

Big Data Analytics (2CS702)

Practical 9	
Rollno: 19BCE055	Name: TEJAS DOBARIYA
Division: A	Batch: E1

Aim: Setup Cassandra environment in your system and apply Create, Update, Read and Delete operations.

Setting up Environment:

Try It Out: Cassandra Query Language (CQL):

STEP 1 OF 7

Create a keyspace

Let's first start learning CQL by creating a keyspace, using the `CREATE KEYSPACE` command.

```
CREATE KEYSPACE demo WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
```

A keyspace is a way to logically group a collection of database objects together, such as:

- tables
- user-defined types
- user-defined functions
- and more!

Terminal

```
$ sleep 3; wait.sh  
Starting up Cassandra... [\\]
```

Setting Up Key Spaces:

```
Your Interactive Bash Terminal.
$ cqlsh
Connected to Cassandra Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 4.0-beta4 | CQL spec 3.4.5 | Native protocol v4]
Use HELP for help.
cqlsh>
cqlsh>
cqlsh> CREATE KEYSPACE demo WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> describe keyspaces

demo      system_auth      system_schema    system_views
system    system_distributed system_traces     system_virtual_schema

cqlsh> 
```

The command is used to describe all the keyspaces in the Cassandra cluster. Cassandra automatically creates keyspaces that have names starting with system and uses these as the data dictionary for the cluster.

Creating Tables and Inserting Data:

```
token@cqlsh:bda> CREATE TABLE IF NOT EXISTS students (
... email TEXT,
... name TEXT ,
... password TEXT,
... user_id UUID,
... PRIMARY KEY (( email ))
... );
token@cqlsh:bda> select * from students;

email | name | password | user_id
-----+-----+-----+-----
```

View Table Data:

```
token@cqlsh:bda> insert into students (email,name,password,user_id)
... values ('tejas@gmail.com','tejas','1234567',11111111-1111-1111-1111-111111111111);
token@cqlsh:bda> select * from students;
```

email	name	password	user_id
tejas@gmail.com	tejas	1234567	11111111-1111-1111-1111-111111111111

(1 rows)

```
token@cqlsh:bda> insert into students (email,name,password,user_id)
... values ('sachin@gmail.com','sachin','123456',22222222-2222-2222-2222-222222222222);
token@cqlsh:bda> select * from students;
```

email	name	password	user_id
tejas@gmail.com	tejas	1234567	11111111-1111-1111-1111-111111111111
sachin@gmail.com	sachin	123456	22222222-2222-2222-2222-222222222222

(2 rows)

Update Data:

```
token@cqlsh:bda> update students set name = 'ayush' where email = 'sachin@gmail.com';
token@cqlsh:bda> select * from students;
```

email	name	password	user_id
tejas@gmail.com	tejas	1234567	11111111-1111-1111-1111-111111111111
sachin@gmail.com	ayush	123456	22222222-2222-2222-2222-222222222222

(2 rows)

Delete Data and View Changes:

```

token@cqlsh:bda> delete from students where email = 'sachin@gmail.com';
token@cqlsh:bda> select * from students;

email          | name | password | user_id
-----+-----+-----+-----
tejas@gmail.com | tejas | 1234567  | 11111111-1111-1111-1111-111111111111

(1 rows)
token@cqlsh:bda> delete name from students where email = 'tejas@gmail.com';
token@cqlsh:bda> select * from students;

email          | name | password | user_id
-----+-----+-----+-----
tejas@gmail.com | null | 1234567  | 11111111-1111-1111-1111-111111111111

(1 rows)
token@cqlsh:bda> 

```