

## Project 3: Cassandra

1&2) (One creates and the other really shows they are created, so I combined these two)

1-> create keyspace key with replication = {'class':'SimpleStrategy',  
'replication\_factor':1};

(Creates the environment we are going to use)

2-> use key;

(uses the said environment)

```
cqlsh> create keyspace key with replication = {'class':'SimpleStrategy', 'replication_factor':1};  
cqlsh> use key;  
cqlsh:key>
```

3) describe keyspaces;

(Shows the available environments)

```
cqlsh:key> describe keyspaces;  
  
system_schema  system  system_distributed  a_keyspace  
system_auth    key     system_traces  
  
cqlsh:key> _
```

4) CREATE TABLE key.user(email\_id text PRIMARY KEY, first\_name text, object\_id uuid, phone\_no text);

(Creates a table in the environment given the values you want it to be available)

```
cqlsh:key> CREATE TABLE key.user(email_id text PRIMARY KEY, first_name text, object_id uuid, phone_no text);  
cqlsh:key> _
```

5) describe tables;

(Shows which tables exist in this environment)

```
cqlsh:key> describe tables;  
  
user  
  
cqlsh:key>
```

6) describe key;

(Shows what this environment is and the tables it has (the schema basically))

```
cqlsh:key> describe key;  
  
CREATE KEYSPACE key WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'} AND durable_writes = true;  
  
CREATE TABLE key.user (  
    email_id text PRIMARY KEY,  
    first_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.SizeTieredCompactionStrategy', '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:key>
```

7) INSERT INTO key.user(email\_id,first\_name, object\_id, phone\_no) VALUES

('ajay.kedia@colorado.edu','Ajay',d0b941ea-1efd-365e-a4fb-4c9de7396840,'7202031694');

(Adds values to the table)

```
cqlsh:key> INSERT INTO key.user(email_id,first_name, object_id, phone_no) VALUES ('ajay.kedia@colorado.edu','Ajay',d0b941ea-1efd-365e-a4fb-4c9de7396840,'7202031694');  
cqlsh:key>
```

(stretched to make at least a little bit legible)

8) create index if not exists ind on key.user(phone\_no);

(Creates an index for better search later)

```
cqlsh:key> create index if not exists ind on key.user(phone_no);
cqlsh:key>
```

9) update key.user set first\_name='AJ' where email\_id='ajay.kedia@colorado.edu';

(Updates the value of a specified column of a specified row)

```
cqlsh:key> select * from user
... ;

email_id          | first_name | object_id          | phone_no
-----+-----+-----+-----
ajay.kedia@colorado.edu |      Ajay | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694
(1 rows)
cqlsh:key> update key.user set first_name='AJ' where email_id='ajay.kedia@colorado.edu';
cqlsh:key> select * from user
... ;

email_id          | first_name | object_id          | phone_no
-----+-----+-----+-----
ajay.kedia@colorado.edu |         AJ | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694
(1 rows)
cqlsh:key>
```

10)

a) alter table key.user add last\_name text;

(Adds a column to the table)

```
cqlsh:key> select * from user
... ;

email_id          | first_name | object_id          | phone_no
-----+-----+-----+-----
ajay.kedia@colorado.edu |         AJ | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694
(1 rows)
cqlsh:key> alter table key.user add last_name text;
cqlsh:key> select * from user
... ;

email_id          | first_name | last_name | object_id          | phone_no
-----+-----+-----+-----+-----
ajay.kedia@colorado.edu |         AJ |      null | d0b941ea-1efd-365e-a4fb-4c9de7396840 | 7202031694
(1 rows)
cqlsh:key>
```

b) alter table key.user drop last\_name;

(Deletes the column from a table)

```
cqlsh:key> select * from user
... ;
```

email_id	first_name	last_name	object_id	phone_no
ajay.kedia@colorado.edu	AJ	null	d0b941ea-1efd-365e-a4fb-4c9de7396840	7202031694

```
(1 rows)
cqlsh:key> alter table key.user drop last_name;
cqlsh:key> select * from user
... ;
```

email_id	first_name	object_id	phone_no
ajay.kedia@colorado.edu	AJ	d0b941ea-1efd-365e-a4fb-4c9de7396840	7202031694

```
(1 rows)
cqlsh:key> _
```

c) alter table key.user rename email\_id to email;

(Renames ONLY THE PRIMARY KEY)

```
cqlsh:key> select * from user
... ;
```

email_id	first_name	object_id	phone_no
ajay.kedia@colorado.edu	AJ	d0b941ea-1efd-365e-a4fb-4c9de7396840	7202031694

```
(1 rows)
cqlsh:key> alter table key.user rename first_name to name;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot rename non PRIMARY KEY part"
cqlsh:key> alter table key.user rename email_id to email;
cqlsh:key> select * from user;
```

email	first_name	object_id	phone_no
ajay.kedia@colorado.edu	AJ	d0b941ea-1efd-365e-a4fb-4c9de7396840	7202031694

```
(1 rows)
cqlsh:key> _
```

11) select \* from user;

(How to do queries in Cassandra)

```
cqlsh:key> select * from user;
```

email	first_name	object_id	phone_no
ajay.kedia@colorado.edu	AJ	d0b941ea-1efd-365e-a4fb-4c9de7396840	7202031694

(1 rows)  
cqlsh:key> \_

12&13) (Once again combined, as they are quite similar)

12-> drop table key.user;

(Deletes the table in the environment)

13-> drop keyspace key;

(Deletes the environment)

```
cqlsh:key> drop table key.user
... ;
cqlsh:key> drop keyspace key
... ;
cqlsh:key> describe keyspaces
```

system_schema	system	system_traces
system_auth	system_distributed	a_keyspace

cqlsh:key> \_

(key is gone)