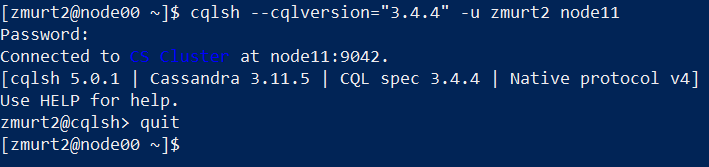
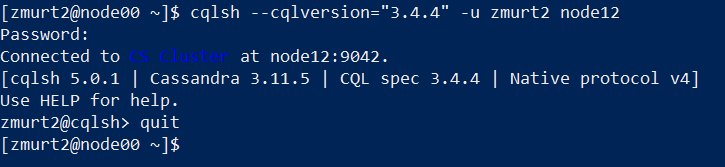
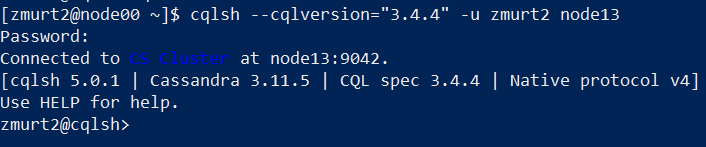
**Assignment 1:**

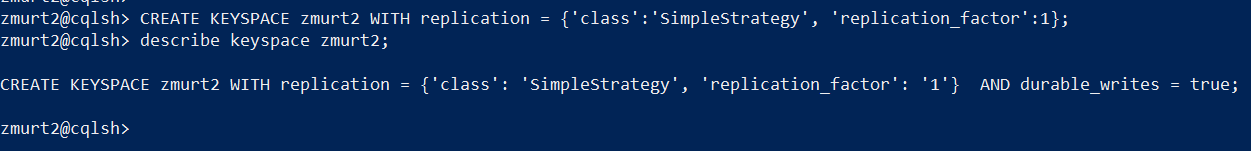






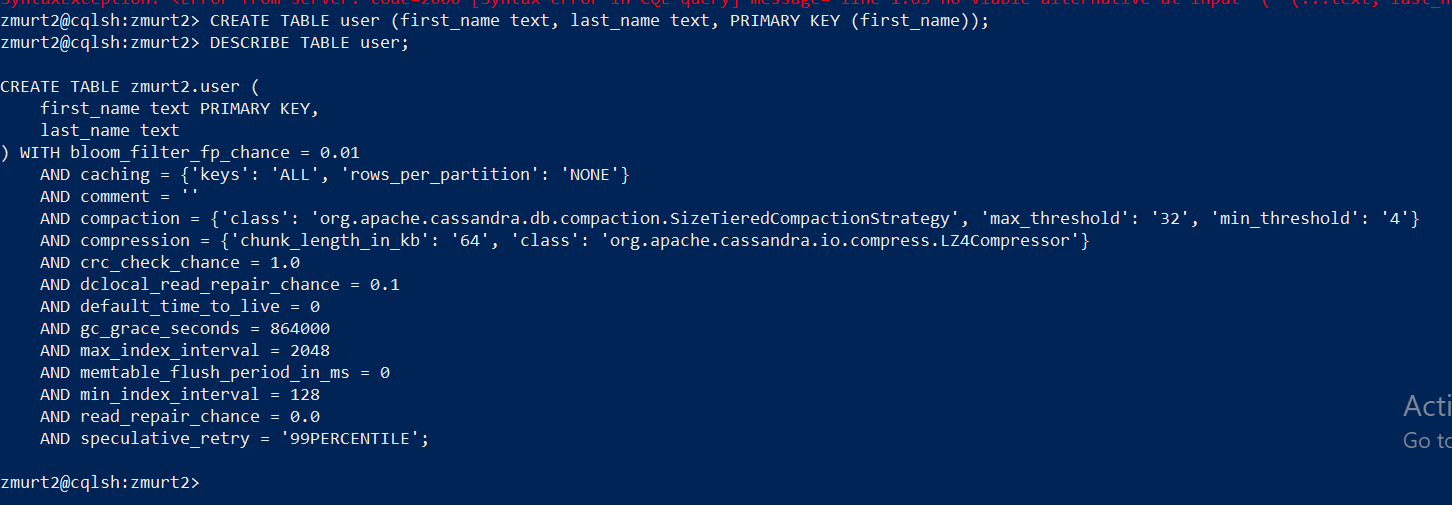
**Assignment 2:**





**Assignment 3 is on the next page.**

**Assignment 3:**



**Assignment 4:**

Cassandra, CQL (Cassandra Query Language) is often confused with SQL and RDBMS, but their difference is significant. Running the query that would expect to return all values from the table that contain my last name, “Murtaza”, will only give the error:

“InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"

The error received signifies that the Cassandra table indexes only the column that is listed in the “primary key” clause. Using “where” clause to search for a value, one needs to explicitly define the column in question in the primary key clause.

**Assignment 5:**

Using a **secondary index**  may serve as a workaround. Secondary Indexes are indexes that would be created to amalgamate with the primary index, which would help with searching with the use of various attributes. We can therefore search ‘first\_name’ along with ‘last\_name’, nothing that ‘first\_name’ is the primary key, and the query would henceforth become:

create index user\_last\_name on user(last\_name);

This would instantiate an index, named ‘user\_last\_name’ which will aid in searching values that use only ‘last\_name’. Then, we can use:

select \* from user where last\_name = 'Murtaza';

**Screenshot to prove this is on the next page:**

