



2CS702 - Big Data Analytics

Practical 9

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

Author: Darshil Maru 20BCE514

Guide: Dr. Purnima Gandhi

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

Create Keyspace:-

- Syntax:- CREATE KEYSPACE keyspace_name WITH replication = {'class': 'SimpleStrategy', 'replication_factor': n};
- The replication map determines how many copies of the data are kept in a given data centre.
- 'SimpleStrategy': Assigns the same replication factor to the entire cluster. Use for evaluation and single data center test and development environments only.

```
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 keyspace lists all the keyspaces available in that particular cluster.

```
cqlsh> DESCRIBE KEYSPACES;

demo      system_auth      system_schema      system_views
system     system_distributed system_traces      system_virtual_schema

cqlsh>
```

CREATE Operation:-

- Syntax:- CREATE TABLE [IF NOT EXISTS] [keyspace_name.] table_name(column_definition [] PRIMARY KEY (column_name, [column_name]));

```
cqlsh> CREATE TABLE demo.users(lastname text PRIMARY KEY, firstname text, email text);
cqlsh>
```

- In our case, keyspace_name is demo, and the table_name is users.
- The column name is lastname, firstname, email and datatype is text.
- PRIMARY KEY is lastname as a single primary key.

INSERT Operation:-

- Syntax:- INSERT INTO [keyspace_name.] table_name (column_list) VALUES (column_values);

```
cqlsh:demo> INSERT INTO users(lastname, firstname, email) VALUES('John', 'Cena', 'johncena@xyz.com');
cqlsh:demo> INSERT INTO users(lastname, firstname, email) VALUES('Shane', 'Watson', 'shanewatson@xyz.com');
cqlsh:demo> INSERT INTO users(lastname, firstname, email) VALUES('Cristiano', 'Ronaldo', 'cristianor@xyz.com');
```

- SELECT * FROM table_name;

```
cqlsh:demo> SELECT * FROM users;
```

lastname	email	firstname
Shane	shanewatson@xyz.com	Watson
Audly	sandie@gmail.com	Sandie
Fry	vivienne@gmail.com	Vivienne
Phi	krystle@gmail.com	Krystle
John	johncena@xyz.com	Cena
Cimbura	trudie@gmail.com	Trudie
Karna	zia@gmail.com	Zia
Ursulette	aeriela@gmail.com	Aeriela
Timon	nananne@gmail.com	Nananne
Callista	katleen@gmail.com	Katleen
Cristiano	cristianor@xyz.com	Ronaldo
Harl	gilda@gmail.com	Gilda

(12 rows)

cqlsh:demo>

Powered by **Kata**oda

- SELECT * FROM table_name WHERE column_name='value';

```
cqlsh:demo> SELECT * FROM users WHERE lastname='Karna';
```

lastname	email	firstname
Karna	zia@gmail.com	Zia

(1 rows)

cqlsh:demo>

Powered by **Kata**oda

UPDATE Operation:-

- Syntax:- UPDATE table_name SET column_name = 'value' WHERE column_name = 'value';

```
cqlsh:demo> UPDATE users SET email='karnazia@xyz.com' WHERE lastname='Karna';
cqlsh:demo> SELECT * FROM users WHERE lastname='Karna';
```

lastname	email	firstname
Karna	karnazia@xyz.com	Zia

(1 rows)

cqlsh:demo>

Powered by **Kata**oda

DELETE Operation:-

- Syntax:- DELETE FROM table_name WHERE column_name='value';

```
cqlsh:demo> DELETE FROM users WHERE lastname='Karna';
```

cqlsh:demo>

Powered by **Kata**oda