# 1. Xác định các thực thể có trong mxh:

## Chủ thể người: User, Group, Organization

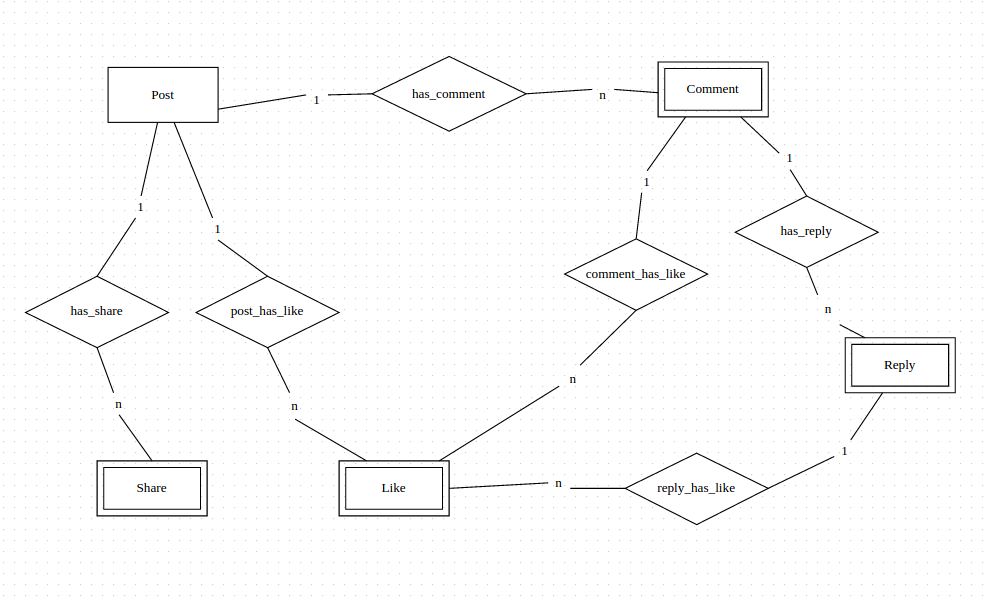
* 1. Các đối tượng con (nested object): Location (Address), Communication (Phone/Email...) Education, Occupation ...
  2. Các dẫn xuất quan hệ của chủ thể người: Friend, FriendList, Following, Blocked, Member...

## Đối tượng tương tác: Post, Like, Share, Comment, Reply, Request

* 1. Các đối tượng con (nested object): Video, Image (media)
  2. Các dẫn xuất quan hệ của đối tượng tương tác: like\_post, like\_reply, like\_comment…

# 2. Xác định mối quan hệ giữa các thực thể (1..n, n..n)

Ví dụ: Hình vẽ minh họa các thực thể theo sprint 1



# 3. Xác định các truy vấn dữ liệu cần phải đáp ứng => số lượng các table, nội dung của các cột trong từng table.

## Phải xác định được primary key, partition key và clustering columns:

* 1. Dữ liệu của 1 row sẽ được lưu trữ hoàn toàn trên 1 node cassandra và được indexing theo partition key (trùng với primary key nếu là key đơn hoặc column đầu tiên trong primary key - nếu là key phức hợp), nếu tận dụng user\_id để xây dựng primary key ta hoàn toàn có thể đảm bảo được việc lưu trữ, truy vấn lấy các dữ liệu của 1 user (profile, post, media ...) một cách nhanh chóng mà không tốn thời gian cho việc discover data giữa các node (hoặc cluster)

## Từ chiều lấy dữ liệu của chủ thể và đối tượng tương tác => table tương ứng.

* 1. E.g comment có thể được truy vấn theo 2 kịch bản:
     1. Show tất cả comment của 1 post
     2. Show tất cả comment của 1 user (trên nhiều post) theo thời gian.

=> Để việc lấy dữ liệu thuận tiện thì nên thiết kế 2 bảng comment\_by\_post, comment\_by\_user.

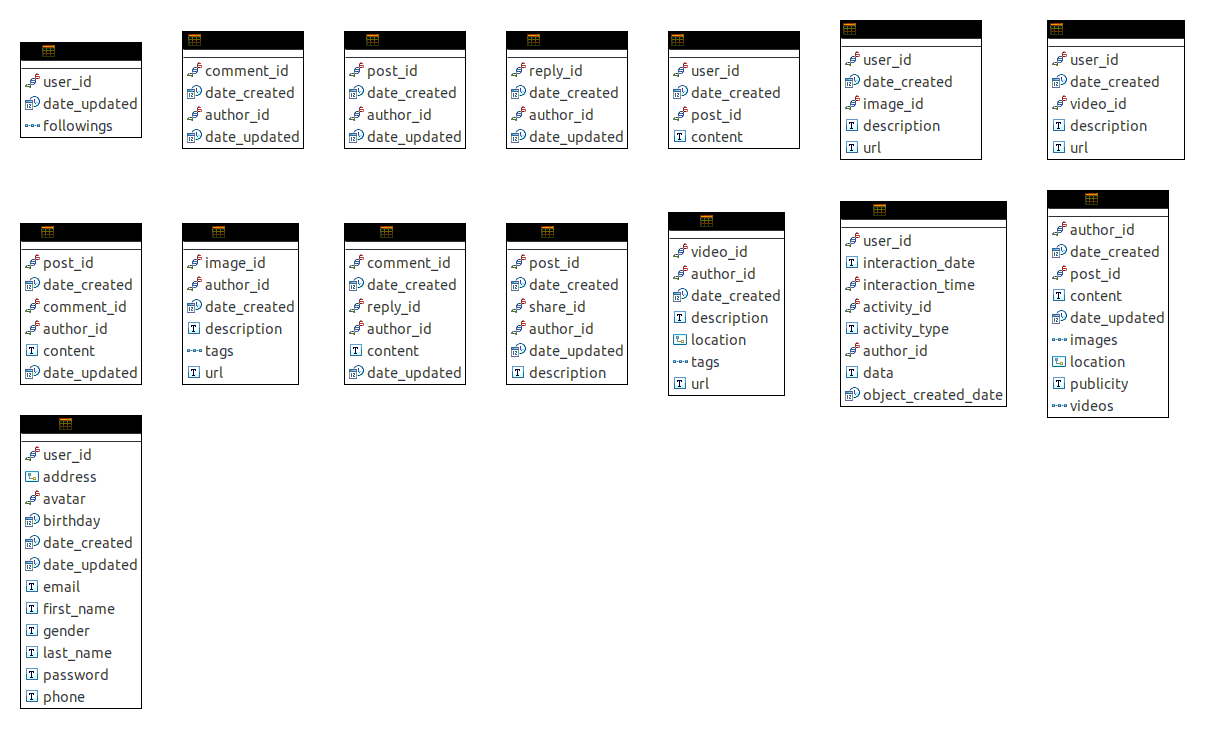
Tham khảo thêm các file slide:

[Data modeling in Cassandra](https://docs.google.com/presentation/d/1WEBqHCixu5eu4QsIWnVzc_fBkLNEY6s1PNvSNa7-naQ/edit?usp=sharing)

[Basic rules of cassandra data modeling](http://www.datastax.com/dev/blog/basic-rules-of-cassandra-data-modeling)

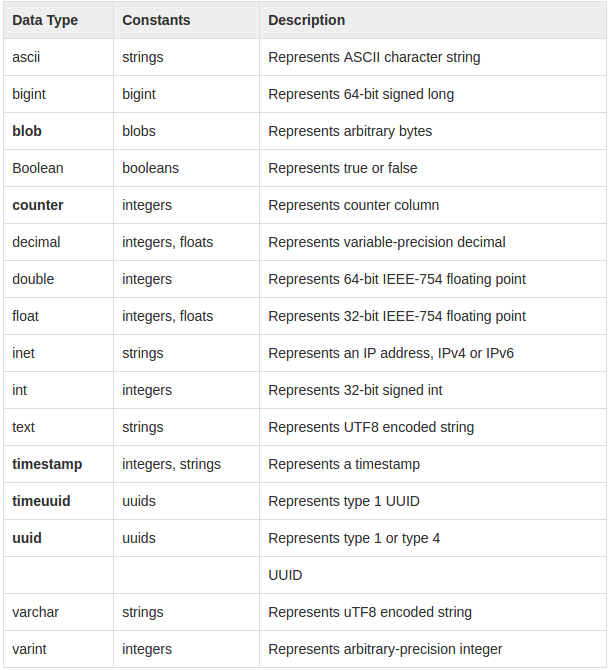
[Data modeling example](http://docs.datastax.com/en/cql/3.1/cql/ddl/ddl_intro_c.html)

# 4. Schema hocvalam (xem file hocvalam.cql)



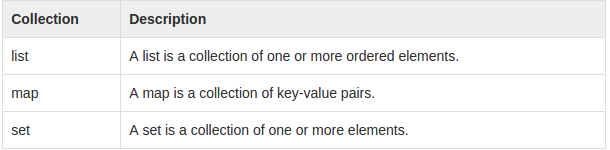
# 5. Các chú ý về sử dụng data type của Cassandra

## Basic type



* Các function sử dụng với type timeuuid: dateOf(), now(), minTimeuuid(), maxTimeuuid(), unixTimestampOf()
* Sử dụng blob cho các trường hợp cần lưu trữ object dạng serializable (e.g class Java) về cơ bản các dữ liệu image, video, document sẽ truy cập thông qua CDN nên không lưu trữ trực tiếp trong DB
* timestamp chú ý Cassandra chỉ phân biệt kiểu dữ liệu này tới millisecond, nghĩa là 3 chữ số thập phân sau second, do đó khi thao tác insert query dữ liệu cần chú ý. Có thể query timestamp theo format yyyy-mm-dd'T'HH:mm:ssZ hoặc integer đều được.
* Chỉ sử dụng counter type cho table bao gồm counter type và primary key, không thể tạo index trên counter column

## 2. Collection types:



E.g:

### Sử dụng set

CREATE TABLE users (

user\_id text PRIMARY KEY,

first\_name text,

last\_name text,

emails set<text> );

Set value phải unique và được đánh dấu bởi {}

INSERT INTO users (user\_id, first\_name, last\_name, emails) VALUES('frodo', 'Frodo', 'Baggins', {'f@baggins.com', ['baggins@gmail.com](mailto:%27baggins@gmail.com)'});

Để add/remove phần thử khỏi set sử dụng toán tử +/-

UPDATE users  
 SET emails = emails + {'fb@friendsofmordor.org'} WHERE user\_id = 'frodo';

UPDATE users  
 SET emails = emails - {'fb@friendsofmordor.org'} WHERE user\_id = 'frodo';

Để remove toàn bộ set, có thể dử dụng UPDATE hoặc DELETE

UPDATE users SET emails = {} WHERE user\_id = 'frodo';  
hoặc  
DELETE emails FROM users WHERE user\_id = 'frodo';

khi đấy kết quả trả về của query cho trường có type set sẽ là null

SELECT user\_id, emails FROM users WHERE user\_id = 'frodo';

user\_id | emails  
*---------+---------*  
 frodo | null

### Sử dụng list

ALTER TABLE users ADD top\_places list<text>;

Dữ liệu dạng list được đánh dấu bởi [ ], vì list là collection có sắp xếp nên các phép insert update cũng phải được thực hiện theo index chính xác, ví dụ: đầu cuối hoặc giữa

UPDATE users  
 SET top\_places = [ 'the shire' ] + top\_places WHERE user\_id = 'frodo';

UPDATE users  
 SET top\_places = top\_places + [ 'mordor' ] WHERE user\_id = 'frodo';

UPDATE users SET top\_places[2] = 'riddermark' WHERE user\_id = 'frodo';

Xóa phần tử khỏi list thông qua index hoặc phép toán -

DELETE top\_places[3] FROM users WHERE user\_id = 'frodo';

Hoặc

UPDATE users  
 SET top\_places = top\_places - ['riddermark'] WHERE user\_id = 'frodo';

### Sử dụng map

ALTER TABLE users ADD todo map<timestamp, text>;

Dữ liệu dạng map được đánh dấu bởi {key : value}

UPDATE users  
 SET todo =  
 { '2012-9-24' : 'enter mordor',  
 '2014-10-2 12:00' : 'throw ring into mount doom' }  
 WHERE user\_id = 'frodo';

Update giá trị của các value trong map thông qua key tương ứng

UPDATE users SET todo['2014-10-2 12:00'] = 'throw my precious into mount doom'  
 WHERE user\_id = 'frodo';

Add phần tử vào map bằng toán tử +

UPDATE users SET todo = todo + { '2013-9-22 12:01' : 'birthday wishes to Bilbo', '2013-10-1 18:00': 'Check into Inn of Pracing Pony'} WHERE user\_id='frodo';

Có thể set time to live TTL cho 1 bản ghi cụ thể trong map

UPDATE users USING TTL 23 --sau 23s bản ghi sẽ bị delete  
 SET todo['2012-10-1'] = 'find water' WHERE user\_id = 'frodo';

Có thể tạo index theo key cho 1 map:

CREATE INDEX todoIndex on users (KEYS(todo));

## 3. User-defined type

Sử dụng trong trường hợp cần lưu trữ một nhóm các thuộc tính - không cần thiết phải tạo một thực thể độc lập, ví dụ: phương thức liên lạc, địa chỉ…

E.g:

CREATE TYPE hocvalam.address (

street text,

city text,

zip int,

phones set<frozen<hocvalam.phone>>,

location frozen<hocvalam.location>

);

khai bảo và sử dụng user-defined type luôn phải đi kèm với keyword frozen

CREATE TABLE hocvalam.user (

user\_id uuid,

***address frozen<hocvalam.address>,***

.....

PRIMARY KEY (user\_id)

)

Có thể thực hiện alter user-defined type hoặc drop type tương tự như trường hợp với table

ALTER field\_name TYPE new\_type  
| ( ADD field\_name new\_type )  
| ( RENAME field\_name TO field\_name )   
 ( AND field\_name TO field\_name ) …

DROP TYPE IF EXISTS type\_name;

6. Việc sử dụng CDN để lưu trữ dữ liệu image, video, file thông qua pk của các đối tượng này UUID => tùy theo provider, DNS => cách thức lưu trữ trong db (có lưu url hay sử dụng dynamic).