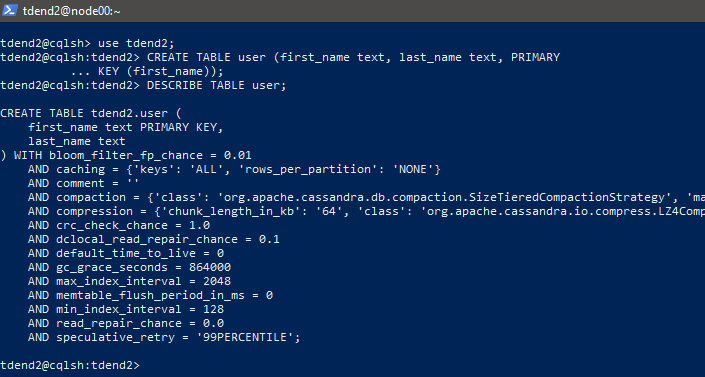
**NOSQL-CFDB-Cassandra**

**3.3. Create a table**

Create a simple table, ‘user’, with two columns, ‘first\_name’ and ‘last\_name’. ‘first\_name’ is a primary key in this table.

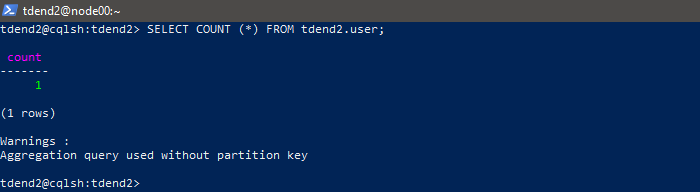


Insert a row into the table with your first name and last name



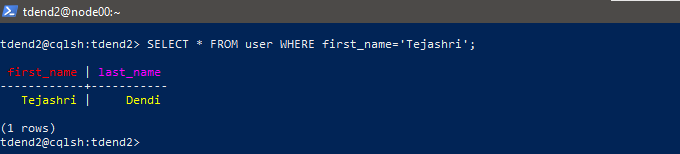
------------------------------------------------------------------------------------------------------------------------------------------

3.4. Querying  
A simple aggregation query.

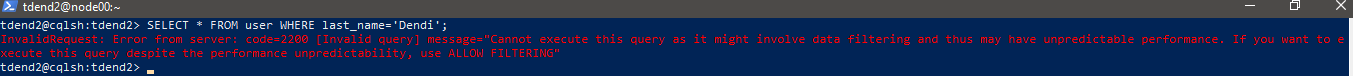


We have only one row in the table.  
Query- with the ‘WHERE’ clause using the first\_name column.

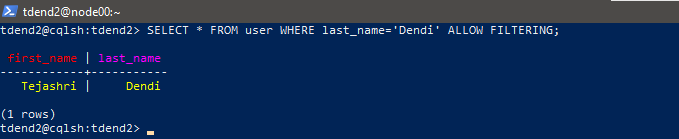
SELECT \* FROM user WHERE first\_name='Tejashri';



Used the last\_name column:



Added “ALLOW FILTERING” at the end of the query. However, it’s not recommended because of performance issues.



**Assignment** #**3** (20pts): In Section 3.4, Surprisingly, we got an error message when you used the

last\_name column in the WHERE clause. Explain why this happens.Hint: see your previous ‘create table’ command

**Solution**:

‘last\_name’ is not a primary key and a non-indexed column. . Also partition key is not defined. So, Cassandra cannot guarantee predictable performance for this query, especially on large datasets. But ‘ALLOW FILTERING’ may result in slower performance, especially as the dataset grows.

**Assignment #4 (20pts):** Let’s assume that the table, user, has a massive number of rows in it,

and the column is heavily used in many queries. So, we should design the table to allow users to

use the column ‘last\_name’ in the ‘where’ clause. Explain how we can solve this problem. Why

‘Allow filtering’ is not the best solution?Note: ‘Allow filtering’ is not a solution.

**Solution**:

To efficiently query on the last\_name column without relying on ALLOW FILTERING, we need to incorporate this column into the primary key or create a secondary index. But secondary indexes come with some performance considerations, such as increased storage overhead and potential performance degradation during updates. Using ALLOW FILTERING can also result in unpredictable query performance, as the execution time depends on the size of the dataset and the number of partitions that need to be scanned.

+++++++++++++++++++++++++++++++++++++++++THE END+++++++++++++++++++++++++++++++++++++