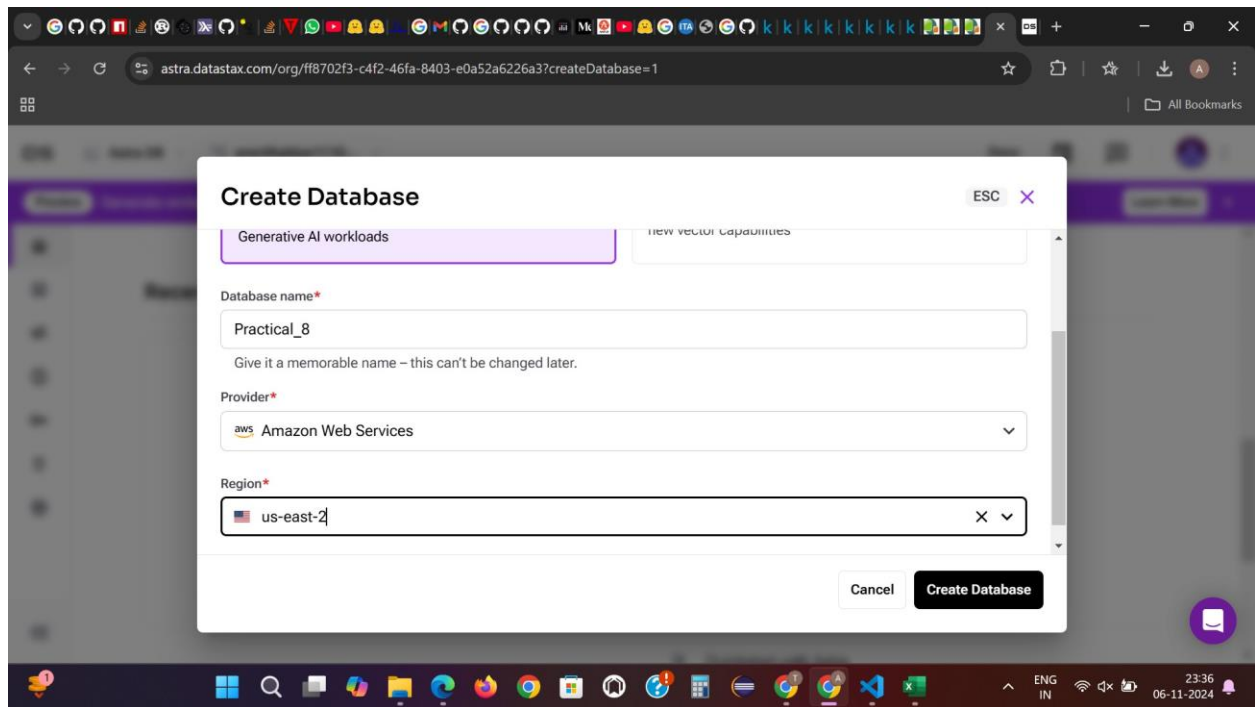


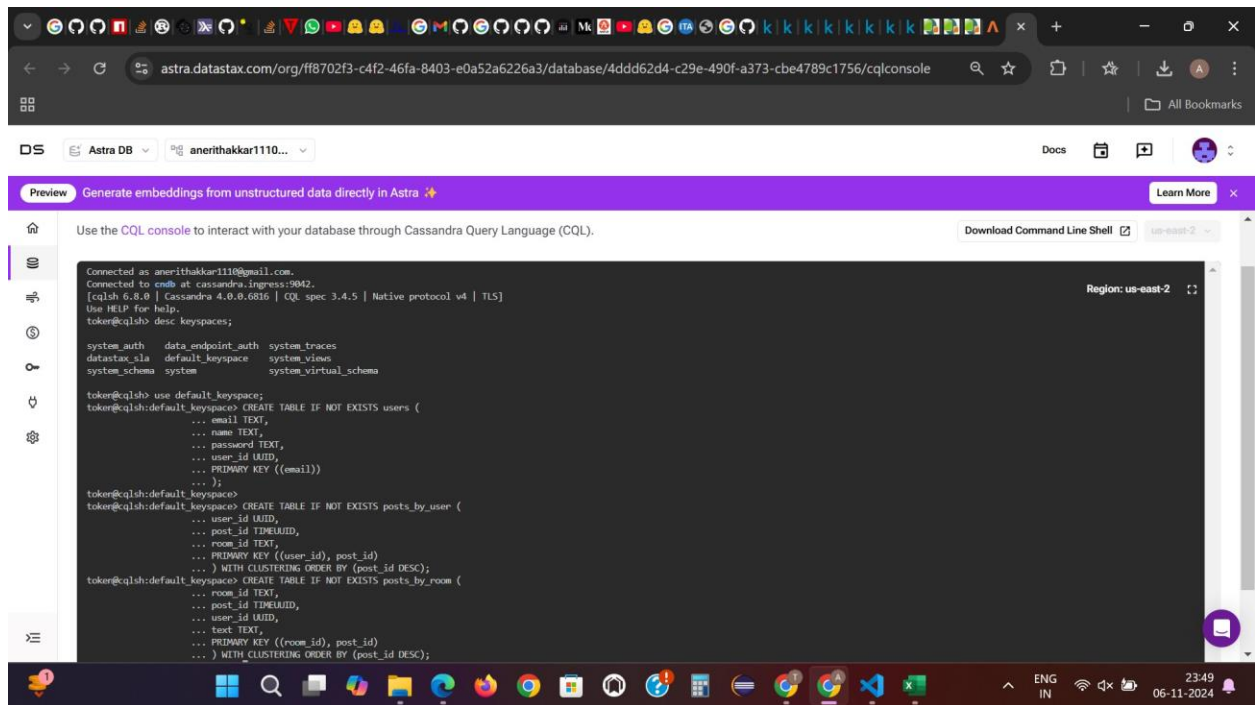
Roll No.: 21BCE037

Name: Chaudhari Rushali Bhagvan

Practical: 8

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





CRUDOperations:

- Table 1

Create table

```
token@cqlsh> use default_keyspace;
token@cqlsh:default_keyspace> CREATE TABLE IF NOT EXISTS users (
    ...     email TEXT,
    ...     name TEXT,
    ...     password TEXT,
    ...     user_id UUID,
    ...     PRIMARY KEY ((email))
    ... );
```

Insert values and read operation

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

email	name	password	user_id
(0 rows)			

```
token@cqlsh:default_keyspace> INSERT INTO users (email, name, password, user_id)
... VALUES ('aneri@example.com', 'aneri', 'nirma123', uuid());
token@cqlsh:default_keyspace> select * from users;
```

email	name	password	user_id
aneri@example.com	aneri	nirma123	b72850d6-b1e9-45a1-93cd-e145220eafe5

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

Update operation

```
token@cqlsh:default_keyspace> UPDATE users
... SET name = 'prachi', password = 'krishna123'
... WHERE email = 'aneri@example.com';
token@cqlsh:default_keyspace> select *from users;
```

email	name	password	user_id
aneri@example.com	prachi	krishna123	b72850d6-b1e9-45a1-93cd-e145220eafe5

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

Delete operation

```
token@cqlsh:default_keyspace> DELETE FROM users
... WHERE email = 'aneri@example.com';
token@cqlsh:default_keyspace> select *from users;
```

email	name	password	user_id
(0 rows)			

```
token@cqlsh:default_keyspace>
```

- Table 2

Create table

```
token@cqlsh:default_keyspace> CREATE TABLE IF NOT EXISTS posts_by_user (
...     user_id UUID,
...     post_id TIMEUUID,
...     room_id TEXT,
...     text TEXT,
...     PRIMARY KEY ((user_id), post_id)
... ) WITH CLUSTERING ORDER BY (post_id DESC);
```

Insert values and read operation

```
token@cqlsh:default_keyspace> INSERT INTO posts_by_user (user_id, post_id, room_id, text)
... VALUES (uuid(), now(), 'Room123', 'This is my first post in Room123.');
```

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

user_id	post_id	room_id	text
479ca6e0-7c18-432e-aa22-5e10ec567d29	31d7a460-8f7b-11ef-ae60-0daab57b1e56	Room123	This is my first post in Room123.

(1 rows)

Update operation

```
token@cqlsh:default_keyspace> UPDATE posts_by_user
... SET text = 'This is an updated post.'
... WHERE user_id = 479ca6e0-7c18-432e-aa22-5e10ec567d29
... AND post_id = 31d7a460-8f7b-11ef-ae60-0daab57b1e56;
```

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

user_id	post_id	room_id	text
479ca6e0-7c18-432e-aa22-5e10ec567d29	31d7a460-8f7b-11ef-ae60-0daab57b1e56	Room123	This is an updated post.

(1 rows)

Delete operation

```
token@cqlsh:default_keyspace> DELETE FROM posts_by_user
... WHERE user_id = 479ca6e0-7c18-432e-aa22-5e10ec567d29
... AND post_id = 31d7a460-8f7b-11ef-ae60-0daab57b1e56;
```

```
token@cqlsh:default_keyspace> select 8 from posts_by_user;
```

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot infer type for term 8 in selection clause (try using a cast to force a type)"
```

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

user_id	post_id	room_id	text
---------	---------	---------	------

- Table 3

Create table

```
token@cqlsh:default_keyspace> CREATE TABLE IF NOT EXISTS posts_by_room (
...     room_id TEXT,
...     post_id TIMEUUID,
...     user_id UUID,
...     text TEXT,
...     PRIMARY KEY ((room_id), post_id)
... ) WITH CLUSTERING ORDER BY (post_id DESC);
```

Insert values and read operation

```
token@cqlsh:default_keyspace> INSERT INTO posts_by_room (room_id, post_id, user_id, text)
... VALUES ('Room123', now(), uuid(), 'This is my first post in Room123.');
```

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

room_id	post_id	text	user_id
Room123	e2d521b0-8f7c-11ef-ae60-0daab57b1e56	This is my first post in Room123.	2d22b4cb-07ce-42aa-b757-481bc29b2c35

(1 rows)

Update operation

```
token@cqlsh:default_keyspace> UPDATE posts_by_room
... SET text = 'This is an updated post in Room123.'
... WHERE room_id = 'Room123' AND post_id = e2d521b0-8f7c-11ef-ae60-0daab57b1e56;
```

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

room_id	post_id	text	user_id
Room123	e2d521b0-8f7c-11ef-ae60-0daab57b1e56	This is an updated post in Room123.	2d22b4cb-07ce-42aa-b757-481bc29b2c35

(1 rows)

Delete operation

```
token@cqlsh:default_keyspace> DELETE FROM posts_by_room
... WHERE room_id = 'Room123' AND post_id = e2d521b0-8f7c-11ef-ae60-0daab57b1e56;
```

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

room_id	post_id	text	user_id
---------	---------	------	---------

(0 rows)