**Tạo Key Space**

CREATE KEYSPACE group1\_keyspace WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': '1'} AND durable\_writes = 'true';

use group1\_keyspace;

**CREATE TABLE, INSERT (1 khóa chính không có row key)**

CREATE TABLE employee\_by\_id (id int PRIMARY KEY, name text, positon text);

INSERT INTO employee\_by\_id (id, name, positon) VALUES (1, 'Khanh', 'CEO');

INSERT INTO employee\_by\_id (id, name, positon) VALUES (2, 'Nhi', 'Manager');

INSERT INTO employee\_by\_id (id, name, positon) VALUES (3, 'Lam', 'Leader');

**CREATE TABLE, INSERT (row key (1 column) và 1 cột phân vùng clustering column)**

CREATE TABLE employee\_by\_car\_make (car\_make text, id int, car\_model text, PRIMARY KEY (car\_make, id));

INSERT INTO employee\_by\_car\_make (car\_make, id, car\_model) VALUES ('BMW', 1, 'A Car');

INSERT INTO employee\_by\_car\_make (car\_make, id, car\_model) VALUES ('AUDI', 2, 'B Car');

INSERT INTO employee\_by\_car\_make (car\_make, id, car\_model) VALUES ('BMW', 3, 'C Car');

SELECT \* FROM employee\_by\_car\_make;

INSERT INTO employee\_by\_car\_make (car\_make, id, car\_model) VALUES ('AUDI', 4, 'D Car');

**CREATE TABLE, INSERT (row key (2 column) và các cột phân vùng clustering column)**

CREATE TABLE employee\_by\_car\_make\_and\_model (car\_make text, car\_model text, id int, age int, name text, PRIMARY KEY ( (car\_make, car\_model), age, id ));

INSERT INTO employee\_by\_car\_make\_and\_model (car\_make, car\_model, age, id, name) VALUES ('BMW', 'A Car', 18, 1, 'Khanh');

INSERT INTO employee\_by\_car\_make\_and\_model (car\_make, car\_model, age, id, name) VALUES ('BMW', 'A Car', 18, 2, 'Nhi');

INSERT INTO employee\_by\_car\_make\_and\_model (car\_make, car\_model, age, id, name) VALUES ('BMW', 'A Car', 18, 5, 'Lam');

INSERT INTO employee\_by\_car\_make\_and\_model (car\_make, car\_model, age, id, name) VALUES ('BMW', 'A Car', 18, 5, 'Lam');

INSERT INTO employee\_by\_car\_make\_and\_model (car\_make, car\_model, age, id, name) VALUES ('BMW', 'D Car', 20, 8, 'Thao');

INSERT INTO employee\_by\_car\_make\_and\_model (car\_make, car\_model, age, id, name) VALUES ('BMW', 'D Car', 17, 1, 'Thong');

INSERT INTO employee\_by\_car\_make\_and\_model (car\_make, car\_model, age, id, name) VALUES ('AUDI', 'E Car', 18, 1, 'Justin');

**SELECT table employee\_by\_id**

SELECT \* FROM employee\_by\_id ;

SELECT \* FROM employee\_by\_id WHERE id = 1;

**SELECT table employee\_by\_car\_make**

SELECT \* FROM employee\_by\_car\_make;

SELECT \* FROM employee\_by\_car\_make WHERE car\_make = 'BMW';

SELECT \* FROM employee\_by\_car\_make WHERE car\_make = 'BMW' AND id = 3;

**SELECT table employee\_by\_car\_make\_and\_model**

SELECT \* FROM employee\_by\_car\_make\_and\_model ;

SELECT \* FROM employee\_by\_car\_make\_and\_model WHERE car\_make= 'BMW' AND car\_model= 'A Car' ;

SELECT \* FROM employee\_by\_car\_make\_and\_model WHERE car\_make= 'BMW' AND car\_model= 'A Car' AND age = 18;

SELECT \* FROM employee\_by\_car\_make\_and\_model WHERE car\_make= 'BMW' AND car\_model= 'A Car' AND age = 18 AND id = 2;

**Update table employee\_by\_id**

SELECT \* FROM employee\_by\_id;

UPDATE employee\_by\_id SET positon = 'Employee' where id= 1;

**Update table employee\_by\_car\_make**

SELECT \* FROM employee\_by\_car\_make;

UPDATE employee\_by\_car\_make SET car\_model = 'F Car' WHERE car\_make = 'BMW' AND id = 1;

**Update table employee\_by\_car\_make\_and\_model**

SELECT \* FROM employee\_by\_car\_make\_and\_model;

UPDATE employee\_by\_car\_make\_and\_model SET name = 'Chris Brown' WHERE car\_make = 'BWM' AND car\_model = 'A Car' AND age = 18 AND id = 1;

**DELETE table employee\_by\_id**

SELECT \* FROM employee\_by\_car\_make\_and\_model;

DELETE FROM employee\_by\_id WHERE id= 2;

**Delete table employee\_by\_car\_make**

SELECT \* FROM employee\_by\_car\_make;

DELETE FROM employee\_by\_car\_make WHERE car\_make = 'BMW' ;

DELETE FROM employee\_by\_car\_make WHERE car\_make = 'AUDI' AND id=2 ;

**Update table employee\_by\_car\_make\_and\_model**

SELECT \* FROM employee\_by\_car\_make\_and\_model;

DELETE FROM employee\_by\_car\_make\_and\_model WHERE car\_make = 'BMW' AND car\_model = 'A Car';

DELETE FROM employee\_by\_car\_make\_and\_model WHERE car\_make = 'BMW' AND car\_model = 'D Car' AND age= 20;

DELETE FROM employee\_by\_car\_make\_and\_model WHERE car\_make = 'BMW' AND car\_model = 'D Car' AND age= 17 AND id=1 ;

**ALTER TABLE employee\_by\_id (Collection)**

ALTER TABLE employee\_by\_id ADD phone set<text>;

UPDATE employee\_by\_id SET phone = {'123', '456'} WHERE id = 1;

UPDATE employee\_by\_id SET phone = phone + { '555'} WHERE id= 1;

UPDATE employee\_by\_id SET phone = phone - { '555'} WHERE id= 1;

UPDATE employee\_by\_id SET phone ={} WHERE id=1;

**Import file csv**

CREATE TABLE test\_import\_export (car\_make text, car\_model text, start\_year int, id int, first\_name text, last\_name text, department text, PRIMARY KEY (car\_make, car\_model, start\_year, id));

SELECT \* FROM test\_import\_export ;

COPY test\_import\_export (car\_make, car\_model, start\_year, id, first\_name, last\_name, department) FROM '/home/group1/Desktop/cassandra.csv' WITH DELIMITER = ',' AND HEADER = 'TRUE';

**Export file csv**

COPY test\_import\_export (car\_make, car\_model, start\_year, id, first\_name, last\_name, department) TO '/home/group1/Desktop/test\_export.csv' WITH DELIMITER = ',' AND HEADER = 'true';

Hoặc chỉ đỉnh vài cột

COPY test\_import\_export (car\_make, car\_model, first\_name, department) TO '/home/group1/Desktop/test\_export\_by\_column.csv' WITH DELIMITER = ',' AND HEADER = 'true' ;

**Dùng python để tương tác với Cassandra**

CREATE TABLE test\_python (id uuid PRIMARY KEY, first\_name text, last\_name text);

from cassandra.cluster import Cluster

# Tạo một kết nối đến Cassandra

cluster = Cluster()

# Tạo một phiên làm việc

session = cluster.connect('group1\_keyspace')

# Tạo một phiên làm việc

result = session.execute("INSERT INTO test\_python (id, first\_name, last\_name) VALUES (uuid(), 'Khanh', 'Ho')")

print ("INSERT")

#đóng kết nối phiên

session.shutdown()

# thuc hien cau truy van

result = session.execute("SELECT \* FROM test\_python")

for row in result:

print(row.first\_name, row.last\_name)

**Truy vấn cột theo kiểu tạo một view (giúp truy vấn vào cột không phải row key hay clustering column nhanh hơn)**

CREATE MATERIALIZED VIEW employee\_by\_department AS SELECT \* FROM test\_import\_export WHERE department IS NOT NULL AND car\_make IS NOT NULL AND car\_model IS NOT NULL AND start\_year IS NOT NULL AND id IS NOT NULL PRIMARY KEY (department , car\_make, car\_model, start\_year ,id);

SELECT \* FROM employee\_by\_department;

**=>Sau đó thì cập nhật department bên table test\_import\_export thì department bên materialized view employee\_by\_apartment cũng sẽ thay đổi theo**

SELECT \* FROM test\_import\_export;

UPDATE test\_import\_export SET department = 'IT' WHERE car\_make = 'BMW' AND car\_model = 'Compact' AND start\_year = 2012 AND id= 5;

SELECT \* FROM employee\_by\_department;

**Secondary Indexs Truy vấn theo cột không phải là row key và có thể là clustering column (cột phân vùng) hoặc không:**

SELECT \* FROM employee\_by\_id WHERE name='Khanh' ALLOW FILTERING;

UPDATE employee\_by\_id SET phone = {'111', '222'} WHERE id = 1;

SELECT \* FROM employee\_by\_id WHERE phone CONTAINS '111' ALLOW FILTERING;

**Secondary Indexs tạo Index và truy vần theo name**

CREATE INDEX ON employee\_by\_id (name);

SELECT \* FROM employee\_by\_id WHERE name='Lam';

**TimeStamp**

SELECT id, name, writetime (name) FROM employee\_by\_id;

**TTL (Time to live) sau 30s tự động sẽ mất giá trị sau khi đc đổi => null**

UPDATE employee\_by\_id USING TTL 30 SET name='HK' WHERE id=3;

SELECT \* FROM employee\_by\_id;

**ID là một khóa chính về thời gian**

CREATE TABLE employee\_by\_timeuuid (id timeuuid PRIMARY KEY, first\_name text, last\_name text);

INSERT INTO employee\_by\_timeuuid (id, first\_name, last\_name) VALUES (now (), 'Khanh', 'Ho');

INSERT INTO employee\_by\_timeuuid (id, first\_name, last\_name) VALUES (now (), 'Nhi', 'Nguyen');

INSERT INTO employee\_by\_timeuuid (id, first\_name, last\_name) VALUES (now (), 'Lam', 'Nguyen');

**ID là một khóa chính ngẫu nhiên**

CREATE TABLE employee\_by\_uuid (id uuid PRIMARY KEY, first\_name text, last\_name text);

INSERT INTO employee\_by\_timeuuid (id, first\_name, last\_name ) VALUES (uuid(), 'Khanh', 'Ho');

INSERT INTO employee\_by\_uuid (id, first\_name, last\_name ) VALUES (uuid(), 'Nhi', 'Nguyen');

INSERT INTO employee\_by\_uuid (id, first\_name, last\_name ) VALUES (uuid(), 'Lam', 'Nguyen');

**Counter (chỉ update không thể insert) (dùng để theo dõi số lượng như like, cmt trên fb)**

CREATE TABLE parchaces\_by\_customer (id uuid PRIMARY KEY, parcheces counter);

**Tạo ra một thằng id ngẫu nhiên bằng lệnh update (update 1 thằng nào đó thì ghi id của nó vào)**

UPDATE parchaces\_by\_customer SET parcheces = parcheces + 1 WHERE id= uuid();

UPDATE parchaces\_by\_customer SET parcheces = parcheces + 1 WHERE id= 2405c0c4-403c-44b7-a730-7cc42556c59c;

UPDATE parchaces\_by\_customer SET parcheces = parcheces - 1 WHERE id= 2405c0c4-403c-44b7-a730-7cc42556c59c;