CSCI 3287 Design and Analysis of Data System

Project #3 NoSQL Lab – Cassandra

Overview:

This Project is worth 50 points (out of 1000) toward your final grade. It is due on Thursday, Dec. 13th, at 11:59 p.m. **No extension can be allowed for this assignment**. Your submission should be a PDF document submitted as a file via the link found in the **Project Assignment** section of the Week 15 Moodle-- which is the same place where you got this file.

This project will give you hands-on practice in working with the Cassandra "NoSQL" database software.

Objectives:

1.Become familiar with Cassandra

2.Install Cassandra on your computer (Windows, Linux, Mac)

3.Create and load a Cassandra database

4.Perform several basic operations against your Cassandra database

Deliverables:
Capture screen shots to show evidence of having completed assigned operations (1 – 10) described below. Number each screen shot with the number of the assigned operation/task (1 – 10).
Number each screen shot and include some text to describe the task accomplished which is visible in the screen shot. Assemble (Copy & Paste) all screen shots and text into a document. Save the document as a PDF.

Submission:

Use the submission link in the Project Assignment section of the Week 15 Moodle -- which is the same place where you got this file.
This is an **individual** assignment, no collaboration allowed. Each student must submit your own final deliverable for this assignment

Introduction:

Requirements :
1) Windows

- Windows 7 or Windows 2008 server
- The latest version of Java 7 or above. If not, you can follow the instructions in this tutorial. Install Java 8 on your Windows
- Either the Firefox or Chrome Web browser for DataStax OpsCenter (which doesn't support Internet Explorer yet)
- Windows Installation Instructions follow below in Appendix A

2) Ubuntu

- The latest version of Java 7 or above, either the Oracle Java Standard Edition 8 or OpenJDK 8 is installed. If not, you can follow the instructions in this tutorial Install Java 8 on your Ubuntu.
- If you are going to use cqlsh, make sure that the latest version of Python 2.7 or above is installed on your server.
- Ubuntu Installation instructions follow below in Appendix B

3) Mac OS X

- The latest version of Java 7 or above, either the Oracle Java Standard Edition 8 or OpenJDK 8 is installed. If not, you can follow the instructions in this tutorial Install Java 8 on your Mac OS X.
- If you are going to use cqlsh, make sure that the latest version of Python 2.7 or above is installed on your server.
- Mac Installation instructions follow below in Appendix C

Appendix A –Windows Installation

Download the Software : Windows
The first step is to download the software you'll need for your Windows machine.

https://academy.datastax.com/planet-cassandra//cassandra

Note: Download the latest one 3.9.0 Version but this one will not support DataStax OpsCenter as it was removed by Apache. You have to download it separately. Instruction given below but it's optional.

Optional: DataStax makes available the DataStax Community Edition, which contains the latest community version of Apache Cassandra, along with the Cassandra Query Language (CQL) utility, and a free edition of DataStax OpsCenter, which is the tool you'll want to use for managing and monitoring your Cassandra cluster on Windows. To get Datastax Community Edition, go to the downloads page and select the Windows installation package for your version of Microsoft Windows. Note that 32 and 64-bit installers are offered.

Appendix B- Ubuntu Installation:

I'll show you How to install Apache Cassandra on a Ubuntu 16.04. Apache Cassandra is a NoSQL database management system which is free and open-source. It allows managing large amounts of data with high availability without compromising the performance. Installing Apache Cassandra on Ubuntu 16.04 is an easy task, just follow the steps bellow and you should have it done in few minutes.

The documentation is available at http://cassandra.apache.org/doc/latest/ and will help you to learn how to configure and use the service for your projects.

First of all, connect to your Linux server via SSH, update the package index and upgrade all your installed software to the latest version available. You can do that by using the following commands: More Information

```
sudo apt-get update
sudo apt-get upgrade
```

Install Java 8 on Ubuntu 16.04

To install Java 8 on your Ubuntu 16.04 VPS run the following command:

```
sudo apt-get install default-jdk
```

To verify that Java 8 is installed you can use:

```
java -version
```

The output should be very similar to the one below:

```
openjdk version "1.8.0_131" OpenJDK Runtime Environment (build 1.8.0_131-8u131-b11-
0ubuntu1.16.04.2-b11) OpenJDK 64-Bit Server VM (build 25.131-b11, mixed mode)
```

Install Apache Cassandra on Ubuntu 16.04

To install Apache Cassandra on your server, first you need to add the Cassandra repository. At the moment of writing this tutorial, the latest stable release of Cassandra is 3.x.x. Therefore, run the following command to add the Cassandra repository on your server:

```
echo "deb http://www.apache.org/dist/cassandra/debian 311x main" | sudo tee -a
/etc/apt/sources.list.d/cassandra.sources.list
```

Next, add the Cassandra repository keys:

```
curl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key add - sudo apt-key adv --
keyserver pool.sks-keyservers.net --recv-key A278B781FE4B2BDA
```

Update the package index:

```
sudo apt-get update
```

Finally, install Apache Cassandra using the following command:

```
sudo apt-get install cassandra
```

Start, Stop and Enable Apache Cassandra on Ubuntu 16.04
To start the Apache Cassandra service on your server, you can use the following command:

```
sudo systemctl start cassandra.service
```

To stop the service, you can use the command below:

```
sudo systemctl stop cassandra.service
```

If the service is not already enabled on system boot, you can enable it by using the command below:

```
sudo systemctl enable cassandra.service
```

Appendix C – Installation on Mac

Mac Installation

If you use Mac OS X as your platform for development work, then you may be interested to know how easy it is to use Apache Cassandra on the Mac. The following shows you how to download and setup Cassandra, its utilities, and also use DataStax OpsCenter, which is a browser-based, visual management and monitoring tool for Cassandra.

Download the Software

DataStax makes available the DataStax Community Edition, which contains the latest community version of Apache Cassandra, along with the Cassandra Query Language (CQL) utility, and a free edition of DataStax OpsCenter. To get Datastax Community Edition, go to Planet Cassandra and download both Cassandra and OpsCenter, and select the tar downloads of both the DataStax Community Server and OpsCenter.

You can also use the curl command on Mac to directly download the files to your machine. For example, to download the DataStax Community Server, you could enter the following at terminal prompt:

```
curl -OL http://downloads.datastax.com/community/dsc.tar.gz
```

Install Cassandra

Once your download of Cassandra finishes, move the file to whatever directory you'd like to use for testing Cassandra. Then uncompress the file (whose name will change depending on the version you're downloading):

```
tar -xzf dsc-cassandra-1.2.2-bin.tar.gz
```

Then switch to the new Cassandra bin directory and start up Cassandra

Now that you have Cassandra running, the next thing to do is connect to the server and begin creating database objects. This is done with the Cassandra Query Language (CQL) utility. CQL is a very SQL-like language that lets you create objects as you're likely used to doing in the RDBMS world**. Type ./cqlsh in your bin directory to make sure the cassandra executable.**

Database Operations (50 points)

These tasks are shown in detail in Mac environment. However, once you have installed the software the tasks are pretty much the same regardless of platform. So even if you are running on Linux or Windows, please read through the Mac environment to get a feel for these tasks.

1. Create a keyspace (schema or "collection") (4)
2. Use a keyspace (4)
3. Describe a keyspace (4)
4. Create and Describe a table (column family) (6)
5. Insert a new row (4)
6. Create an index (4)
7. Update a value (4)
8. Add/Drop/Rename a column (9)
9. Run a select query (5)
10. Drop a table and Drop keyspace (4)

A simple record creation
All Cassandra commands are case-insensitive.

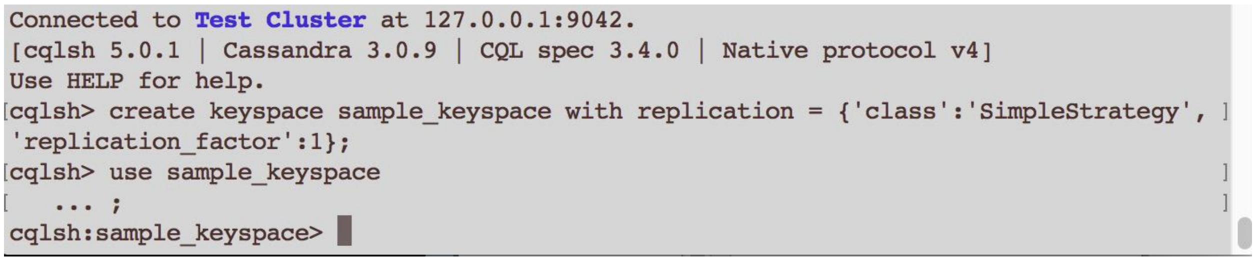
1) Create a keyspace

create keyspace sample_keyspace with replication={'class':'SimpleStrategy', 'replication_factor':1};
Note: Once a keyspace is created, you can create column families (the primary data object in Cassandra), insert data, query data, and more:
If you want to study about different strategy in Cassandra and it's replication factor, the link below provided the detailed instruction. https://bit.ly/2zFc82h

What is a keyspace?

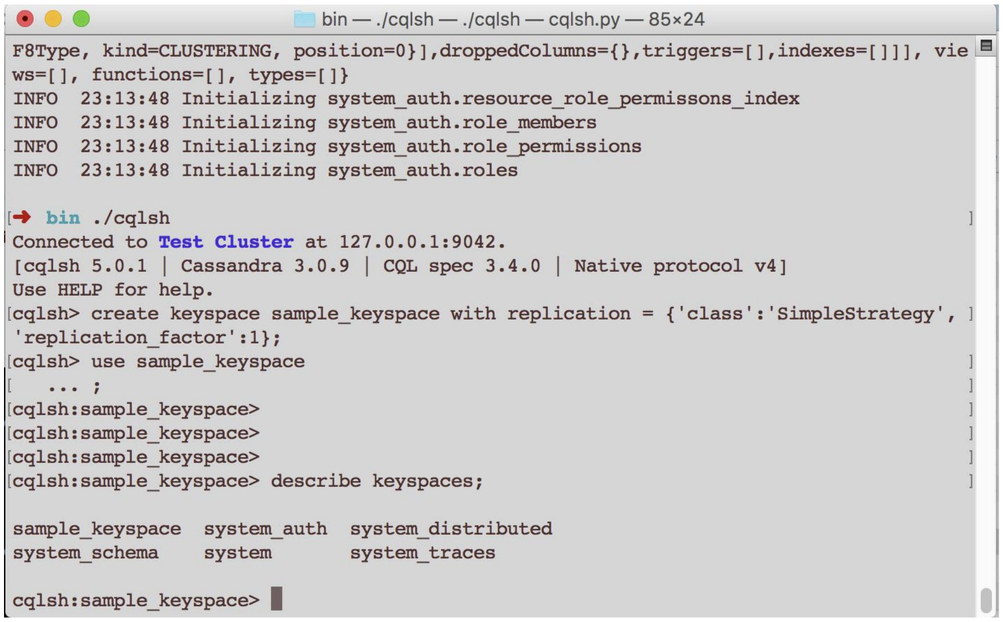
A keyspace is a logical container for data tables and indexes. It can be compared to an Oracle Schema or a SQL Server database. Keyspaces also define how the data is replicated to the various nodes

2) Use keyspace;
use sample_keyspace;



```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.0.9 | CQL spec 3.4.0 | Native protocol v4]
Use HELP for help.
cqlsh> create keyspace sample_keyspace with replication = {'class':'SimpleStrategy',
'replication_factor':1};
cqlsh> use sample_keyspace
   ... ;
cqlsh:sample_keyspace>
```

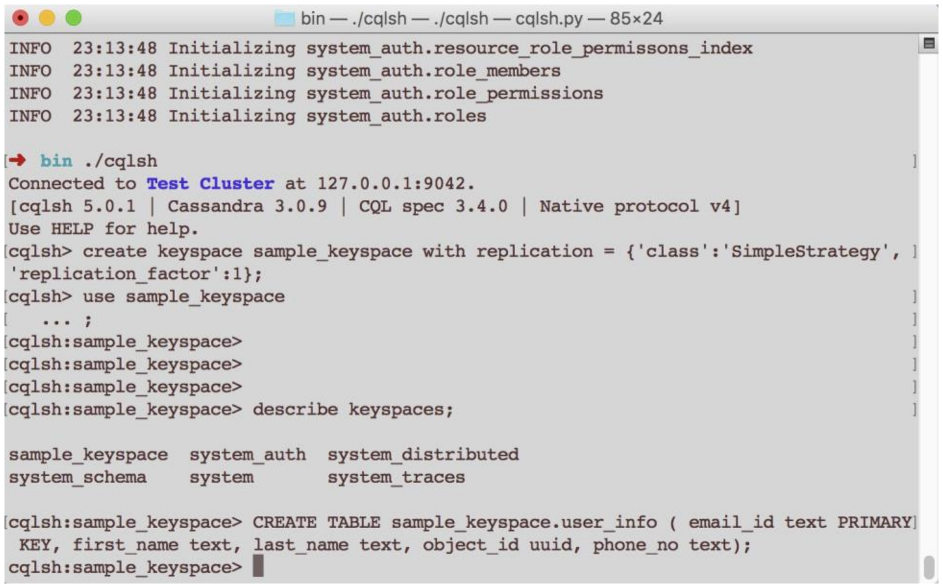
3) If you want to see all keyspaces in cassandra, just type describe keyspaces;



```
F8Type, kind=CLUSTERING, position=0}],droppedColumns={},triggers=[],indexes=[]]], vie
ws=[], functions=[], types=[]}
INFO  23:13:48 Initializing system_auth.resource_role_permissons_index
INFO  23:13:48 Initializing system_auth.role_members
INFO  23:13:48 Initializing system_auth.role_permissions
INFO  23:13:48 Initializing system_auth.roles

➜ bin ./cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.0.9 | CQL spec 3.4.0 | Native protocol v4]
Use HELP for help.
cqlsh> create keyspace sample_keyspace with replication = {'class':'SimpleStrategy',
'replication_factor':1};
cqlsh> use sample_keyspace
   ... ;
cqlsh:sample_keyspace>
cqlsh:sample_keyspace>
cqlsh:sample_keyspace>
cqlsh:sample_keyspace> describe keyspaces;

sample_keyspace  system_auth  system_distributed
system_schema    system       system_traces

cqlsh:sample_keyspace>
```

4) To Create a table

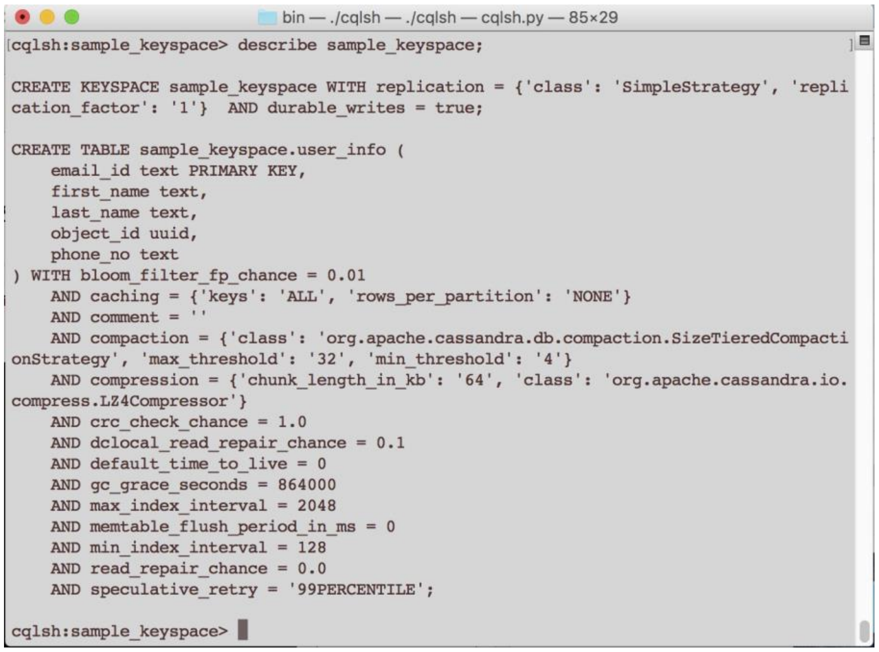
```
CREATE TABLE sample_keyspace.user_info (
email_id text PRIMARY KEY,
first_name text,
last_name text,
object_id uuid,
phone_no text);
```


bin — ./cqlsh — ./cqlsh — cqlsh.py — 85×24

```
INFO  23:13:48 Initializing system_auth.resource_role_permissons_index
INFO  23:13:48 Initializing system_auth.role_members
INFO  23:13:48 Initializing system_auth.role_permissions
INFO  23:13:48 Initializing system_auth.roles

➜ bin ./cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.0.9 | CQL spec 3.4.0 | Native protocol v4]
Use HELP for help.
cqlsh> create keyspace sample_keyspace with replication = {'class':'SimpleStrategy',
'replication_factor':1};
cqlsh> use sample_keyspace
   ... ;
cqlsh:sample_keyspace>
cqlsh:sample_keyspace>
cqlsh:sample_keyspace>
cqlsh:sample_keyspace> describe keyspaces;

sample_keyspace  system_auth  system_distributed
system_schema    system       system_traces

cqlsh:sample_keyspace> CREATE TABLE sample_keyspace.user_info ( email_id text PRIMARY
 KEY, first_name text, last_name text, object_id uuid, phone_no text);
cqlsh:sample_keyspace>
```

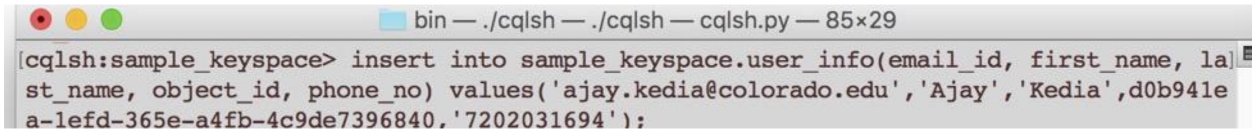
5) If you want to see all column family in cassandra, just type
describe tables;

6) Describe a table;
describe sample_keyspace;


bin — ./cqlsh — ./cqlsh — cqlsh.py — 85×29

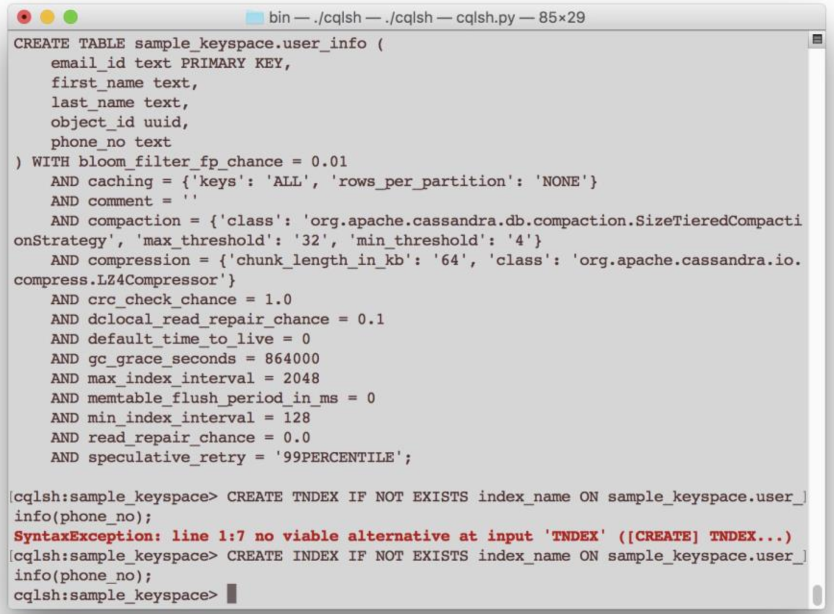
```
cqlsh:sample_keyspace> describe sample_keyspace;

CREATE KEYSPACE sample_keyspace WITH replication = {'class': 'SimpleStrategy', 'repli
cation_factor': '1'}  AND durable_writes = true;

CREATE TABLE sample_keyspace.user_info (
    email_id text PRIMARY KEY,
    first_name text,
    last_name text,
    object_id uuid,
    phone_no text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompacti
onStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.
compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';

cqlsh:sample_keyspace>
```

7) Insert command
insert into cuwallet_platform.user_info(email_id, first_name, last_name, object_id, phone_no) values('ajay.kedia@colorado.edu','Ajay','Kedia',d0b941ea-1efd-365e-a4fb-4c9de7396840,'7202031694');



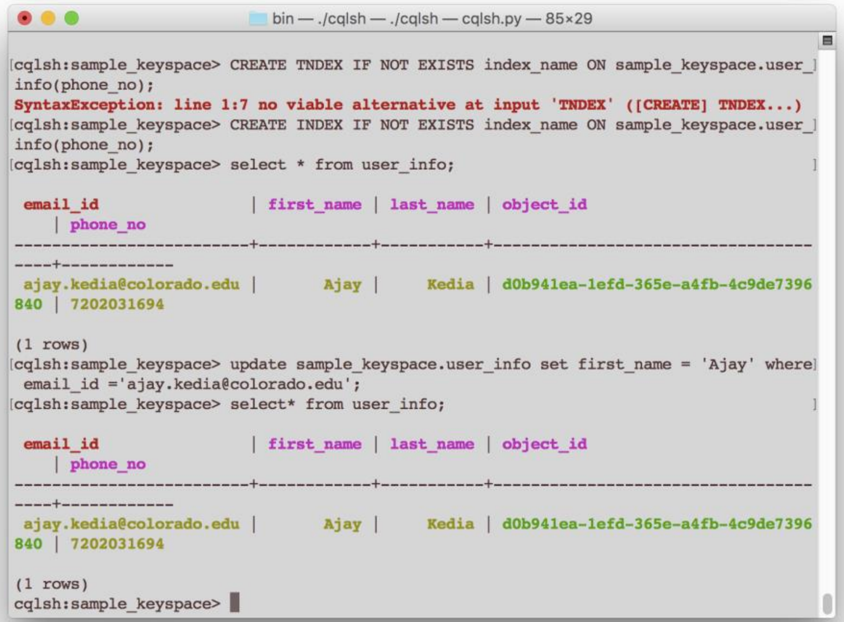
```
[cqlsh:sample_keyspace> insert into sample_keyspace.user_info(email_id, first_name, la]
st_name, object_id, phone_no) values('ajay.kedia@colorado.edu','Ajay','Kedia',d0b941e
a-1efd-365e-a4fb-4c9de7396840,'7202031694');
```

8) Index Creation
CREATE INDEX IF NOT EXISTS index_name ON sample_keyspace.user_info(phone_no);



```
CREATE TABLE sample_keyspace.user_info (
    email_id text PRIMARY KEY,
    first_name text,
    last_name text,
    object_id uuid,
    phone_no text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompacti
onStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.
compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';

[cqlsh:sample_keyspace> CREATE TNDEX IF NOT EXISTS index_name ON sample_keyspace.user_]
info(phone_no);
SyntaxException: line 1:7 no viable alternative at input 'TNDEX' ([CREATE] TNDEX...)
[cqlsh:sample_keyspace> CREATE INDEX IF NOT EXISTS index_name ON sample_keyspace.user_]
info(phone_no);
cqlsh:sample_keyspace>
```

9) Update Command
update sample_keyspace.user_info set first_name = 'Ajay' where email_id = 'ajay.kedia@colorado.edu';

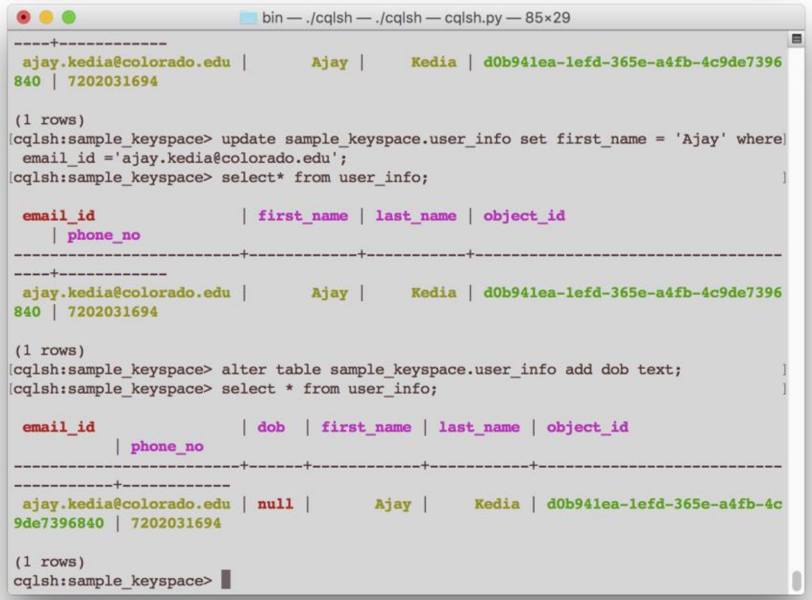


```
[cqlsh:sample_keyspace> CREATE TNDEX IF NOT EXISTS index_name ON sample_keyspace.user_]
 info(phone_no);
SyntaxException: line 1:7 no viable alternative at input 'TNDEX' ([CREATE] TNDEX...)
[cqlsh:sample_keyspace> CREATE INDEX IF NOT EXISTS index_name ON sample_keyspace.user_]
 info(phone_no);
[cqlsh:sample_keyspace> select * from user_info;                                      ]

 email_id                 | first_name | last_name | object_id
    | phone_no
--------------------------+------------+-----------+-----------------------------------
----+------------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396
840 | 7202031694

(1 rows)
[cqlsh:sample_keyspace> update sample_keyspace.user_info set first_name = 'Ajay' where]
 email_id ='ajay.kedia@colorado.edu';
[cqlsh:sample_keyspace> select* from user_info;                                       ]

 email_id                 | first_name | last_name | object_id
    | phone_no
--------------------------+------------+-----------+-----------------------------------
----+------------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396
840 | 7202031694

(1 rows)
cqlsh:sample_keyspace>
```

10) Alter Command: More Info could be seen from this link:

1) Add a column

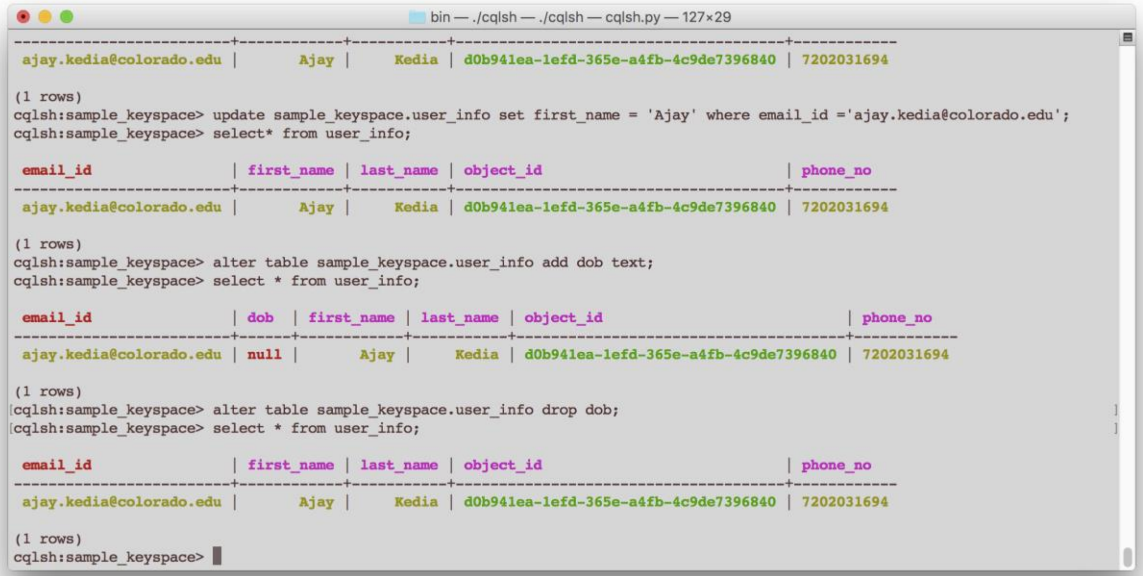
alter table sample_keyspace.user_info add dob text;



```
----+------------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396
840 | 7202031694

(1 rows)
[cqlsh:sample_keyspace> update sample_keyspace.user_info set first_name = 'Ajay' where]
 email_id ='ajay.kedia@colorado.edu';
[cqlsh:sample_keyspace> select* from user_info;                                       ]

 email_id                 | first_name | last_name | object_id
    | phone_no
--------------------------+------------+-----------+-----------------------------------
----+------------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396
840 | 7202031694

(1 rows)
[cqlsh:sample_keyspace> alter table sample_keyspace.user_info add dob text;           ]
[cqlsh:sample_keyspace> select * from user_info;                                      ]

 email_id                 | dob  | first_name | last_name | object_id
           | phone_no
--------------------------+------+------------+-----------+----------------------------
-----------+------------
 ajay.kedia@colorado.edu | null |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c
9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace>
```

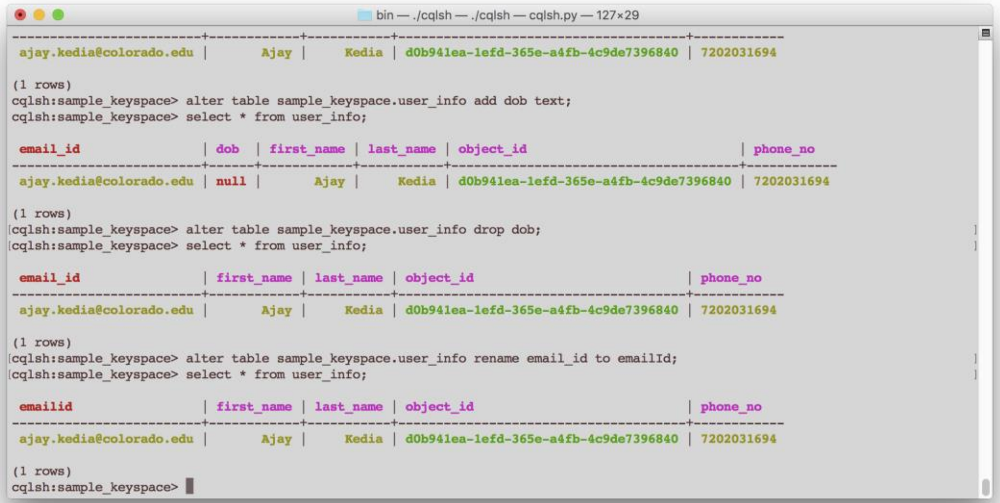
2) Drop a column

alter table sample_keyspace.user_info drop dob ;



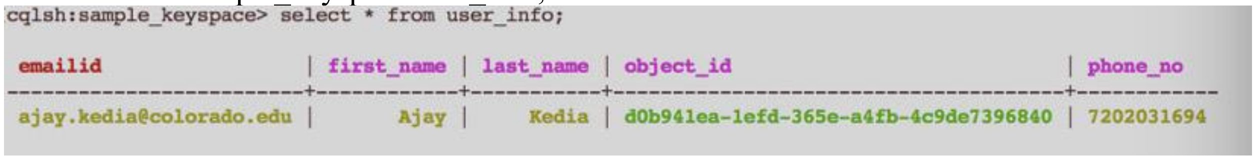
```
-------------------------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> update sample_keyspace.user_info set first_name = 'Ajay' where email_id ='ajay.kedia@colorado.edu';
cqlsh:sample_keyspace> select* from user_info;

 email_id                | first_name | last_name | object_id                            | phone_no
-------------------------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> alter table sample_keyspace.user_info add dob text;
cqlsh:sample_keyspace> select * from user_info;

 email_id                | dob  | first_name | last_name | object_id                            | phone_no
-------------------------+------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu | null |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> alter table sample_keyspace.user_info drop dob;
cqlsh:sample_keyspace> select * from user_info;

 email_id                | first_name | last_name | object_id                            | phone_no
-------------------------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> 
```

3) Rename a column : Only primary key column can be renamed.
alter table sample_keyspace.user_info rename email_id to emailId;



```
-------------------------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> alter table sample_keyspace.user_info add dob text;
cqlsh:sample_keyspace> select * from user_info;

 email_id                | dob  | first_name | last_name | object_id                            | phone_no
-------------------------+------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu | null |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> alter table sample_keyspace.user_info drop dob;
cqlsh:sample_keyspace> select * from user_info;

 email_id                | first_name | last_name | object_id                            | phone_no
-------------------------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> alter table sample_keyspace.user_info rename email_id to emailId;
cqlsh:sample_keyspace> select * from user_info;

 emailid                 | first_name | last_name | object_id                            | phone_no
-------------------------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> 
```

11) Select
Select * from sample_keyspace.user_info;



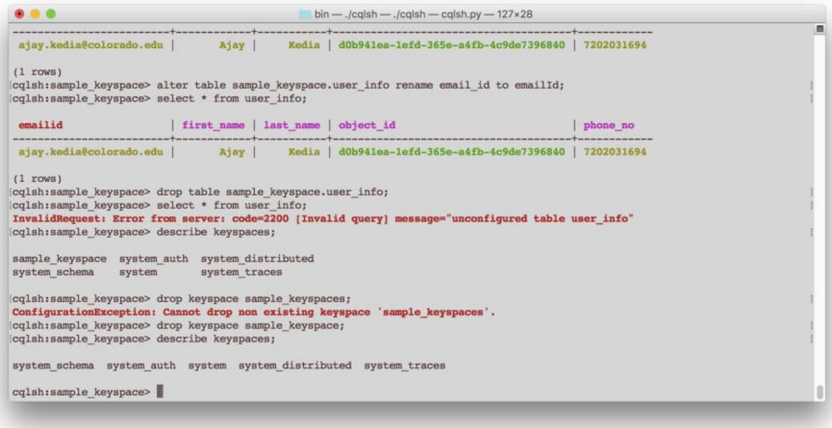
```
cqlsh:sample_keyspace> select * from user_info;

 emailid                 | first_name | last_name | object_id                            | phone_no
-------------------------+------------+-----------+--------------------------------------+-----------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694
```

12) Drop a table

drop table sample_keyspace.user_info;

13) Drop a keyspace

drop keyspace sample_keyspace;



```
bin — ./cqlsh — ./cqlsh — cqlsh.py — 127×28
-------------------------+------------+-----------+--------------------------------------+------------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> alter table sample_keyspace.user_info rename email_id to emailId;
cqlsh:sample_keyspace> select * from user_info;

 emailid                 | first_name | last_name | object_id                            | phone_no
-------------------------+------------+-----------+--------------------------------------+------------
 ajay.kedia@colorado.edu |       Ajay |     Kedia | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694

(1 rows)
cqlsh:sample_keyspace> drop table sample_keyspace.user_info;
cqlsh:sample_keyspace> select * from user_info;
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured table user_info"
cqlsh:sample_keyspace> describe keyspaces;

sample_keyspace  system_auth  system_distributed
system_schema    system       system_traces

cqlsh:sample_keyspace> drop keyspace sample_keyspaces;
ConfigurationException: Cannot drop non existing keyspace 'sample_keyspaces'.
cqlsh:sample_keyspace> drop keyspace sample_keyspace;
cqlsh:sample_keyspace> describe keyspaces;

system_schema  system_auth  system  system_distributed  system_traces

cqlsh:sample_keyspace>
```