

## Agenda

- UML Diagrams ← 16
- Class Diagrams
- Use Case Diagrams
- Schema Design (SQL Schema Design)
  - Finding Cardinalities
  - Case Study
  - How to map diff rel's in Schema design.
- Code the case study in Spring Boot

COMMUNICATION

How to approach UML

PM: Understand Requirements

Timeline

Managers/leads + Architect/CTO

- Design Ques
- Implementation of a System
- Approach

How to communicate

① Discuss via words

- Verbally
- Written
- Email

Ambiguity

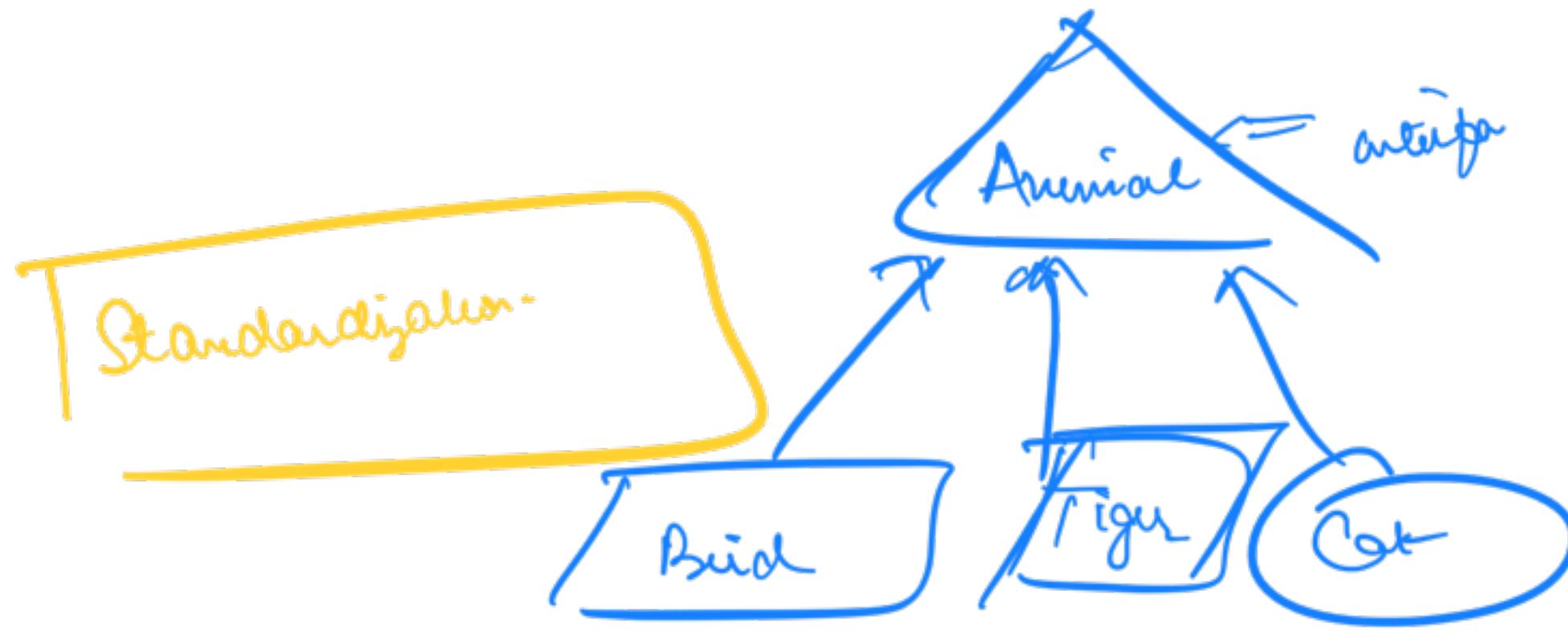
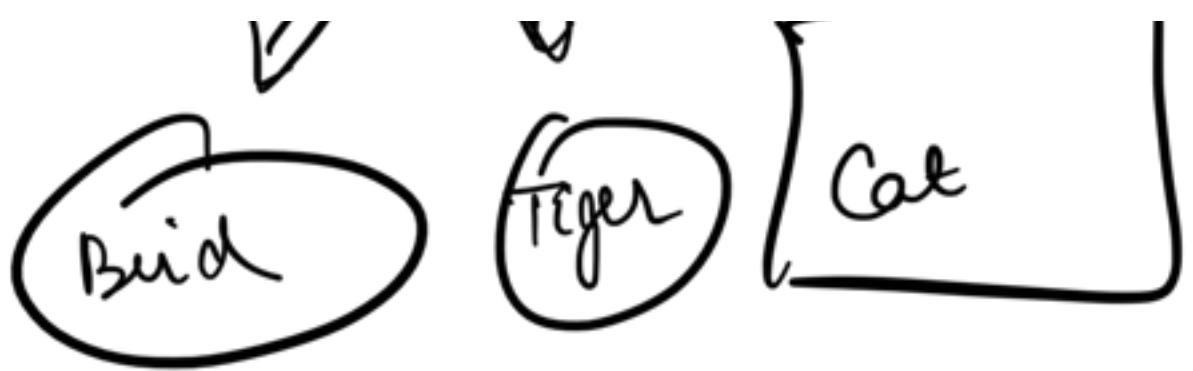
Understand completely

② An image speaks 1000 words

Diagram

- Less Ambiguity
- Ask clarifications
- Give suggestions





UML (Unified Modelling Language)





## 2 Types of UML Diag

① Structural UML Diag

→ concerned around  
structure of  
a codebase

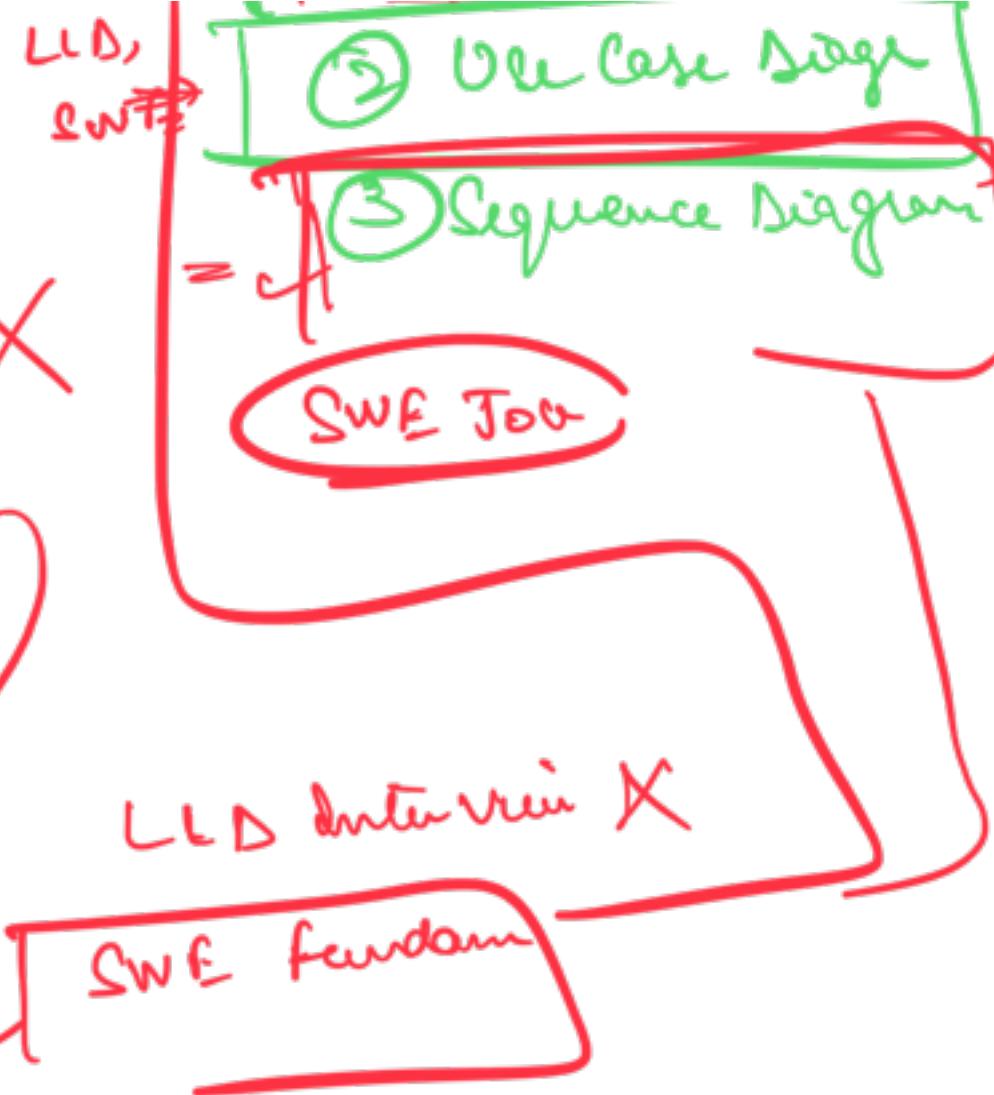
② Behavioral UML Diag

↳ how systems work

① Class Diagram

① Activity Diagram

- ② Component Diagram  
③ Package Diagram  
④ Object Diagram



~~Use Case Diagram, Class Diagram~~

HW: Activity, Sequence Diagram → YT video

↑  
Not up for vote  
SWE Job

Use Case Diagram → Behavioural UML Diagram

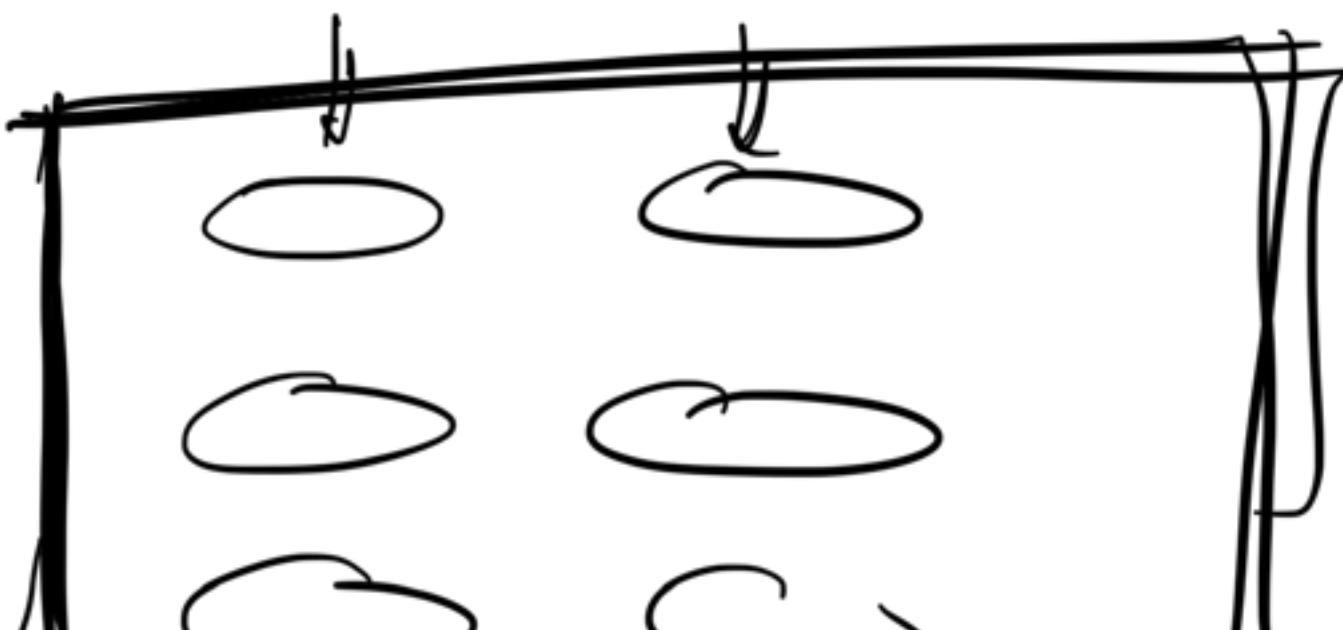
functionalities

→ diff set of use cases that are supported by a software system

→ Who is going to use those services

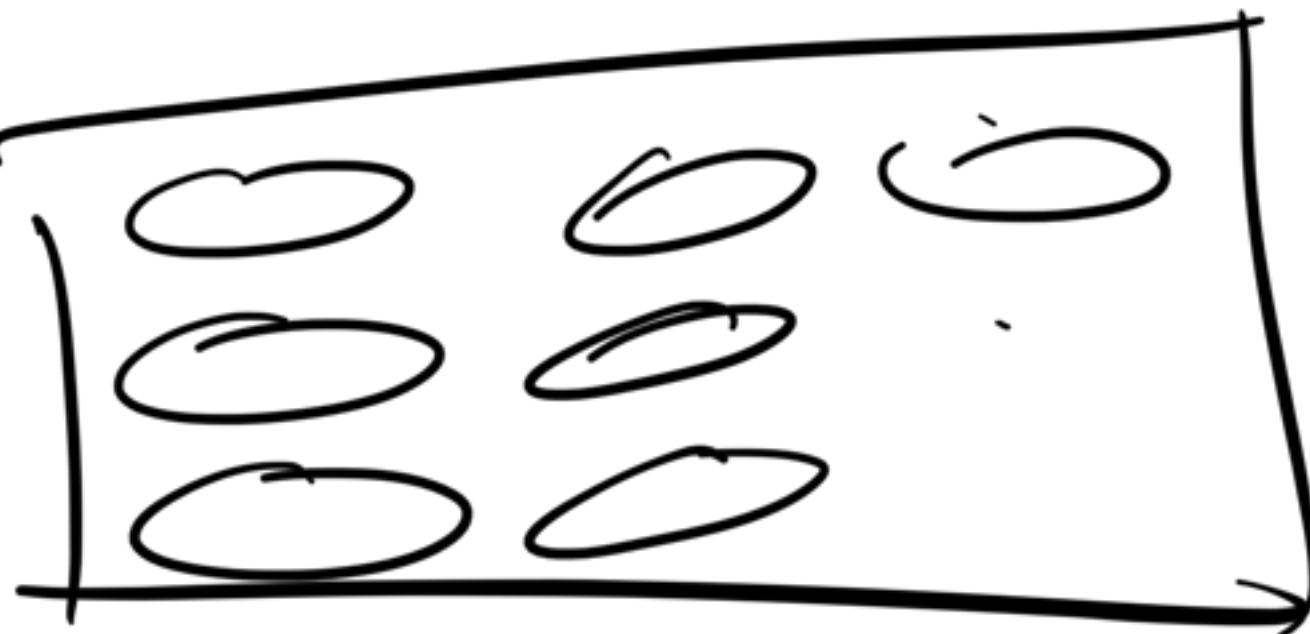
5 key words

① System boundary



~~System~~  
contains the complete scope of the system

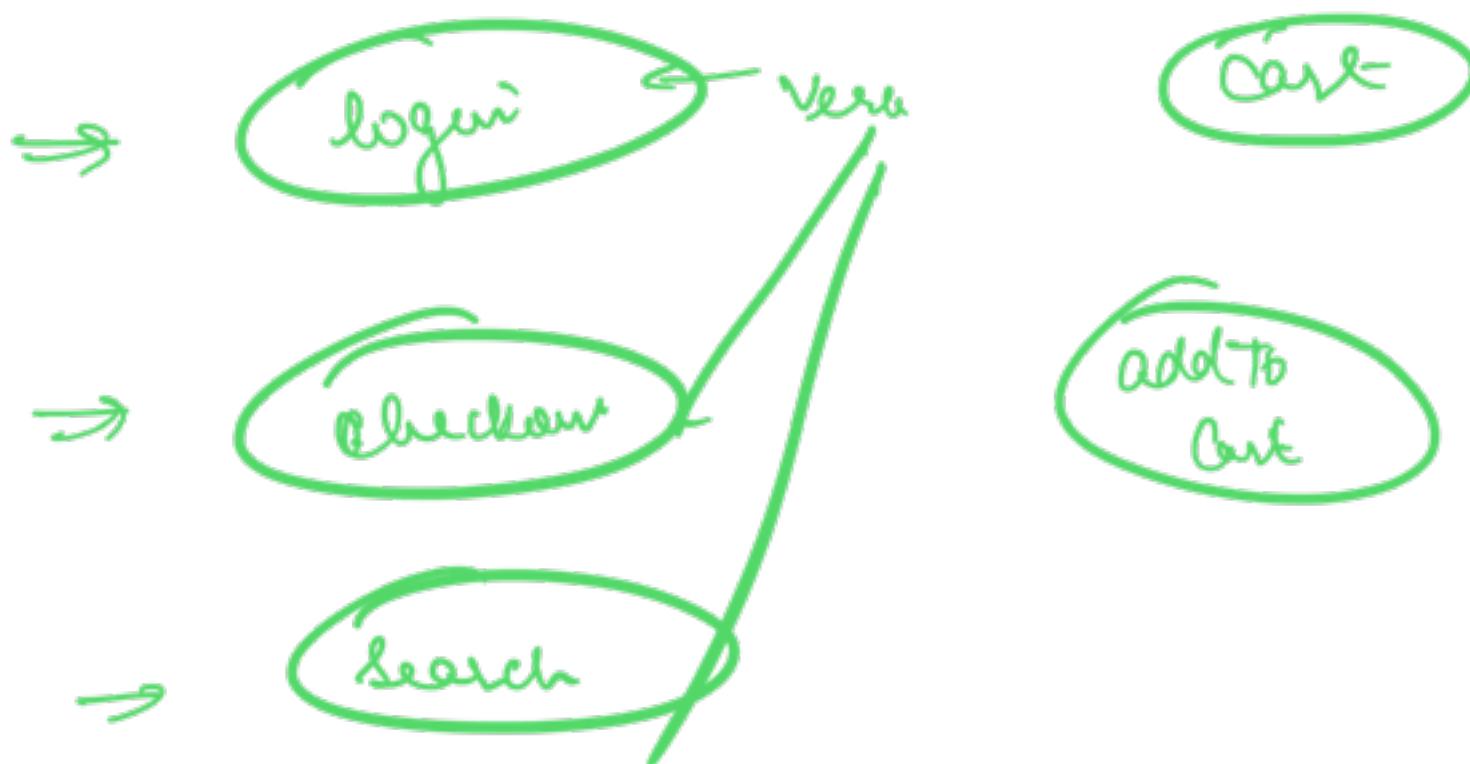
Scope



## ② Use Case

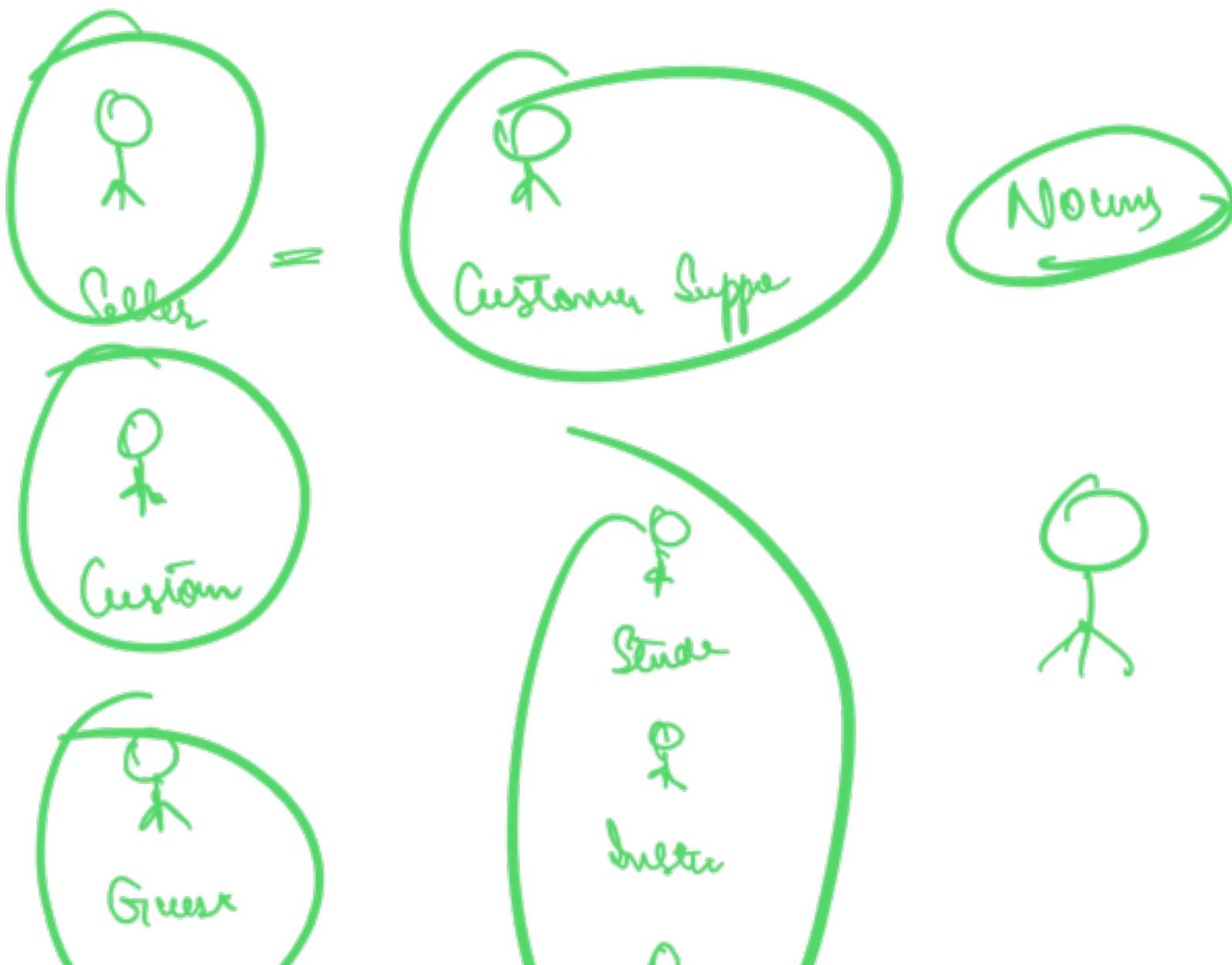
feature / functionality that is supported  
by the system

→ Diff actions that are supported by a sys



### ③ Actor

→ diff entities that interact with the system to perform actions in use case



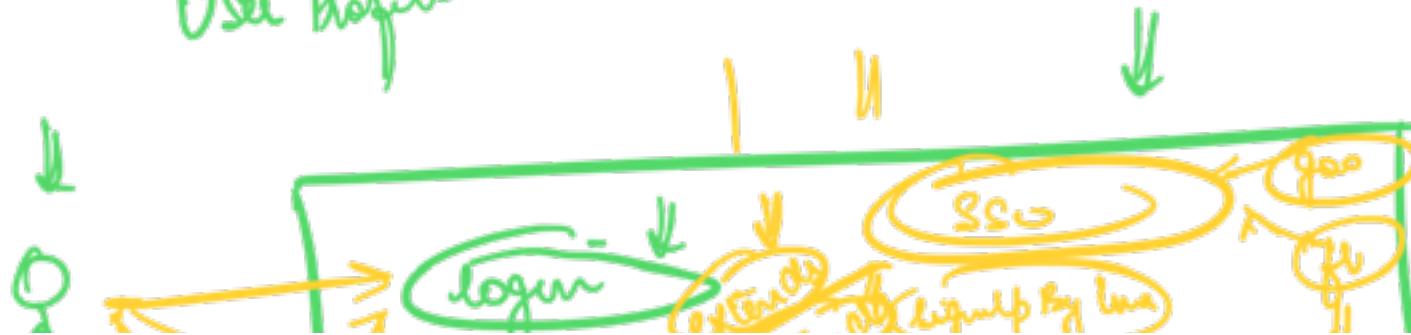


⇒ for a limited set of user

subset of product



User Profile





#### ④ Includes

To perform a use case A, if I need to perform  
 a use case B, say

$$A \xrightarrow{\text{includes}} B$$



logged<sup>4</sup>

Checkout()

→ Checkout()

pay()

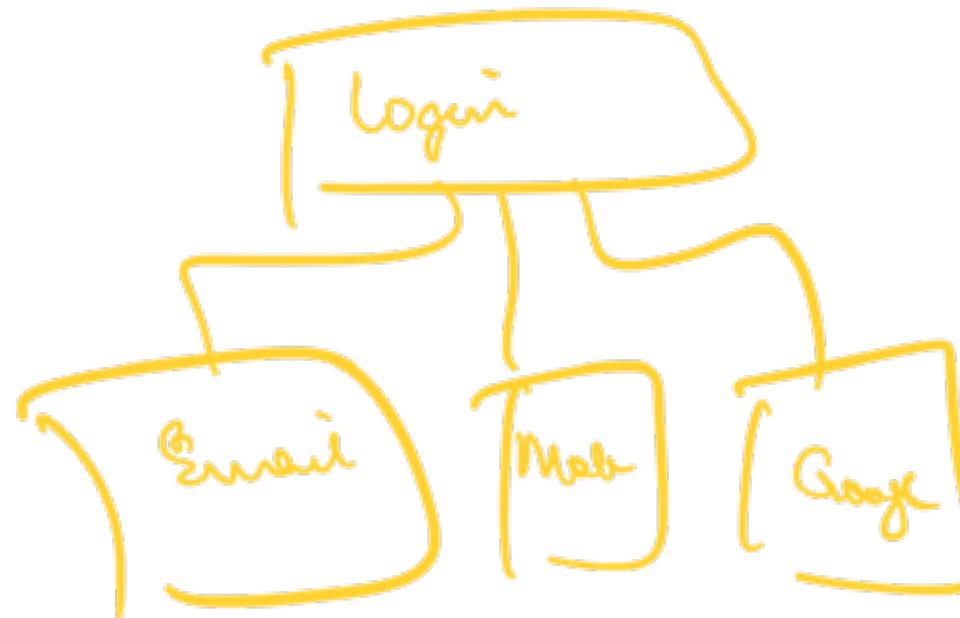
- ① When a user  $B$  is included in  $A$  when checkout, they pay the steps to perform  $A$  without  $B$ .

### ⑤ Extends

If for a particular use case  $A$ , there are multiple ~~specific~~ specialized types  $B, C$ .

→  $B, C, D$  etc extend  $A$

funcA()  
↓  
funcB()



,

### The Task

Draw a use case diagram for Scaler System

2 min

at least 1 include

→ Come with at least 5 use case → at least 1 extend

2) 2 actions

10:04 PM





Solve Question

Open

Output Run Code()

joinClass() {  
---  
---

}

,

getTAHelp() --> ==> ()

→  
→  
-

7

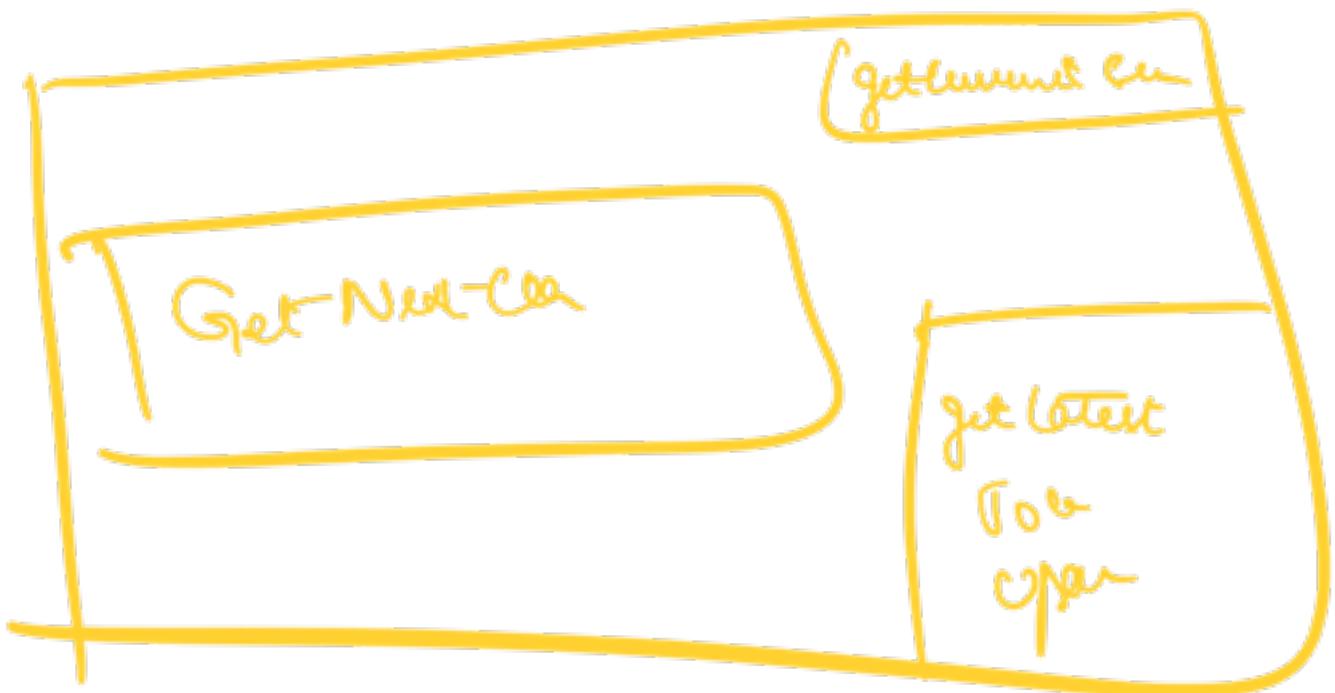
—



Schedule Mentor(Mentor, Student)

Try to think every use case as a function  
in the code :

inside the code of the ~~if~~ if you call



---

Say Boundary Use Case Actor

Includer extender

---

extends --> inheritance /  
interface implement

includes => the f<sup>n</sup> of parent will call  
f<sup>n</sup> of child (Nested f<sup>n</sup>)

Class Diagram =

Represent different entities ~~factors~~ that are  
there in my system

→ Class ←

→ Interface ←

→ Abstract Class =

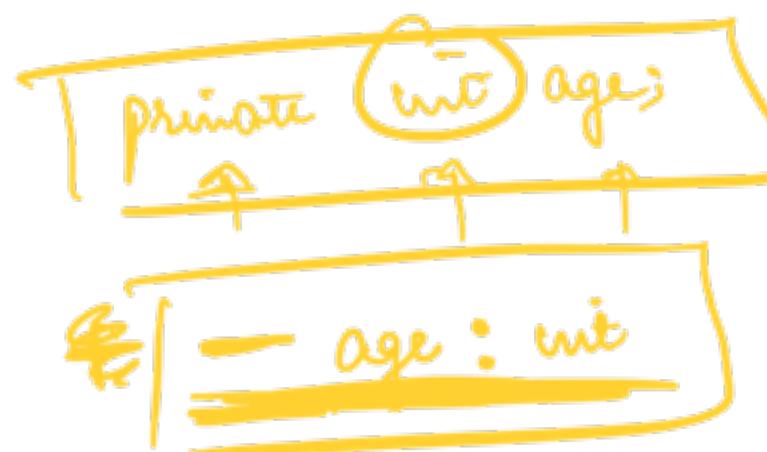
AND relationships between those entities

## Class



public      private      protected  
↓            ↓            ↗  
+            -            ↗

- ① Access Modifiers (public, protected)
- ② data type
- ③ name



private Animal animals

1.

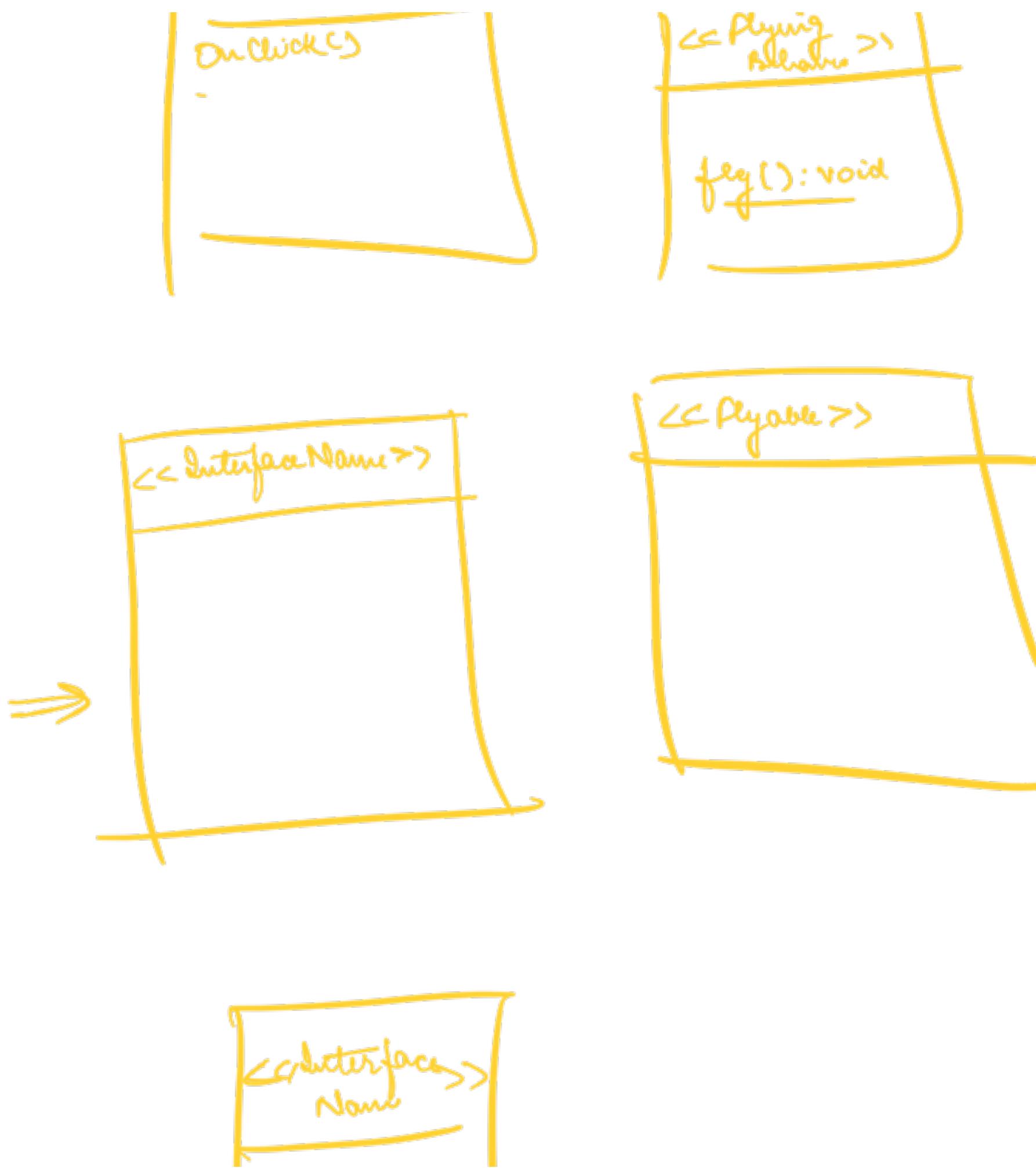
```
- Animal: Animal  
    |  
public Animal getAnimal (name, count)
```

+ getAnimal (String, int) : Animal

Access Modifier      Name ( Type of Parameter ) : Return Type

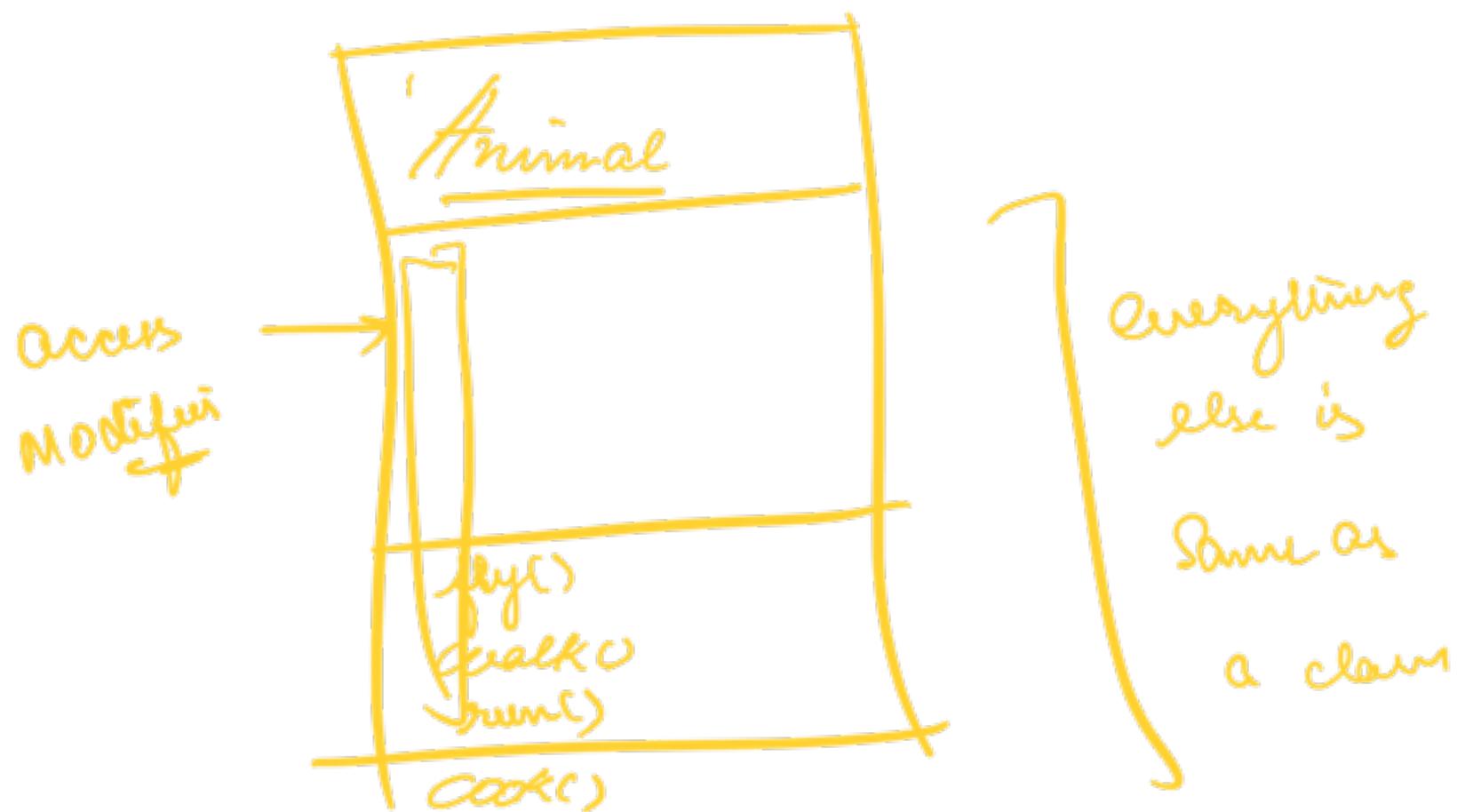
Method Signature

Button





## Abstract Class



is a class with one or more methods declared

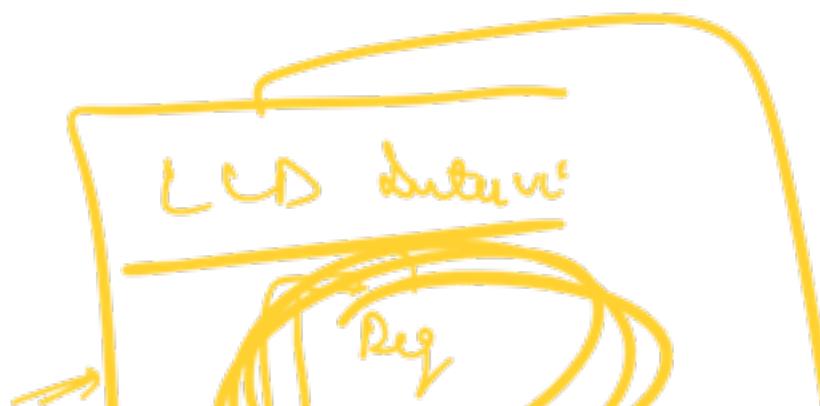
as abstract

↳ no implementation in class

→ Child classes have to implement

→ can't create instance of abstract class

Under score → static attr/ Methods





After Break

Rep Rel<sup>ns</sup>

① Diff Types of Rel<sup>v</sup>

① Aggregative (HAS-A)  $\rightarrow$  Comp  
to Aggr

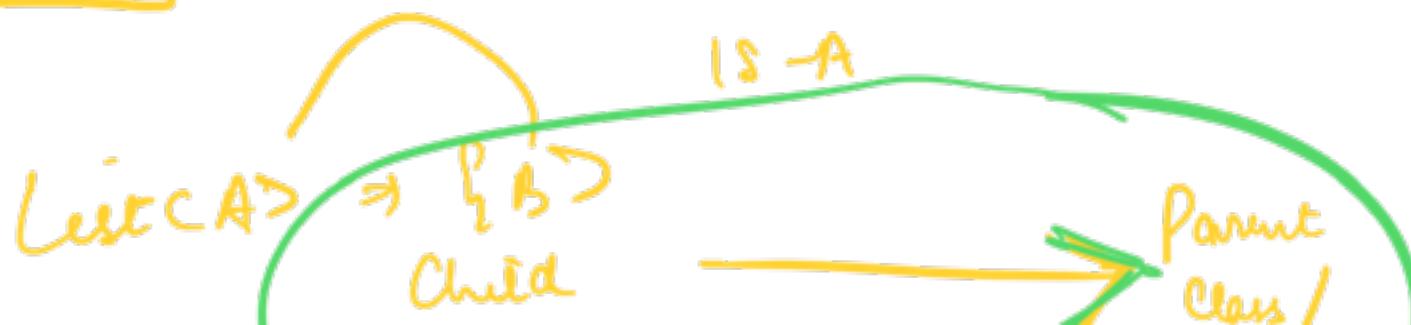
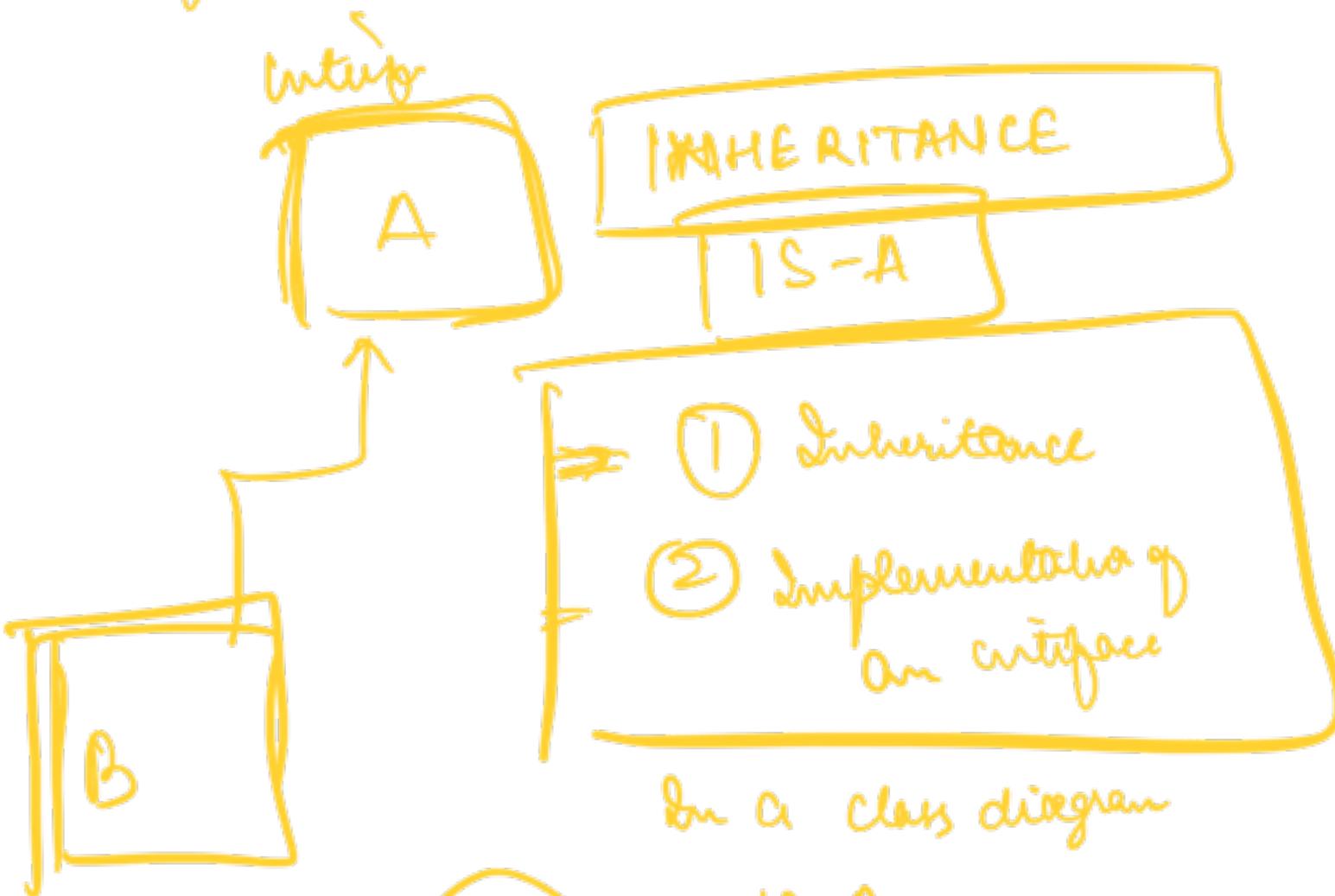
② IS-A

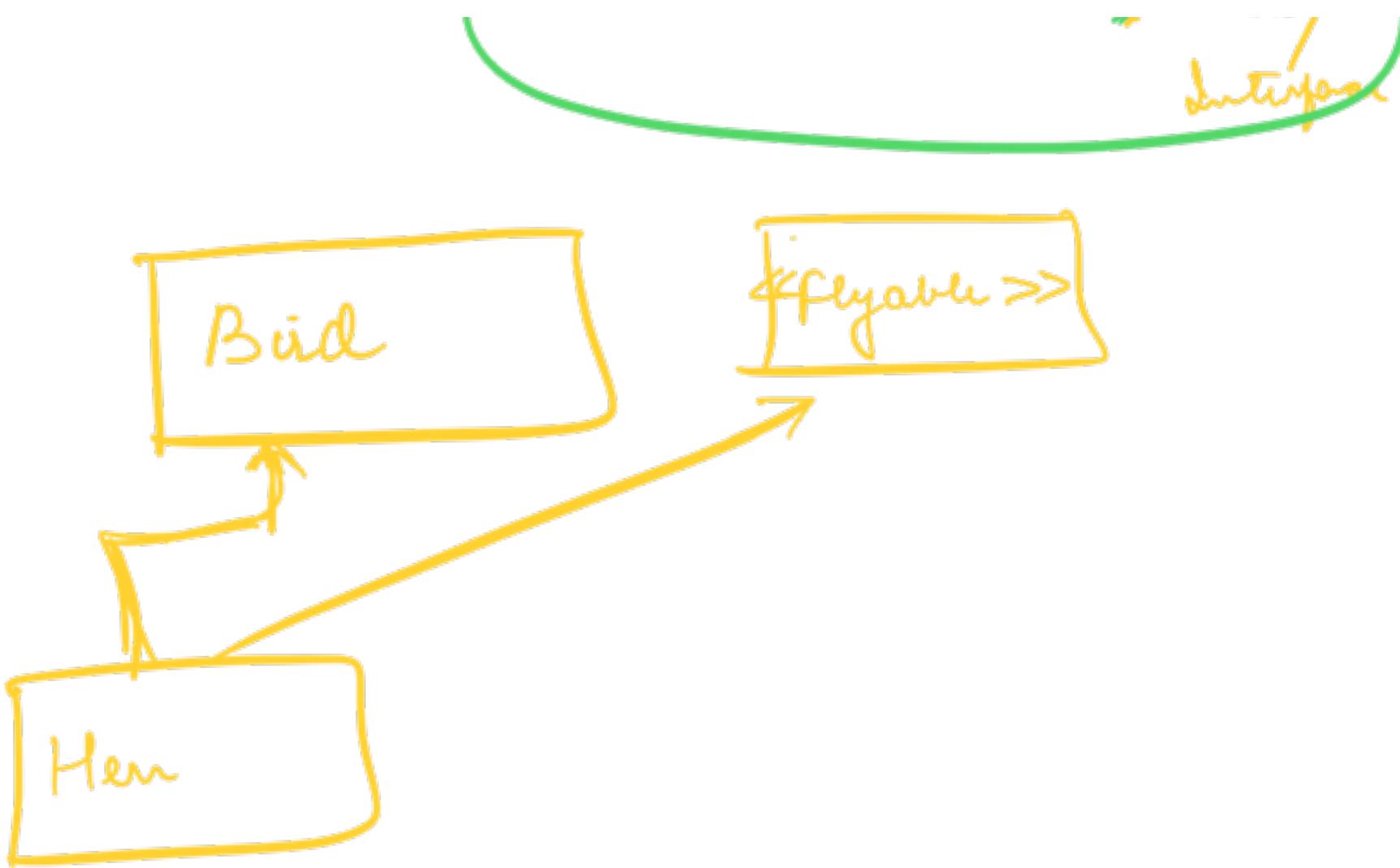
② How to rep in class diag

③ Diff generalization of Rel<sup>ns</sup>  $\rightarrow$  the with case studies

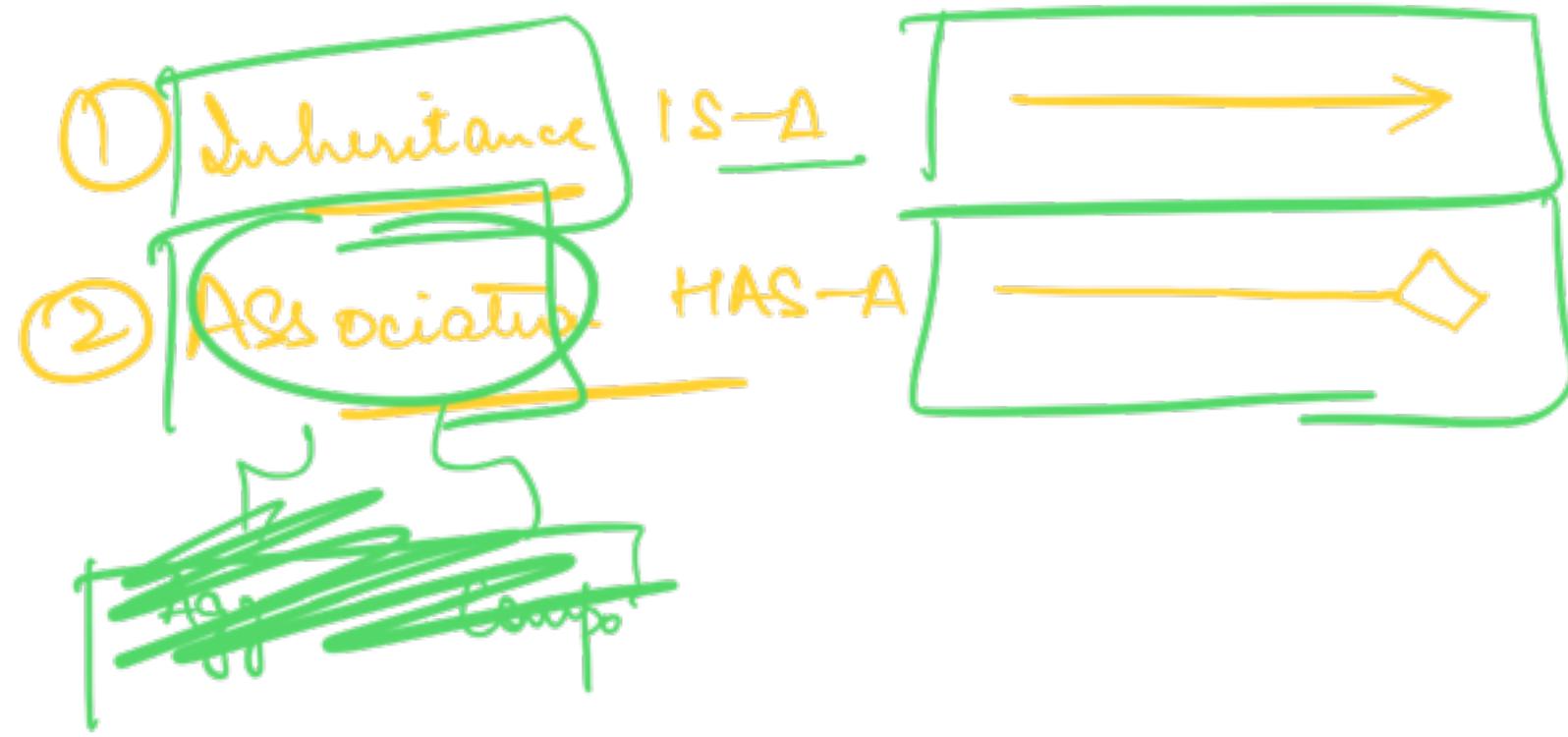
- ④ Scheme Design      ↗' wj -  
⑤ Code
- 

What kind of rel<sup>+</sup> b/w classes have classes





entity  
 When one ~~entity~~ has other entity as an  
 attrribute  
 Flying Bullet ← class  
 Bullet ← class  
 Hen ← class



Association can be of 2 types

① Composition ←

② Aggregation =

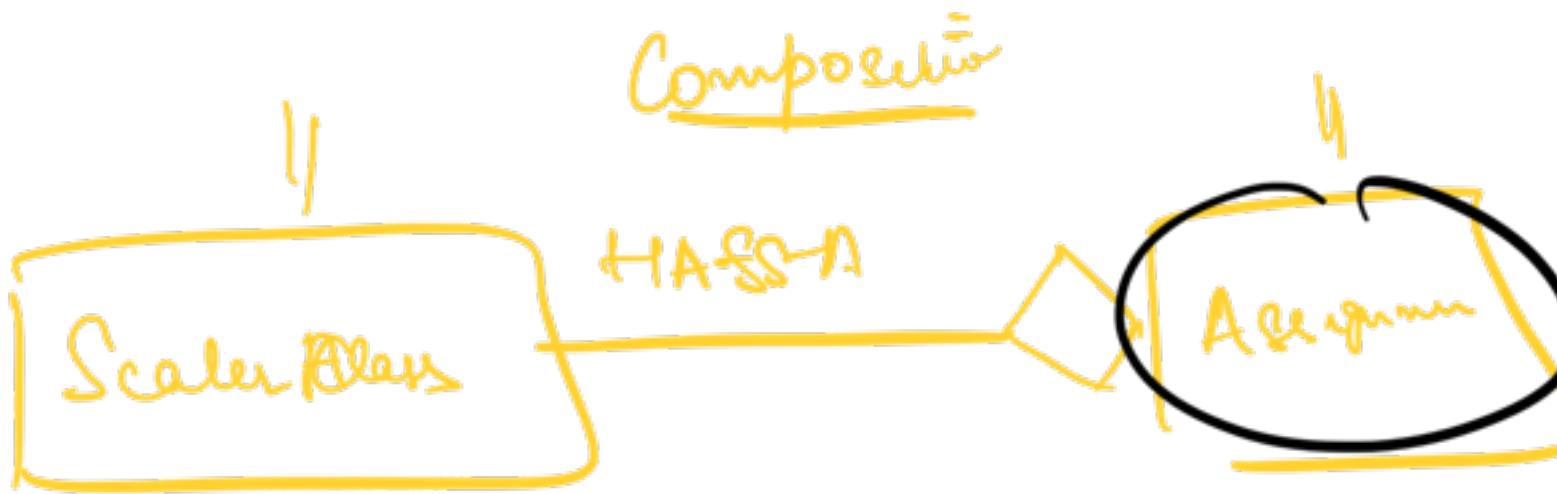
→ Creator

Composition : Class A has composition ref.  
to class B when ~~exists~~

Objects of B don't have any  
independent existence without A



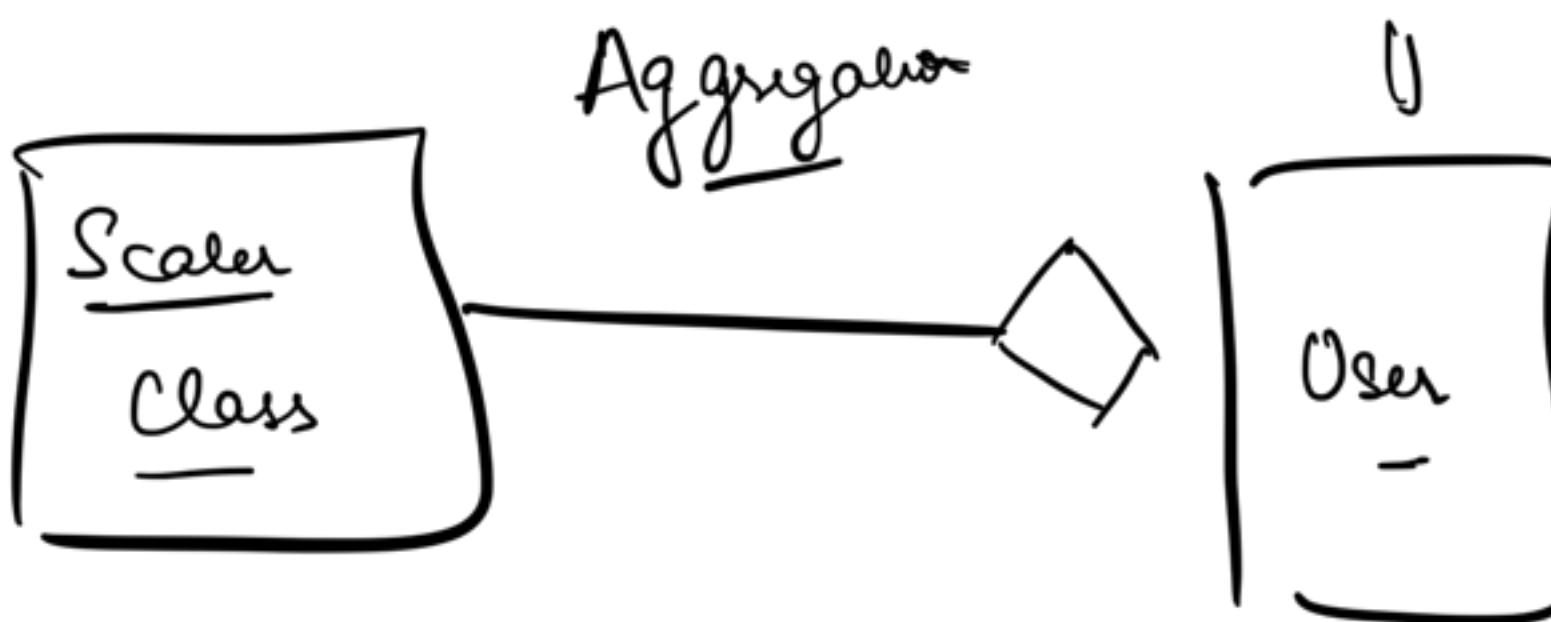
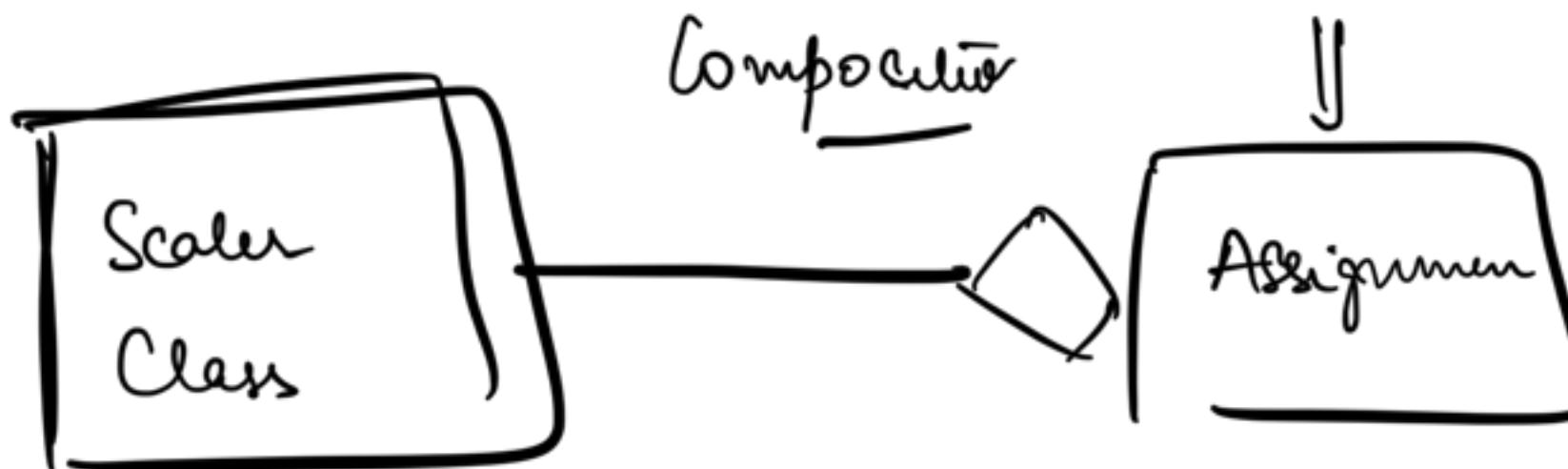
User instruction  
Assignment assignmen



Child has no  
independent  
exists

- 1) Inheritance (IS-A)
- 2) Association (HAS-A)  
→ Composition

→ Aggregation

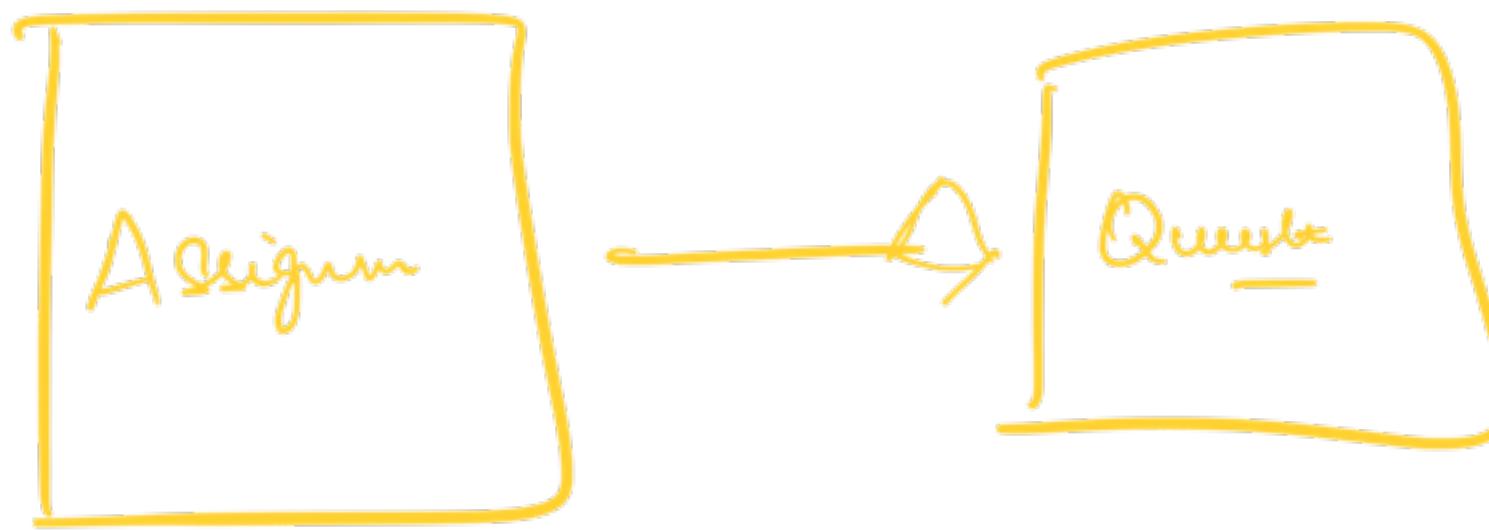


~~Aggregation~~

→ Child has an independent  
existence

~~Aggregate~~ → Collection



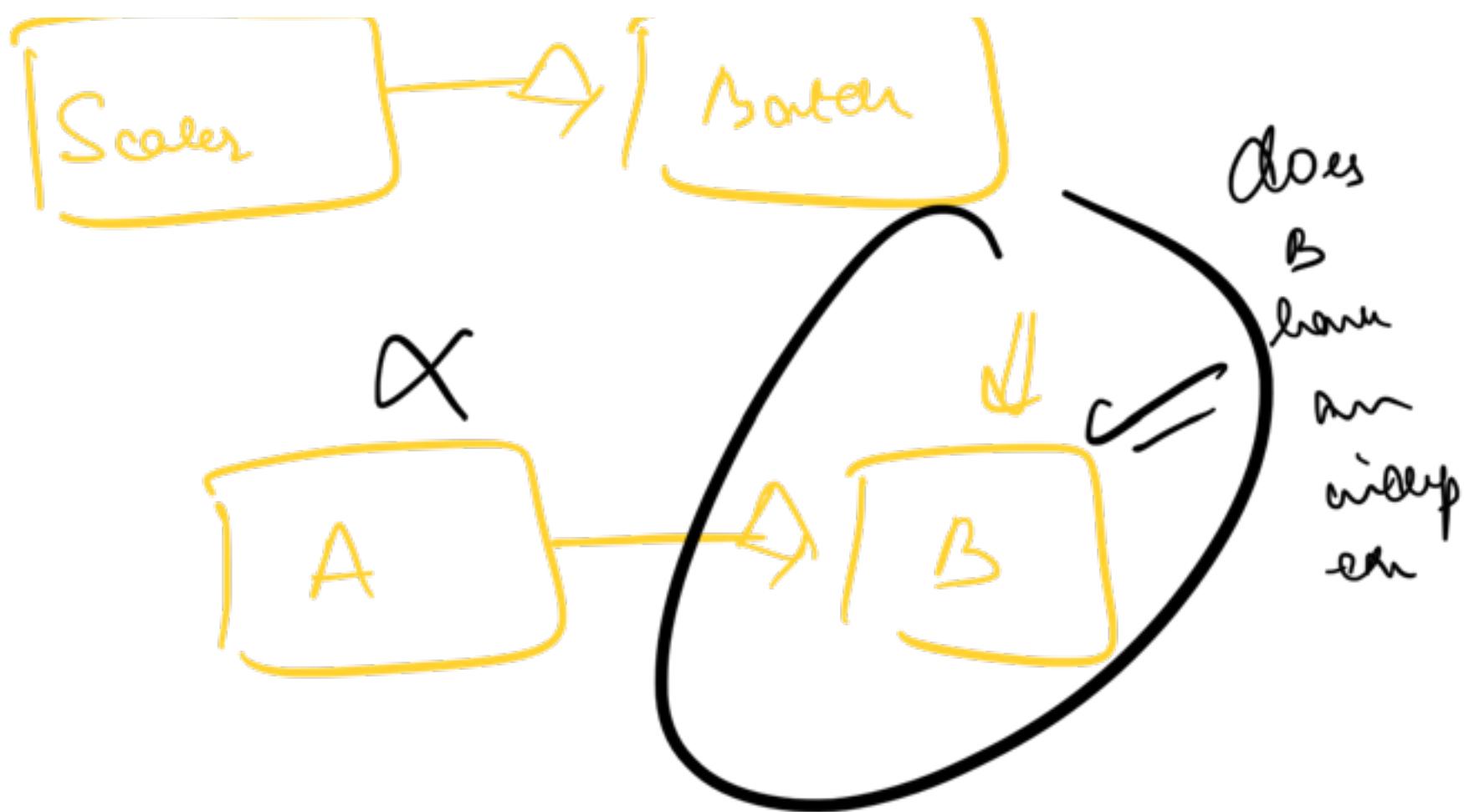


Class Assignment {

List < Quintal >

}



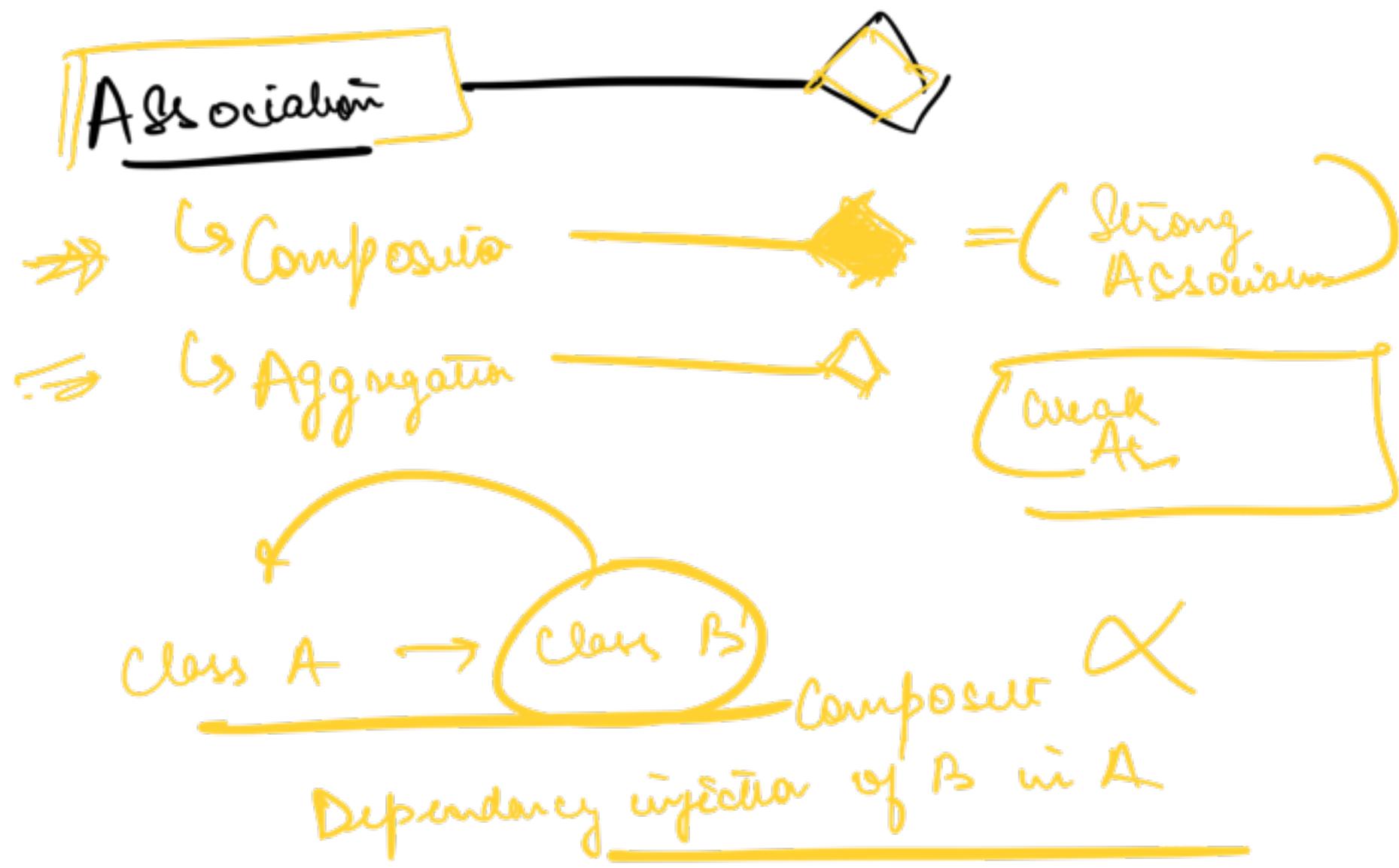


Composition

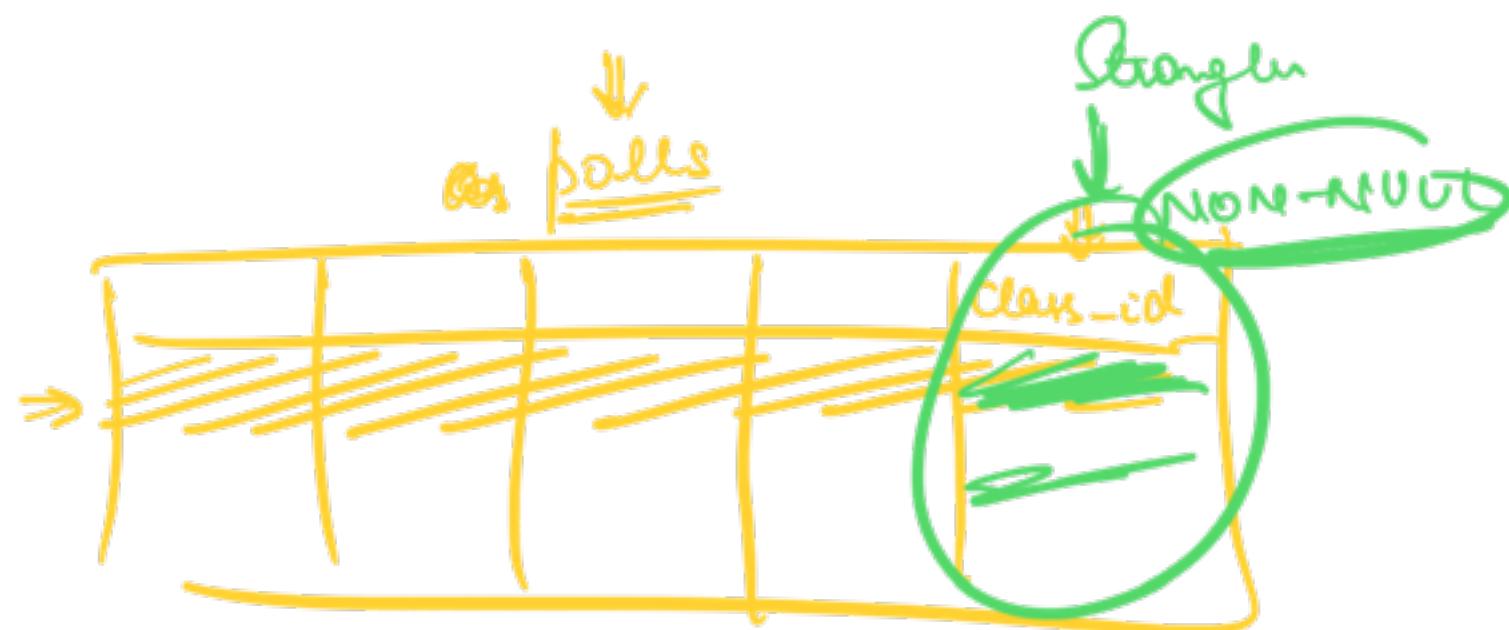


Compo





→ Create Assignment( )



Students


Class A {

B

-

Class A }

List < B >

}

7

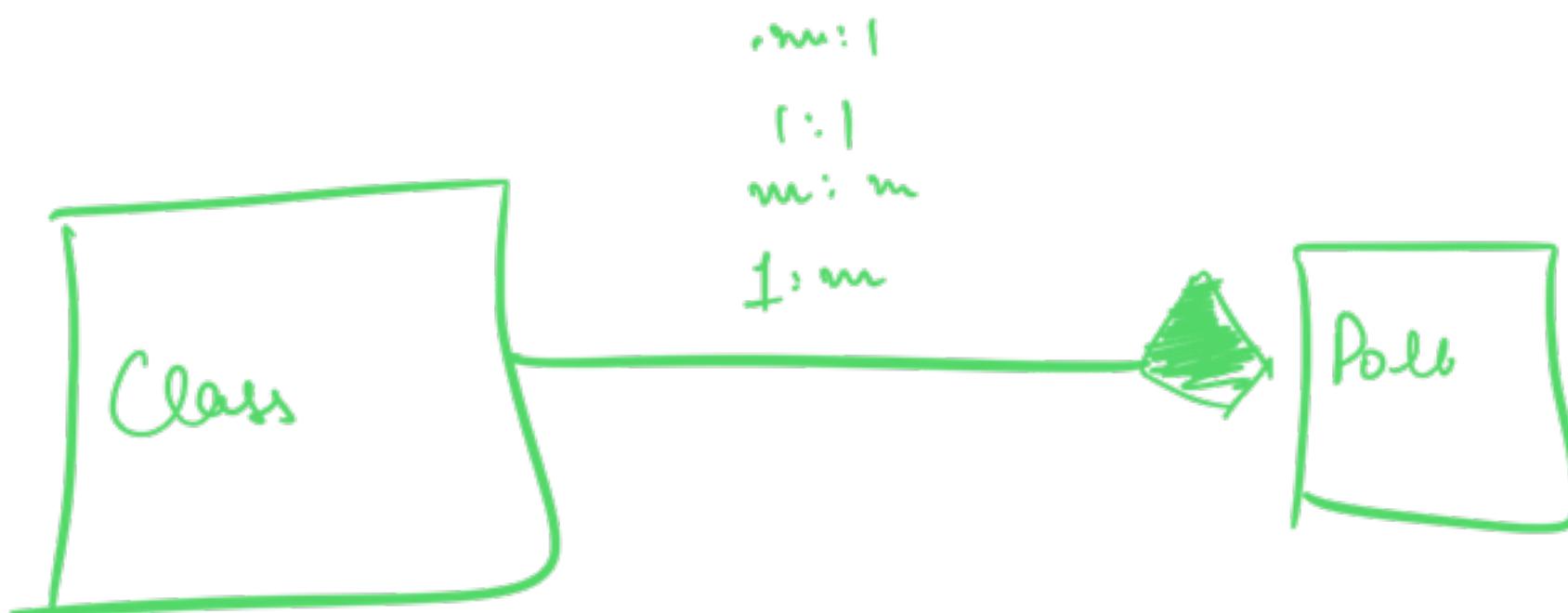
Cardinality : Degree of ~~no~~ HAS-A rel<sup>2</sup>  
between 2 entities

How Many of  
A

associated  
to

How Many  
of B

$l : l$   
 $l : M$   
 $M : l$   
 $M : M$

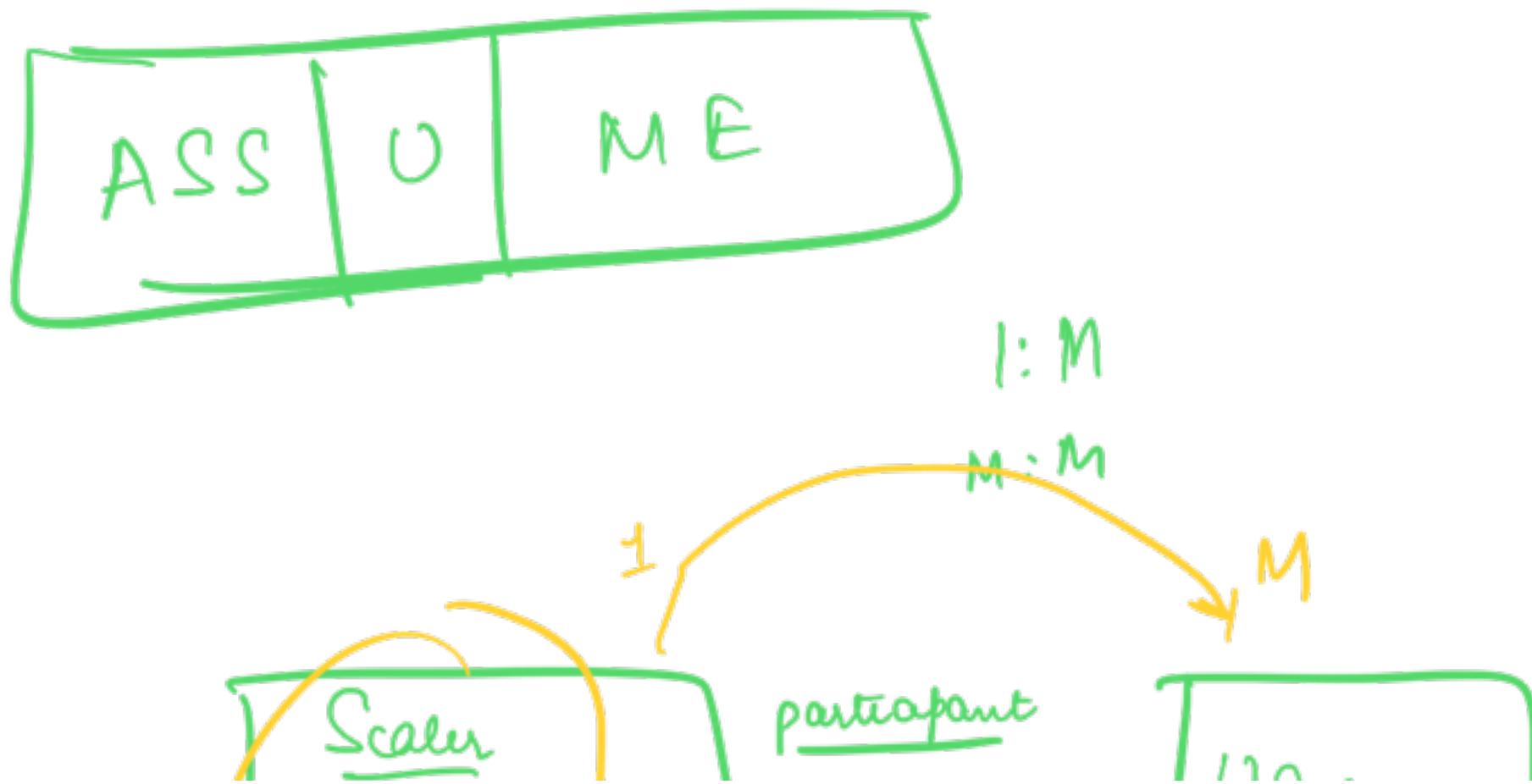


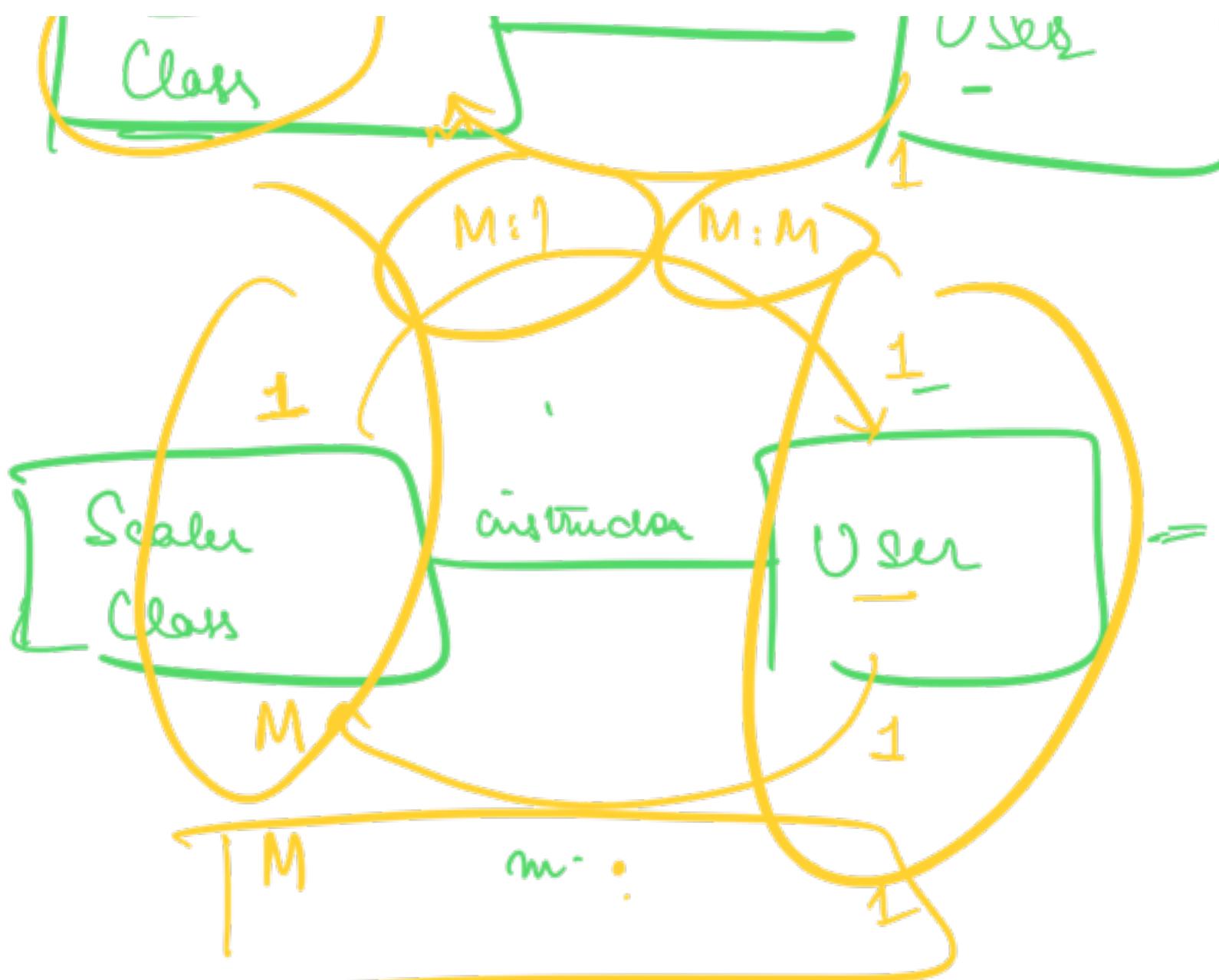
$l : M$   
 $M : M$

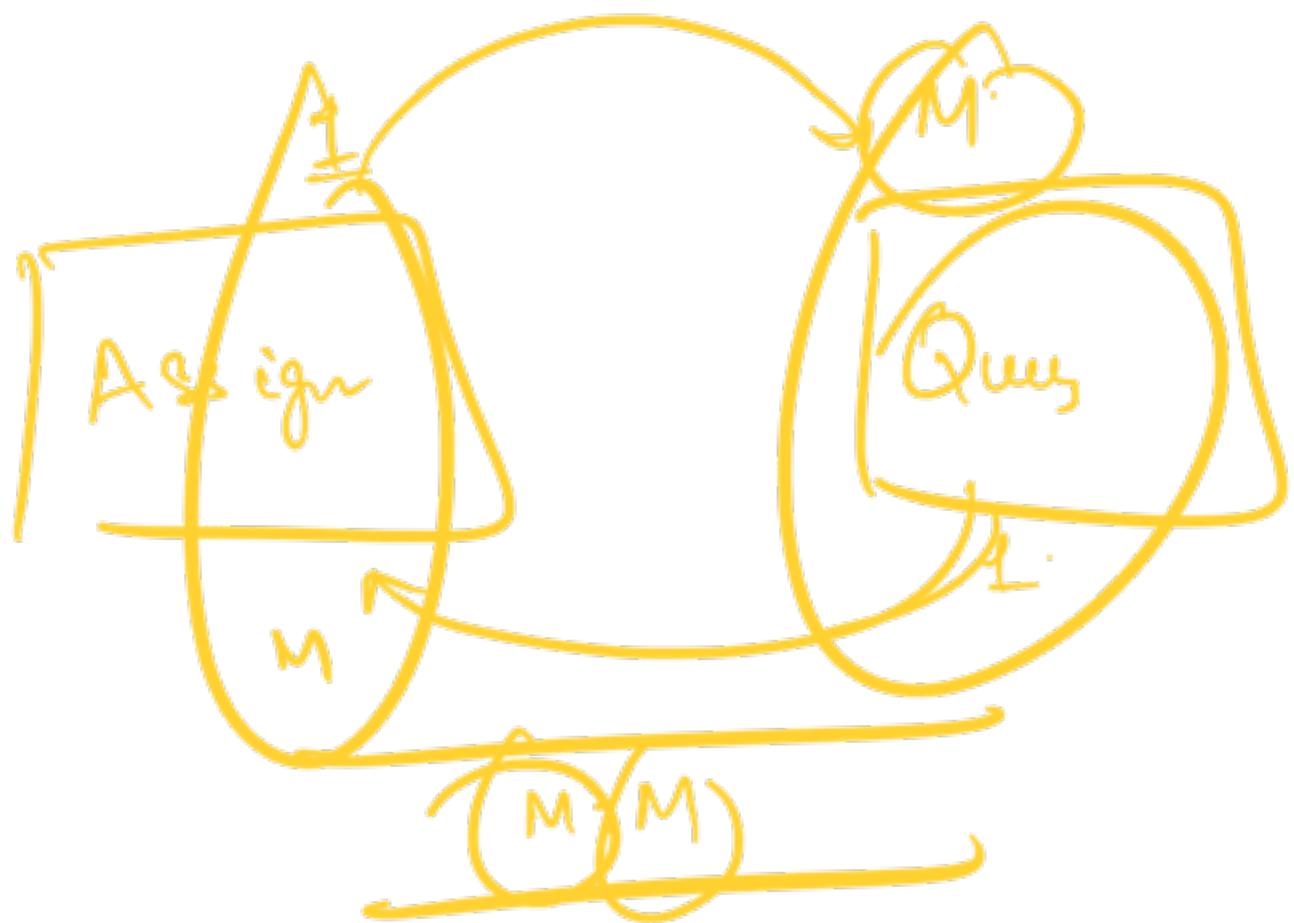
Scalar  
class

User

- ① B/w 2 entities there can be more than 1 rel<sup>n</sup>

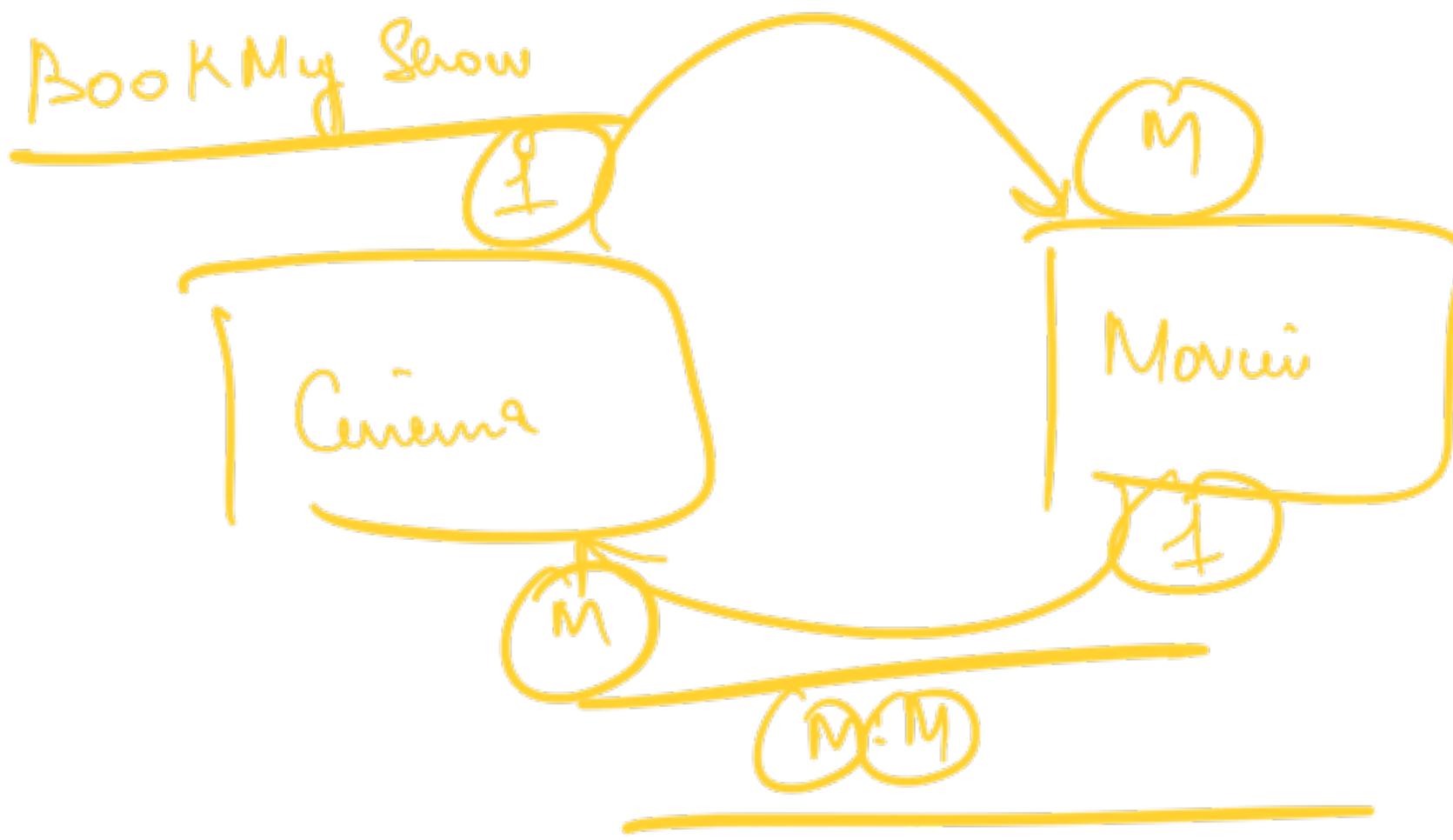
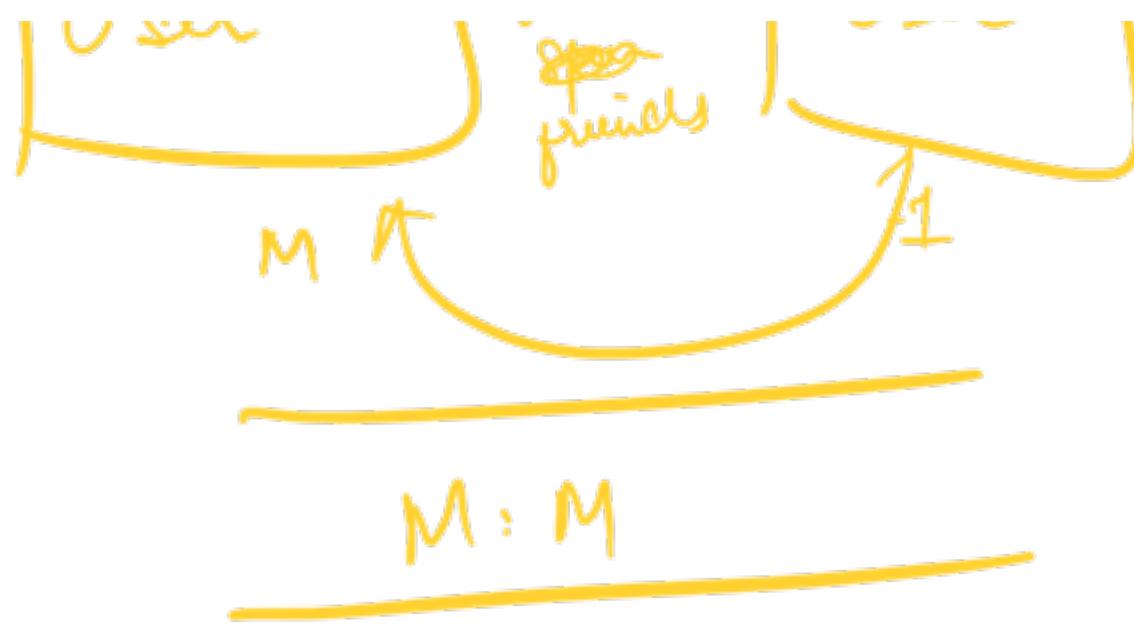






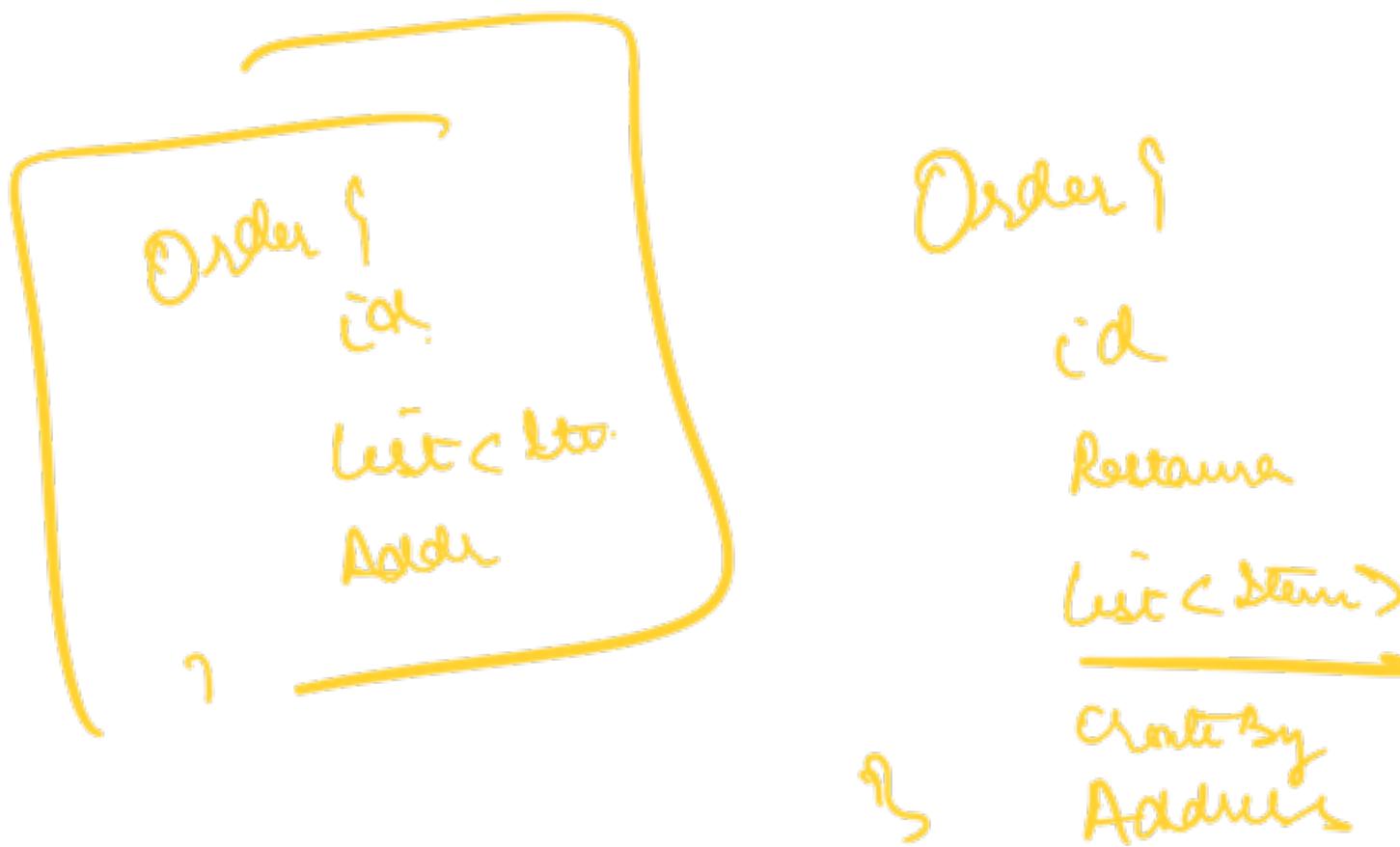
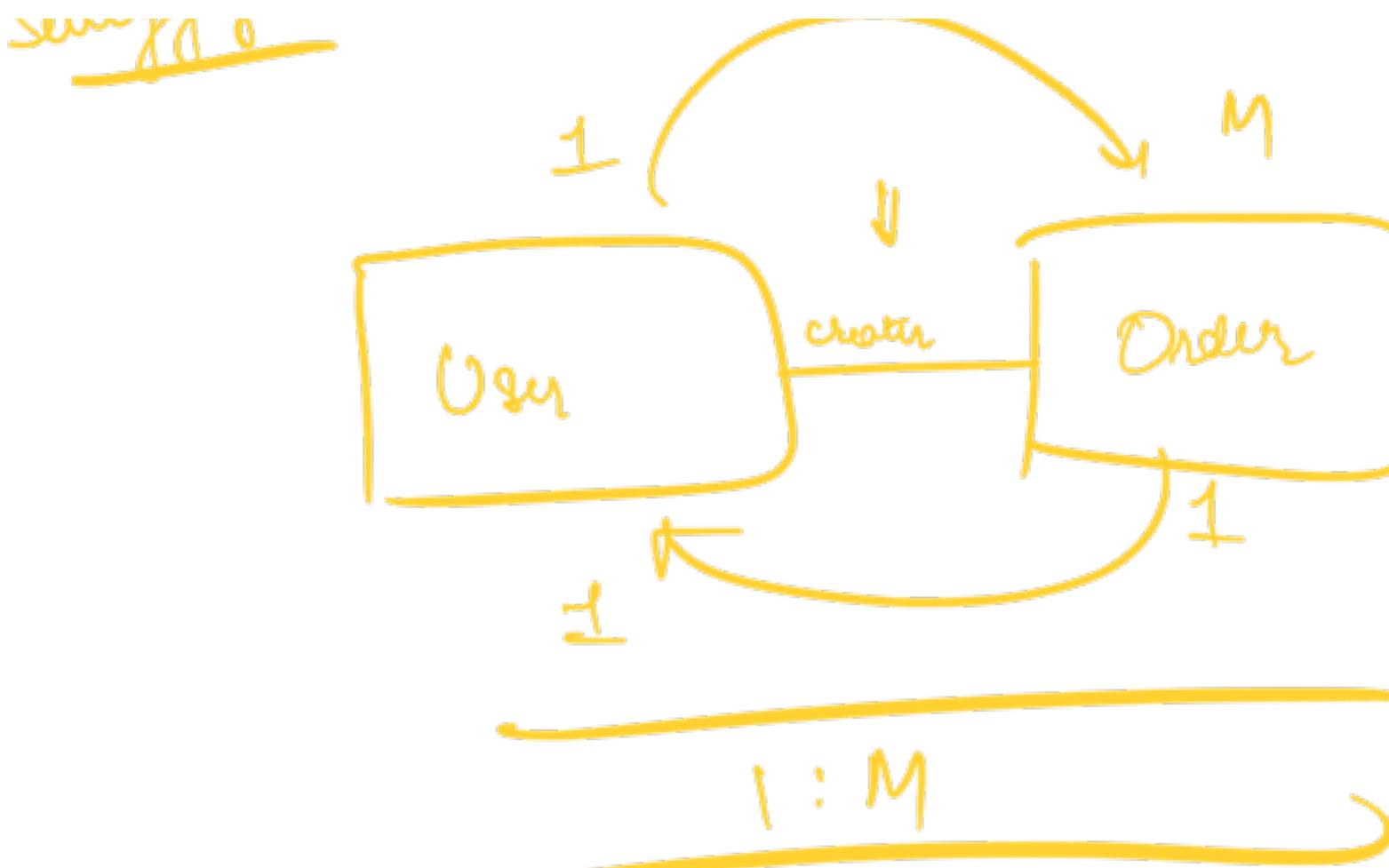
Facebook





Classmate

~~seen 80%~~



## Next class

- How to up each cardinality of  $\sim$
- Code in Spring Boot

- 
- Use case diag
  - class diag
  - Compre/s agree
  - how to find cardinality
    - ↳ clarify the type of rel

---

HW

- si. anal