



Week 9: Cassandra Query Language

January 28

Due date: Wednesday January 29, 2024 at 6:00pm!!!

- สร้างคีໍ່ສເປີ່ງຂຶ້ນ user_management

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

- ຕັ້ງຄ່າ user_management เป็นคីໍ່ສເປີ່ງເຮື່ອມຕົນ (default keyspace)

```
| cqlsh> use user_management;  
| cqlsh:user_management> |
```

3. สร้างตารางชื่อ `users` ดังต่อไปนี้

- 4 คอลัมน์: `last_name`, `first_name`, `email` และ `address`
- คอลัมน์ `last_name`: คีย์พาร์ติชัน
- คอลัมน์ `first_name` และ `email`: คอลัมน์คลัสเตอร์
- คอลัมน์ `address`: คอลัมน์เพย์โหลด

```
cqlsh:user_management> create table users (
    ... last_name text,
    ... first_name text,
    ... email text,
    ... primary key (last_name, first_name, email));
    . . .
```

4. ตรวจสอบโครงสร้างตาราง

```
cqlsh:user_management> select * from users
    . . .

last_name | first_name | email | address
-----+-----+-----+-----+
(0 rows)
```

5. ໄສ່ແຄວໃນຕາງໆດ້ວຍຄໍາຕ່ອໄປນີ້:

Column Name	Value
last_name	Smith
first_name	Alex
email	asmith@gmail.com
address	123 Main st.

```
... values ('Smith', 'Alex', 'asmith@gmail.com', '123 Main st.');
cqlsh:user_management> select * from users ;

last_name | first_name | email           | address
-----+-----+-----+-----+
Smith   |      Alex | asmith@gmail.com | 123 Main st.

(1 rows)
```

6. ໄສ່ແຄວໃນຕາງໆດ້ວຍຄໍາຕ່ອໄປນີ້:

Column Name	Value
last_name	Smith
first_name	Bailey
email	bsmith@gmail.com
address	234 Elm st.

```
cqlsh:user_management> insert into users (last_name, first_name, email, address)
... values ('Smith', 'Bailey', 'bsmith@gmail.com', '234 Elm st.');
cqlsh:user_management> select * from users ;

last_name | first_name | email           | address
-----+-----+-----+-----+
Smith   |      Alex | asmith@gmail.com | 123 Main st.
Smith   |      Bailey | bsmith@gmail.com | 234 Elm st.

(2 rows)
```

7. ใส่ແກຣມຕາງໆດ້ວຍຄ່າຕ່ອໄປນີ້ (Cassandra ຮອງຮັບຄອລັນນີ້ແບບ sparse ຜຶ່ງໝາຍຄວາມວ່າ ຄອລັນນີ້ທີ່ໄມ້ໃຊ້ຄື່ນໍ້າລັກໄມ້ຈໍາເປັນຕ້ອງມີຄ່າ):

Column Name	Value
last_name	Jones
first_name	Chris
email	cjones@gmail.com

```
cqlsh:user_management> insert into users (last_name, first_name, email)
values ('Jones', 'Chris', '
cjones@gmail.com');
```

8. ແສດນີ້ອຫາຂອງຕາງໆ ໂດຍໃຊ້ຄຳສັ່ງ CQL SELECT

```
cqlsh:user_management> select * from users ;
+-----+-----+-----+-----+
| last_name | first_name | email | address |
+-----+-----+-----+-----+
| Jones | Chris | cjones@gmail.com | null |
| Smith | Alex | asmith@gmail.com | 123 Main st. |
| Smith | Bailey | bsmith@gmail.com | 234 Elm st. |
(3 rows)
```

9. เหตุใดจึงเลือกคอลัมน์เหล่านี้สำหรับคีย์หลัก?

ดูวิธีดึงข้อมูลพาร์ติชันเดียวโดยใช้คำสั่ง WHERE: ให้แสดงเนื้อหาของตาราง โดยคอลัมน์ last_name มีค่า Smith

```
cqlsh:user_management> select * from users where last_name = 'Smith';

last_name | first_name | email           | address
-----+-----+-----+-----+
Smith   |      Alex | asmith@gmail.com | 123 Main st.
Smith   |    Bailey | bsmith@gmail.com | 234 Elm st.
```

สมมติว่าเราไม่ต้องการดึงข้อมูลคอลัมน์ทั้งหมดจากແລງ ให้ดึงเฉพาะคอลัมน์ address แทน:

```
cqlsh:user_management> select address from users where last_name = 'Smith';

address
-----
123 Main st.
234 Elm st.
```

หากต้องการขยายฟิลเตอร์ เราสามารถใช้ตัวคั่นด้วยเครื่องหมายจุลภาค (,) ได้ ลองดึงคอลัมน์ first_name และ email:

```
cqlsh:user_management> select first_name, email from users where last_name = 'Smith';

first_name | email
-----+-----+
      Alex | asmith@gmail.com
    Bailey | bsmith@gmail.com

(2 rows)
```

เกิดอะไรขึ้นถ้าต้องการดึงผู้ใช้ทั้งหมด โดยให้คอลัมน์ first_name มีค่า Alex?

```
SELECT * FROM users WHERE first_name = 'Alex';
```

```
cqlsh:user_management> SELECT * FROM users WHERE first_name = 'Alex';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filtering and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, use ALLOW FILTERING"
```

10. เหตุใดการสืบค้น (queries) จึงต้องมีคีย์พาร์ติชันด้วย?

เราจะเห็นว่าการสืบค้นโดยไม่ใช้คีย์พาร์ติชันไม่ทำงาน ลองใช้การสืบค้นอื่น:

```
SELECT * FROM users WHERE last_name = 'Smith' AND email = 'asmith@gmail.com';
```

ตอบ Cassandra จะค้นหาจาก Partition Key และถึงจะเข้าไปเป็น Clustering Column และ Payload Column ด้วยเหตุผลนี้จึงไม่สามารถหาข้อมูลโดยอิงจาก Column ที่ไม่ใช่ Partition Key โดยการหาข้อมูลจะเรียงลำดับเป็น Partition Key, Clustering Column ที่ 1, Clustering Column ที่ 2 และลัด ๆ ไป และสุดท้ายคือ Payload Column คำสั่ง `SELECT * FROM users WHERE last_name = 'Smith' AND email = 'asmith@gmail.com'`; จึงไม่สามารถใช้งานได้ เพราะในขอก่อนหน้าสร้างตารางเรียง Clustering Column เป็น first_name และ email

11. เหตุใดคอลัมน์คลัสเตอร์จึงต้องเรียงตามลำดับการสืบค้น?

Cassandra เก็บແດວຂອງພາර์ຕິ່ນທີ່ເຮັດວຽກຕາມຄອລັນໆຄລັບສເຕ່ອຣ ດັ່ງນັ້ນລອງສ້າງກາຮືບກັນ email ແລະ address ຂອງຜູ້ໃຊ້ທີ່ຄອລັນໆ last_name ມີຄ່າ Smith ແລະ first_name ມີຄ່າ Bailey ໄດ້ຫຼືໄມ້

```
cqlsh:user_management> select email, address from users
... where last_name = 'Smith' and first_name = 'Bailey';
email | address
-----+
bsmith@gmail.com | 234 Elm st.
(1 rows)
```

ຕອບ ທຳໄດ້ ເພຣະ last_name ຄື່ອ partition key first_name ຄື່ອ Clustering Column ທີ່ 1
ໄໝໄດ້ໃຊ້ where ແບບໜຳ Clustering Column ແມ່ຍອນຂໍອທີ່ແລ້ວ

12. ເມື່ອຕັດການປັບປຸງຄ່າຄອລັນໆ ເຮັດວຽກ CQL UPDATE

ລອງປັບປຸງຄ່າຄອລັນໆ address ຂອງ Alex Smith ໃຫ້ເປັນ 987 Main St.:

```
cqlsh:user_management> update users set address = '987 Main St.' where last_name = 'Smith' and first_name = 'Alex' and email = 'asmith@gmail.com';
```

ຈາກເດີມ Chris Jones ໄນມີຄ່າຄອລັນໆ address ລອງໃຫ້ເປັນ 789 Cherry St:

```
cqlsh:user_management> update users
... set address = '789 Cherry St.'
... where last_name = 'Jones' and first_name = 'Chris'
' and email = 'cjones@gmail.com';
cqlsh:user_management> select * from users;
last_name | first_name | email | address
-----+
Jones | Chris | cjones@gmail.com | 789 Cherry St.
Smith | Alex | asmith@gmail.com | 987 Main St.
Smith | Bailey | bsmith@gmail.com | 234 Elm st.
(3 rows)
```

ເກີດອະໄຮຈຶນຄ້າຕ້ອງການເປີ່ຍນອື່ນເອີ່ມລຂອງ Alex Smith ເປັນ alexsmith@gmail.com?

```
UPDATE users SET email = 'alexsmith@gmail.com' WHERE last_name = 'Smith'
```

```
AND first_name = 'Alex' AND email = 'asmith@gmail.com';
```

```
cqlsh:user_management> update users set email = 'alexsmith@gmail.com'
... where last_name = 'Smith' and first_name = 'Alex'
and email = 'asmith@gmail.com';
InvalidRequest: Error from server: code=2200 [Invalid query] message="PRIMARY KEY part email found in SET part"
```

ຕອບ Cassandra ໄນອນຄູາຕໍ່ໃຫ້ແກ້ Primary Key (ທີ່ Partition Key ແລະ Clustering Column)
email ຄື່ອ Clustering Column ຈຶ່ງໄໝ່ສາມາດຮັກແກ້ໄດ້

13. ອັບເດຕແຄວທີ່ໄໝ່ມີອູ້:

```
UPDATE users SET address = '987 Elm St.' WHERE last_name = 'Jones'
```

```
AND first_name = 'Drew' AND email = 'djones@gmail.com';
```

ດູ້ເໜີອນວ່າຈະທຳການໂດຍໄໝ່ມີຂໍອົມືດພລາດ ເກີດອະໄຮຈຶນ ຕຽບສອບໂດຍລອງດຶງຜູ້ໃຊ້ທັງໝາດໃນພາຣີຕິ້ນ Jones?:

```
cqlsh:user_management> update users set address = '987 Elm St.'
... where last_name = 'Jones'
... and first_name = 'Drew'
... and email = 'djones@gmail.com';
cqlsh:user_management> select * from users
... where last_name = 'Jones';

last_name | first_name | email           | address
-----+-----+-----+-----+
Jones   |     Chris  | cjones@gmail.com | 789 Cherry St.
Jones   |      Drew  | djones@gmail.com | 987 Elm St.

(2 rows)
```

ຕອບ ທຳການໂດຍໄໝ່ມີຂໍອົມືດພລາດ ເພຣະຮະບູ Partition Key ແລະ Clustering Column
ຄຽບຄ້ວນແລະຖືກຕ້ອງຕາມລຳດັບ

14. คำสั่ง CQL DELETE ให้เราสามารถลบค่า ทั้งค่าของคอลัมน์และทั้งแถว

ลองลบค่า address สำหรับ Chris Jones:

```
cqlsh:user_management> delete address from users
    ... where last_name = 'Jones'
    ... and first_name = 'Chris'
    ... and email = 'cjones@gmail.com';
cqlsh:user_management> select * from users ;

last_name | first_name | email           | address
-----+-----+-----+-----+
Jones   |     Chris | cjones@gmail.com |      null
Jones   |     Drew  | djones@gmail.com | 987 Elm St.
Smith  |     Alex  | asmith@gmail.com | 987 Main St.
Smith  |     Bailey | bsmith@gmail.com | 234 Elm st.

(4 rows)
```

ลองลบทั้งแถวโดยไม่ต้องระบุค่าใดๆ สำหรับ Drew Jones:

```
cqlsh:user_management> delete from users
    ... where last_name = 'Jones'
    ... and first_name = 'Drew'
    ... and email = 'djones@gmail.com'
    ... ;
cqlsh:user_management> select * from users ;

last_name | first_name | email           | address
-----+-----+-----+-----+
Jones   |     Chris | cjones@gmail.com |      null
Smith  |     Alex  | asmith@gmail.com | 987 Main St.
Smith  |     Bailey | bsmith@gmail.com | 234 Elm st.

(3 rows)
```

ເກີດຂອງໄຈ້ນເມື່ອເຮົາພາຍານລົບຄໍາຄອລິນ໌ຄລ້ສເຕອຣ໌?

```
DELETE email FROM users WHERE last_name = 'Smith' AND first_name = 'Alex'  
AND email = 'asmith@gmail.com';
```

```
cqlsh:user_management> delete email from users where last_name = 'Smith'  
and first_name = 'Alex' and email = 'asmith@gmail.com';  
InvalidRequest: Error from server: code=2200 [Invalid query] message="In  
valid identifier email for deletion (should not be a PRIMARY KEY part)"
```

ຕອບ ເພົ່າວ່າ email ເປັນ Primary Key

15. ລົບພາບຕີ້ນ Smith ທີ່ໜຳໃຫ້ດ້ວຍຄຳສັ່ງເດືອກຫຼືໄວ້

```
cqlsh:user_management> delete from users  
... where last_name = 'Smith';
```

ຕອບ ລົບໄດ້

16. ແສດຜລລັບຮສຸດທ້າຍຂອງຕາரາງ users

```
SELECT * FROM users;
```

```
cqlsh:user_management> select * from users;  
  
last_name | first_name | email | address  
-----+-----+-----+-----  
Jones | Chris | cjones@gmail.com | null  
  
(1 rows)
```