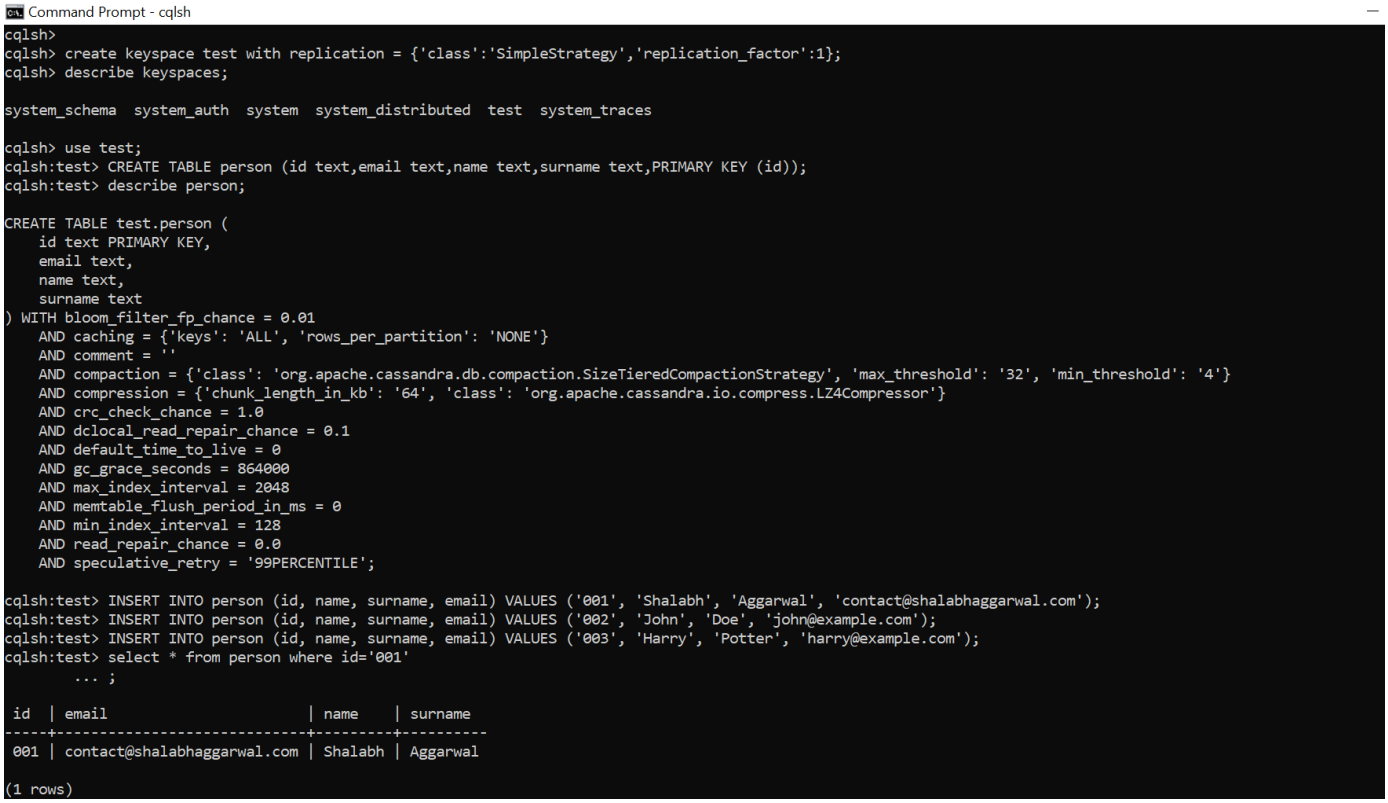
Exercise: Installing and using keyspace, column families and sets with Cassandra

By Priyanka Labh.



4. Create a Keyspace called test with a replication factor 1 and simple strategy.

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

5. Go to the keyspace test

cqlsh> use test;

cqlsh:test>

6.- Create a column family (table) called person with id text, email text, name text, surname text and is as primary key PRIMARY KEY (id));

cqlsh:test> CREATE TABLE person (id text,email text,name text,surname text,PRIMARY KEY (id));

7.- check the schema of the column family person.

cqlsh:test> describe person;

CREATE TABLE test.person (id text PRIMARY KEY, email text,name text,surname 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';

8. Populate person column family.

cqlsh:test> INSERT INTO person (id, name, surname, email) VALUES ('001', 'Shalabh', 'Aggarwal', 'contact@shalabhaggarwal.com');

cqlsh:test> INSERT INTO person (id, name, surname, email) VALUES ('002', 'John', 'Doe', 'john@example.com');

cqlsh:test> INSERT INTO person (id, name, surname, email) VALUES ('003', 'Harry', 'Potter', 'harry@example.com');

9. Get all information from person SELECT * FROM person; 10. get all information from person where id = 001
cqlsh:test> SELECT name FROM person WHERE id='001';

10. get all information from person where id = 001 SELECT name FROM person WHERE id='001';

cqlsh:test> select * from person where id='001';

11.use sets

cqlsh:test> CREATE COLUMNFAMILY users (key varchar PRIMARY KEY,full_name varchar,birth_date int,state varchar,emails set<text>);

12.- create an index on users with column birth_date and other index on state.

cqlsh:test> CREATE INDEX ON users (birth_date);

cqlsh:test> CREATE INDEX ON users (state);

13. populate users columnfamily with the following data:

INSERT INTO users (key, full_name, birth_date, state,emails) VALUES (' pangeles', 'Pilar Angeles', 1975, 'UT', 'mpahotmailcom');

INSERT INTO users (key, full_name, birth_date, state,emails) VALUES ('asmith', 'Alice Smith', 1973, 'WI','asmithGmailcom');

INSERT INTO users (key, full_name, birth_date, state,emails) VALUES ('htayler', 'Howard Tayler', 1968, 'UT','htyHotmailcom');

```
cqlsh:test> select * from users;

 key | birth_date | emails | full_name | state
-----+------------+--------+-----------+-------

(0 rows)
cqlsh:test> INSERT INTO users (key, full_name, birth_date, state,emails) VALUES (' pangeles', 'Pilar Angeles', 1975, 'UT', 'mpahotmailcom');
cqlsh:test> INSERT INTO users (key, full_name, birth_date, state) VALUES ('asmith', 'Alice Smith', 1973, 'WI','asmithGmailcom');
InvalidRequest: Error from server: code=2200 [Invalid query] message="Unmatched column names/values"
cqlsh:test> INSERT INTO users (key, full_name, birth_date, state,emails) VALUES ('asmith', 'Alice Smith', 1973, 'WI','asmithGmailcom');
cqlsh:test> INSERT INTO users (key, full_name, birth_date, state,emails) VALUES ('htayler', 'Howard Tayler', 1968, 'UT','htyHotmailcom');
cqlsh:test> select * from users;

 key       | birth_date | emails         | full_name     | state
-----------+------------+----------------+---------------+-------
   htayler |       1968 |  htyHotmailcom | Howard Tayler |    UT
    asmith |       1973 | asmithGmailcom |   Alice Smith |    WI
  pangeles |       1975 |  mpahotmailcom | Pilar Angeles |    UT

(3 rows)
cqlsh:test>
```

13. Get full name and emails for Pilar Angeles Select full_name, emails from users where key= 'pangeles'

cqlsh:test> Select full_name, emails from users where key= 'pangeles' ;

14. Get key and state from users.

cqlsh:test> select key,state from users;

15. Get all users that live in UT and were born after 1970

cqlsh:test> SELECT * FROM users WHERE state='UT' AND birth_date > 1970 ALLOW FILTERING;

```
cqlsh:test> Select full_name, emails from users where key= 'asmith';

 full_name   | emails
-------------+----------------
 Alice Smith | asmithGmailcom

(1 rows)
cqlsh:test> select key,state from users;

 key       | state
-----------+-------
   htayler |    UT
    asmith |    WI
  pangeles |    UT

(3 rows)
cqlsh:test> SELECT * FROM users WHERE state='UT' AND birth_date > 1970 ALLOW FILTERING;

 key       | birth_date | emails         | full_name     | state
-----------+------------+----------------+---------------+-------
  pangeles |       1975 | mpahotmailcom | Pilar Angeles |    UT

(1 rows)
cqlsh:test>
```