

APPLY BATCH;



```
token@cqlsh:default_keyspace> BEGIN BATCH
                          ... insert into users(user_id,fname,lname)values(1,'jane','smith');
                          ... update users set fname='jonathan' where user_id=2;
                          ... delete from users where user_id=3;
                          ... APPLY BATCH;
token@cqlsh:default_keyspace> SELECT *FROM USERS;

 user_id | fname    | lname
---------+----------+---------
       5 |     siva |  geetha
      10 |  shivani |   netra
       1 |     jane |   smith
       8 |  shankar |   kiran
       2 | jonathan |    jack
       4 |    priya | sweatha
       7 |    kumar | yuvaraj
       6 |     banu |   anish
       9 |  nhurban |    rose

(9 rows)
token@cqlsh:default_keyspace> 
```

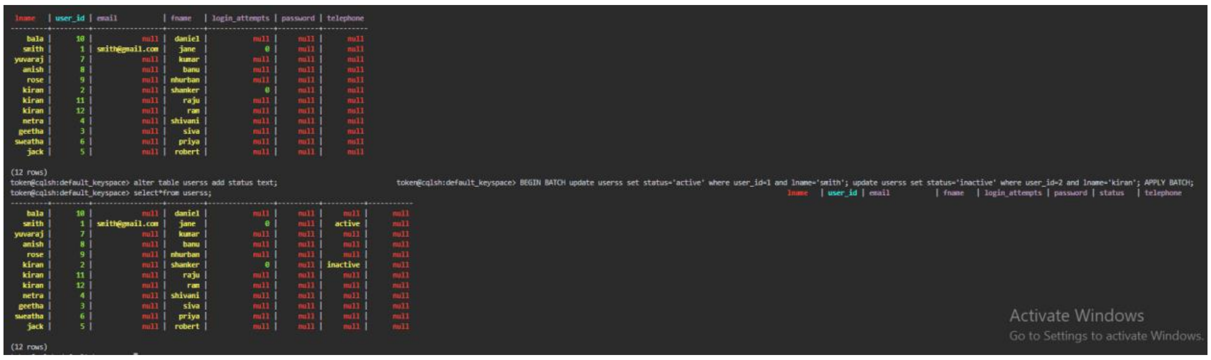
19.Update multiple rows in a batch operation.

BEGIN BATCH

UPDATE users SET status = 'active' WHERE user_id = 1745;

UPDATE users SET status = 'inactive' WHERE user_id = 1746;

APPLY BATCH;



```
 lname   | user_id | email           | fname   | login_attempts | password | telephone
---------+---------+-----------------+---------+----------------+----------+-----------
    bala |      10 |            null |  daniel |           null |     null |      null
   smith |       1 | smith@gmail.com |   jane  |              0 |     null |      null
 yuvaraj |       7 |            null |   kumar |           null |     null |      null
   anish |       8 |            null |    banu |           null |     null |      null
    rose |       9 |            null | nhurban |           null |     null |      null
   kiran |       2 |            null | shanker |              0 |     null |      null
   kiran |      11 |            null |    raju |           null |     null |      null
   kiran |      12 |            null |     ram |           null |     null |      null
   netra |       4 |            null | shivani |           null |     null |      null
  geetha |       3 |            null |    siva |           null |     null |      null
 sweatha |       6 |            null |   priya |           null |     null |      null
    jack |       5 |            null |  robert |           null |     null |      null

(12 rows)
token@cqlsh:default_keyspace> alter table userss add status text;                    token@cqlsh:default_keyspace> BEGIN BATCH update userss set status='active' where user_id=1 and lname='smith'; update userss set status='inactive' where user_id=2 and lname='kiran'; APPLY BATCH;
token@cqlsh:default_keyspace> select*from userss;                                                                     lname   | user_id | email           | fname   | login_attempts | password | status   | telephone
---------+---------+-----------------+---------+----------------+----------+----------+-----------
    bala |      10 |            null |  daniel |           null |     null |     null |      null
   smith |       1 | smith@gmail.com |   jane  |              0 |     null |   active |      null
 yuvaraj |       7 |            null |   kumar |           null |     null |     null |      null
   anish |       8 |            null |    banu |           null |     null |     null |      null
    rose |       9 |            null | nhurban |           null |     null |     null |      null
   kiran |       2 |            null | shanker |              0 |     null | inactive |      null
   kiran |      11 |            null |    raju |           null |     null |     null |      null
   kiran |      12 |            null |     ram |           null |     null |     null |      null
   netra |       4 |            null | shivani |           null |     null |     null |      null
  geetha |       3 |            null |    siva |           null |     null |     null |      null
 sweatha |       6 |            null |   priya |           null |     null |     null |      null
    jack |       5 |            null |  robert |           null |     null |     null |      null

(12 rows)
```

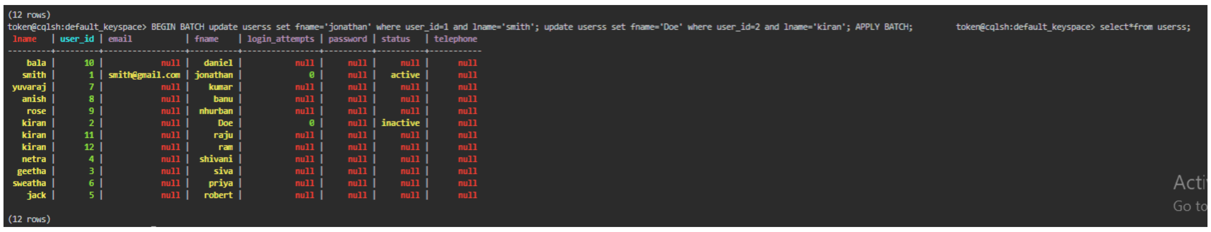
20. Perform conditional updates on multiple rows using a batch operation

BEGIN BATCH

UPDATE users SET fname = 'Jonathan' WHERE user_id = 1745 IF lname = 'Smith';

UPDATE users SET lname = 'Doe' WHERE user_id = 1746 IF fname = 'Jane';

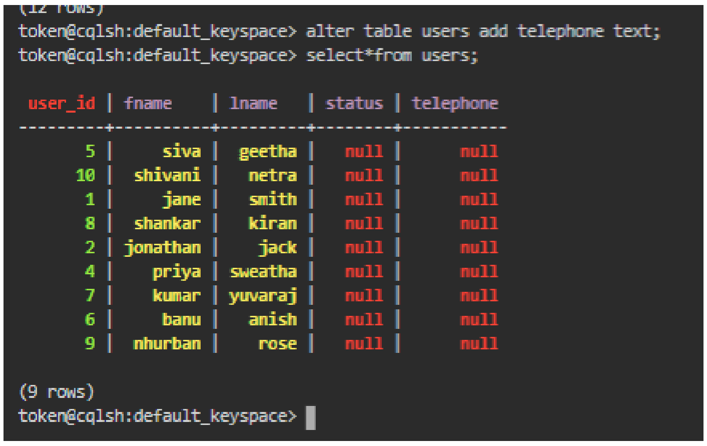
APPLY BATCH;



```
(12 rows)
token@cqlsh:default_keyspace> BEGIN BATCH update userss set fname='jonathan' where user_id=1 and lname='smith'; update userss set fname='Doe' where user_id=2 and lname='kiran'; APPLY BATCH;        token@cqlsh:default_keyspace> select*from userss;
 lname   | user_id | email           | fname    | login_attempts | password | status   | telephone
---------+---------+-----------------+----------+----------------+----------+----------+-----------
    bala |      10 |            null |   daniel |           null |     null |     null |      null
   smith |       1 | smith@gmail.com | jonathan |              0 |     null |   active |      null
 yuvaraj |       7 |            null |    kumar |           null |     null |     null |      null
   anish |       8 |            null |     banu |           null |     null |     null |      null
    rose |       9 |            null |  nhurban |           null |     null |     null |      null
   kiran |       2 |            null |      Doe |              0 |     null | inactive |      null
   kiran |      11 |            null |     raju |           null |     null |     null |      null
   kiran |      12 |            null |      ram |           null |     null |     null |      null
   netra |       4 |            null |  shivani |           null |     null |     null |      null
  geetha |       3 |            null |     siva |           null |     null |     null |      null
 sweatha |       6 |            null |    priya |           null |     null |     null |      null
    jack |       5 |            null |   robert |           null |     null |     null |      null

(12 rows)
```

21. Add a new field named telephone to the users table.

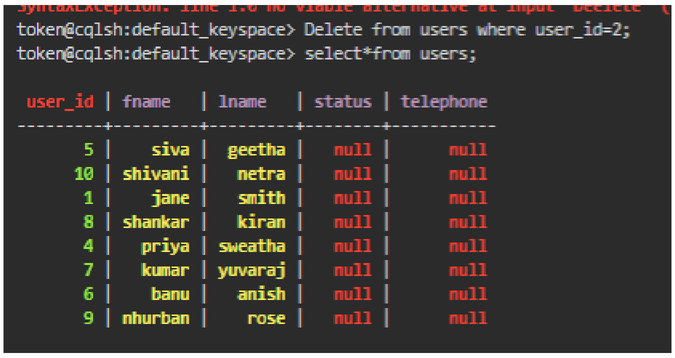
ALTER TABLE users ADD telephone text;



```
(12 rows)
token@cqlsh:default_keyspace> alter table users add telephone text;
token@cqlsh:default_keyspace> select*from users;

 user_id | fname    | lname    | status | telephone
---------+----------+----------+--------+-----------
       5 |     siva |   geetha |   null |      null
      10 |  shivani |   netra  |   null |      null
       1 |     jane |    smith |   null |      null
       8 |  shankar |    kiran |   null |      null
       2 | jonathan |     jack |   null |      null
       4 |    priya |  sweatha |   null |      null
       7 |    kumar |  yuvaraj |   null |      null
       6 |     banu |    anish |   null |      null
       9 |  nhurban |     rose |   null |      null

(9 rows)
token@cqlsh:default_keyspace>
```

22.Delete the user with user_id 1745 from the users table.

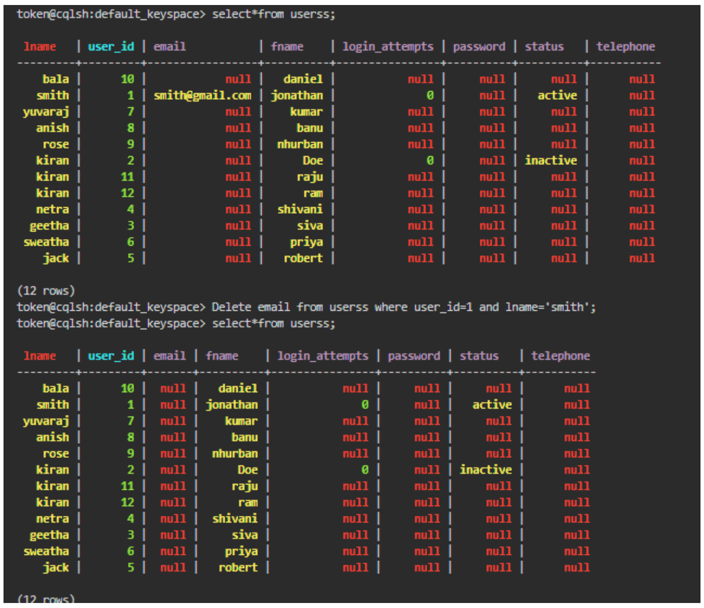
DELETE FROM users WHERE user_id = 1745;



```
SyntaxException: line 1:0 no viable alternative at input 'Deelete' (
token@cqlsh:default_keyspace> Delete from users where user_id=2;
token@cqlsh:default_keyspace> select*from users;

 user_id | fname   | lname   | status | telephone
---------+---------+---------+--------+-----------
       5 |    siva |  geetha |   null |      null
      10 | shivani |   netra |   null |      null
       1 |    jane |   smith |   null |      null
       8 | shankar |   kiran |   null |      null
       4 |   priya | sweatha |   null |      null
       7 |   kumar | yuvaraj |   null |      null
       6 |    banu |   anish |   null |      null
       9 | nhurban |    rose |   null |      null
```

23. Delete the email column value for the user with user_id 1746.

DELETE email FROM users WHERE user_id = 1746;



```
token@cqlsh:default_keyspace> select*from userss;

 lname   | user_id | email           | fname    | login_attempts | password | status   | telephone
---------+---------+-----------------+----------+----------------+----------+----------+-----------
    bala |      10 |            null |   daniel |           null |     null |     null |      null
   smith |       1 | smith@gmail.com | jonathan |              0 |     null |   active |      null
 yuvaraj |       7 |            null |    kumar |           null |     null |     null |      null
   anish |       8 |            null |     banu |           null |     null |     null |      null
    rose |       9 |            null |  nhurban |           null |     null |     null |      null
   kiran |       2 |            null |      Doe |              0 |     null | inactive |      null
   kiran |      11 |            null |     raju |           null |     null |     null |      null
   kiran |      12 |            null |      ram |           null |     null |     null |      null
   netra |       4 |            null |  shivani |           null |     null |     null |      null
  geetha |       3 |            null |     siva |           null |     null |     null |      null
 sweatha |       6 |            null |    priya |           null |     null |     null |      null
    jack |       5 |            null |   robert |           null |     null |     null |      null

(12 rows)
token@cqlsh:default_keyspace> Delete email from userss where user_id=1 and lname='smith';
token@cqlsh:default_keyspace> select*from userss;

 lname   | user_id | email | fname    | login_attempts | password | status   | telephone
---------+---------+-------+----------+----------------+----------+----------+-----------
    bala |      10 |  null |   daniel |           null |     null |     null |      null
   smith |       1 |  null | jonathan |              0 |     null |   active |      null
 yuvaraj |       7 |  null |    kumar |           null |     null |     null |      null
   anish |       8 |  null |     banu |           null |     null |     null |      null
    rose |       9 |  null |  nhurban |           null |     null |     null |      null
   kiran |       2 |  null |      Doe |              0 |     null | inactive |      null
   kiran |      11 |  null |     raju |           null |     null |     null |      null
   kiran |      12 |  null |      ram |           null |     null |     null |      null
   netra |       4 |  null |  shivani |           null |     null |     null |      null
  geetha |       3 |  null |     siva |           null |     null |     null |      null
 sweatha |       6 |  null |    priya |           null |     null |     null |      null
    jack |       5 |  null |   robert |           null |     null |     null |      null

(12 rows)
```

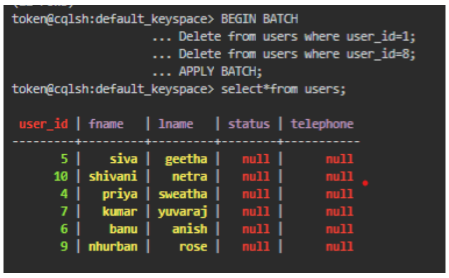
24.Delete multiple users in a batch operation.

BEGIN BATCH

DELETE FROM users WHERE user_id = 1747;

DELETE FROM users WHERE user_id = 1748;

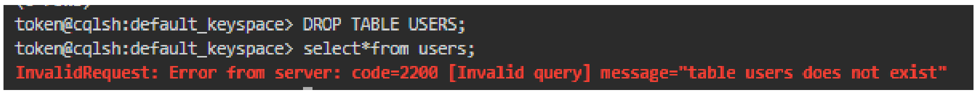
APPLY BATCH;



```
token@cqlsh:default_keyspace> BEGIN BATCH
                   ... Delete from users where user_id=1;
                   ... Delete from users where user_id=8;
                   ... APPLY BATCH;
token@cqlsh:default_keyspace> select*from users;

 user_id | fname   | lname   | status | telephone
---------+---------+---------+--------+-----------
       5 |    siva |  geetha |   null |      null
      10 | shivani |   netra |   null |      null
       4 |   priya | sweatha |   null |      null
       7 |   kumar | yuvaraj |   null |      null
       6 |    banu |   anish |   null |      null
       9 | nhurban |    rose |   null |      null
```

25. Remove the users table.

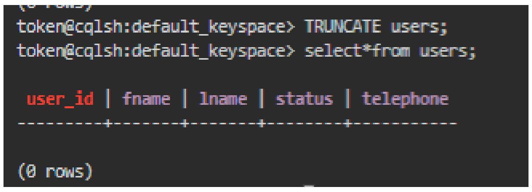
DROP TABLE users;



```
token@cqlsh:default_keyspace> DROP TABLE USERS;
token@cqlsh:default_keyspace> select*from users;
InvalidRequest: Error from server: code=2200 [Invalid query] message="table users does not exist"
```

26. Remove all data from the users table.

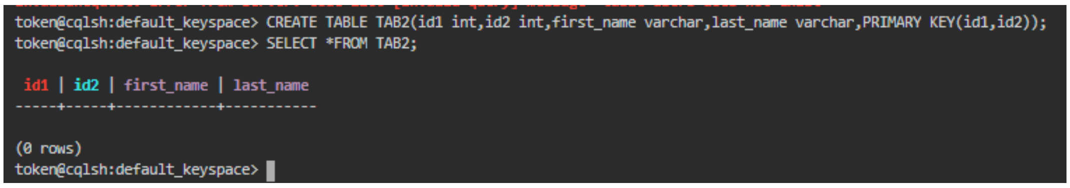
TRUNCATE users;



```
token@cqlsh:default_keyspace> TRUNCATE users;
token@cqlsh:default_keyspace> select*from users;

 user_id | fname | lname | status | telephone
---------+-------+-------+--------+-----------

(0 rows)
```

27. Create a table with a composite key

CREATE TABLE tab2 (id1 int, id2 int, first_name varchar, last_name varchar, PRIMARY KEY (id1, id2));



```
token@cqlsh:default_keyspace> CREATE TABLE TAB2(id1 int,id2 int,first_name varchar,last_name varchar,PRIMARY KEY(id1,id2));
token@cqlsh:default_keyspace> SELECT *FROM TAB2;

 id1 | id2 | first_name | last_name
-----+-----+------------+-----------

(0 rows)
token@cqlsh:default_keyspace> 
```

<u>Result:</u>

Successfully executed Cassandra table and batch operations, enhancing skills in data manipulation and querying.