UML Diagrams

and

Schema Design

Agenda

① UML Diagrams

ⓐ Use Case Diagrams

ⓑ Class Diagrams

② Schema Design

① How to create tables

② How to formulate rel^n b/w tables

Composition

Aggregation

③ How to rep rel^n in DB

④ How to find Cardinality of rel^n

SQL DB Schema Def tables

Mapping Tables

ORM

③ Practical Scenario

Scale's DB

→ Code using Spring Boot

→ Colleagues

→ Business Analyst

→ PM

→ Architect

→ EM

→ Team Lead

Diagrams

→ allow to visually convey. an otherwise difficult to convey concept

→ DB Team

→ Juniors

Google MNC

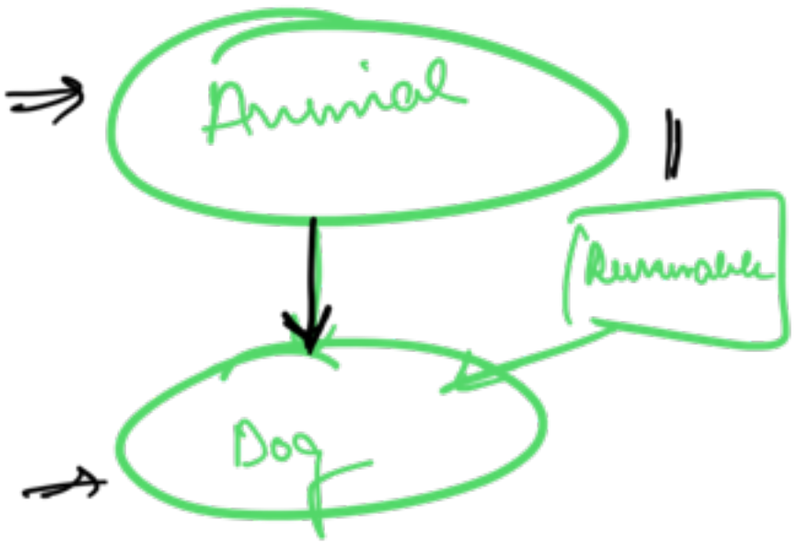
Zoo

Animal → class

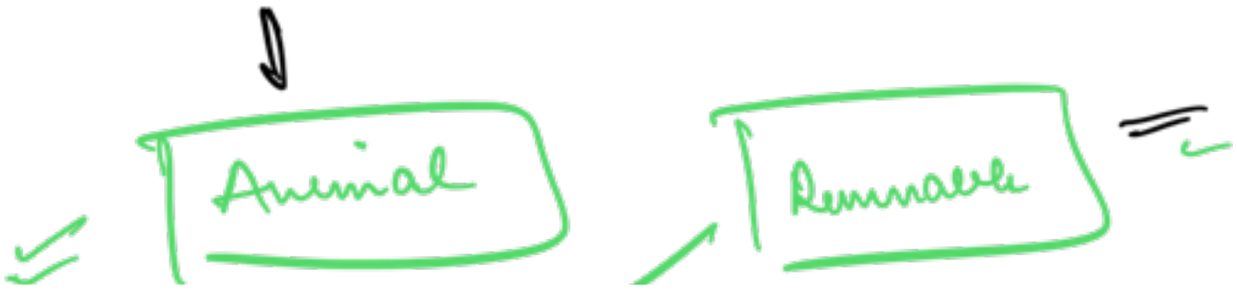
Dog → class

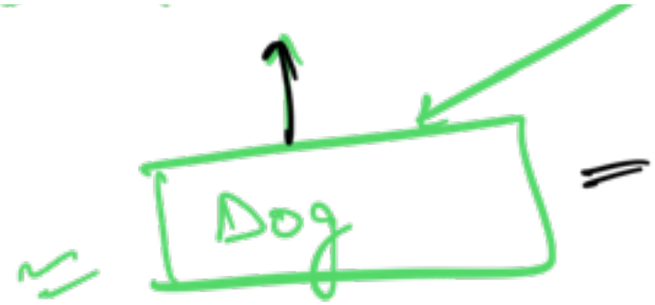
Runnable → interface

India

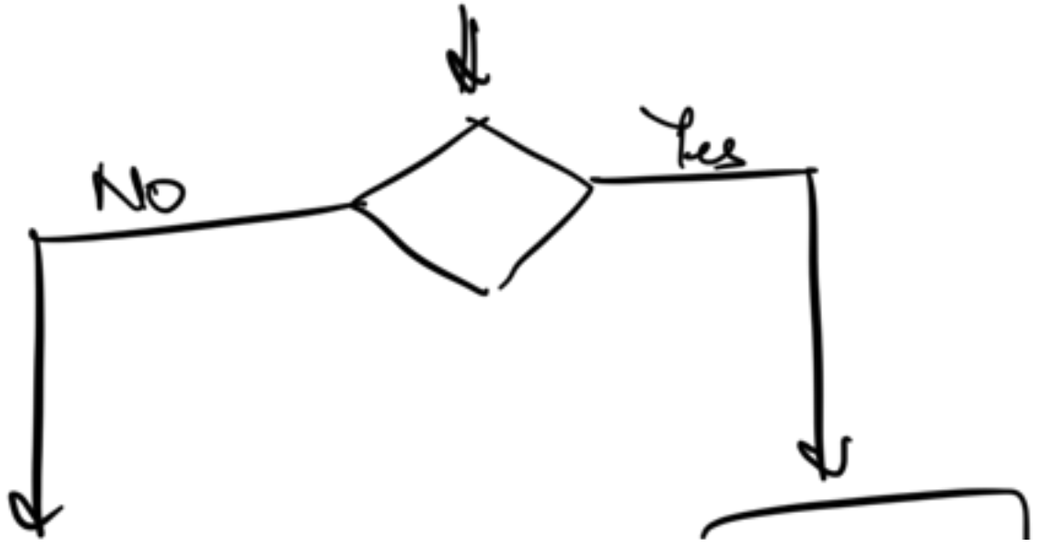


VS



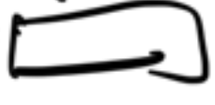


UML ⇒ Unified Modeling Language



Right Click →

Show Diagram



2 Types of UML Diagrams

(1) Behavioural UML Diag

→ What are the diff features of an app^{n}
→ Working of those feature
→ Who is going to use the system

- Use Case Diagrams
- Seq Diagrams

(2) Structural UML Diag

→ How objects are Modelled
→ How projects are Modeled

- Class Diagrams
- Component Diagrams

① Use Case Diagrams

⇒ Diff things / use cases that are supported by my application

① Actors : people / systems who are going to interact with the system

② Use Cases : Diff features that are supported by my app

③ System Boundary : includes every feature that lies inside my app

4 Extends Often a specialization of a feature

5 includes

To draw use case diagram:

(1) Create System Boundary

(2) Identify all actors of the system.

(3) Draw all use cases as ellipse

When a person signs up they should be sent an OTP

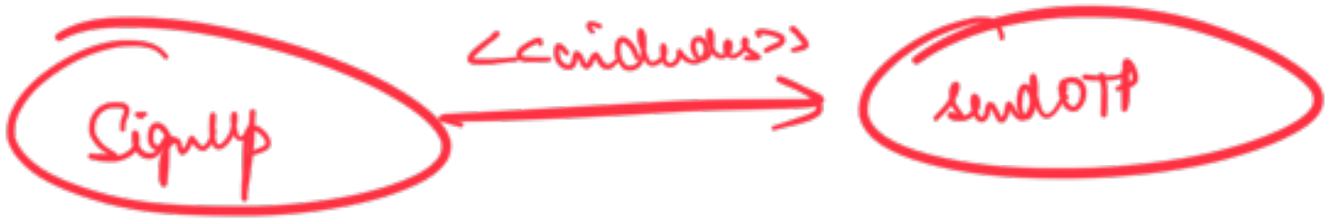
(4) Identify rel^n b/w actors and use cases

(5) Identify inter use-case relationships.

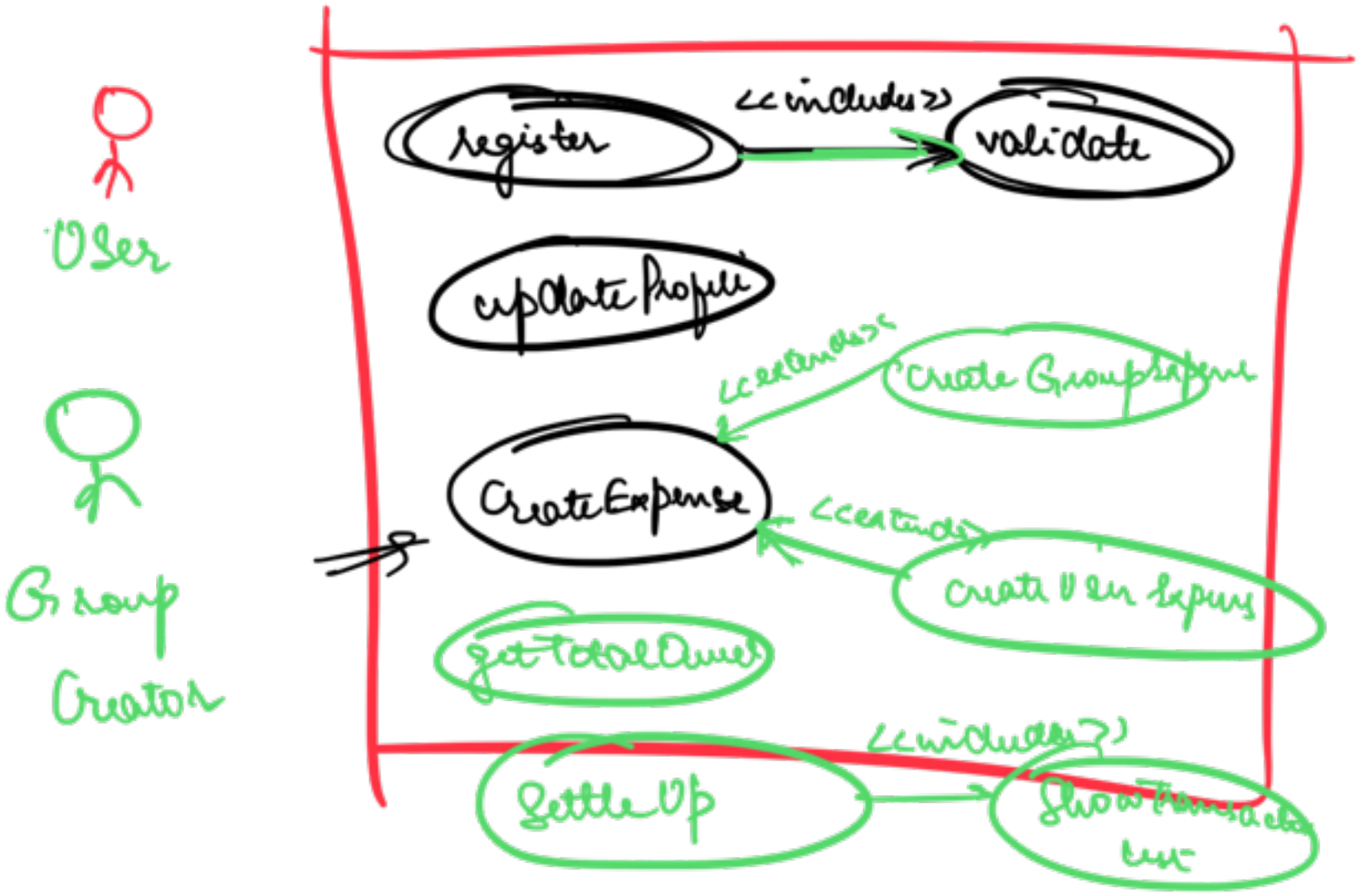
(extends and includes)

9:55 PM —

1. What are diff Actors
2. What are diff use cases
3. Any 1 example of extends
4. Any 1 example of includes.



When ——— then ═══



Assignment

Assume you are an Engineer at ~~Paytm~~ PayTM.

1. Create a set of requirements (atleast 8)
2. Create a use case diagram for those

Req.

Class Diagrams

Blueprint of all the entities of your software System.

- Classes
  - fields
  - methods
  - Access modifiers
- interfaces
- abstract classes
- rel^{s} b/w classes
  - Inheritance (IS-A)

↳ Association (HAS-A)

class A extends B { } ⇒ Inheritance (IS-A)

class A {

B b; ⇒ Association (HAS-A)

}

var_Name : Type

Class

int run(String type, int speed, String —)

}

run(String, int, String): int

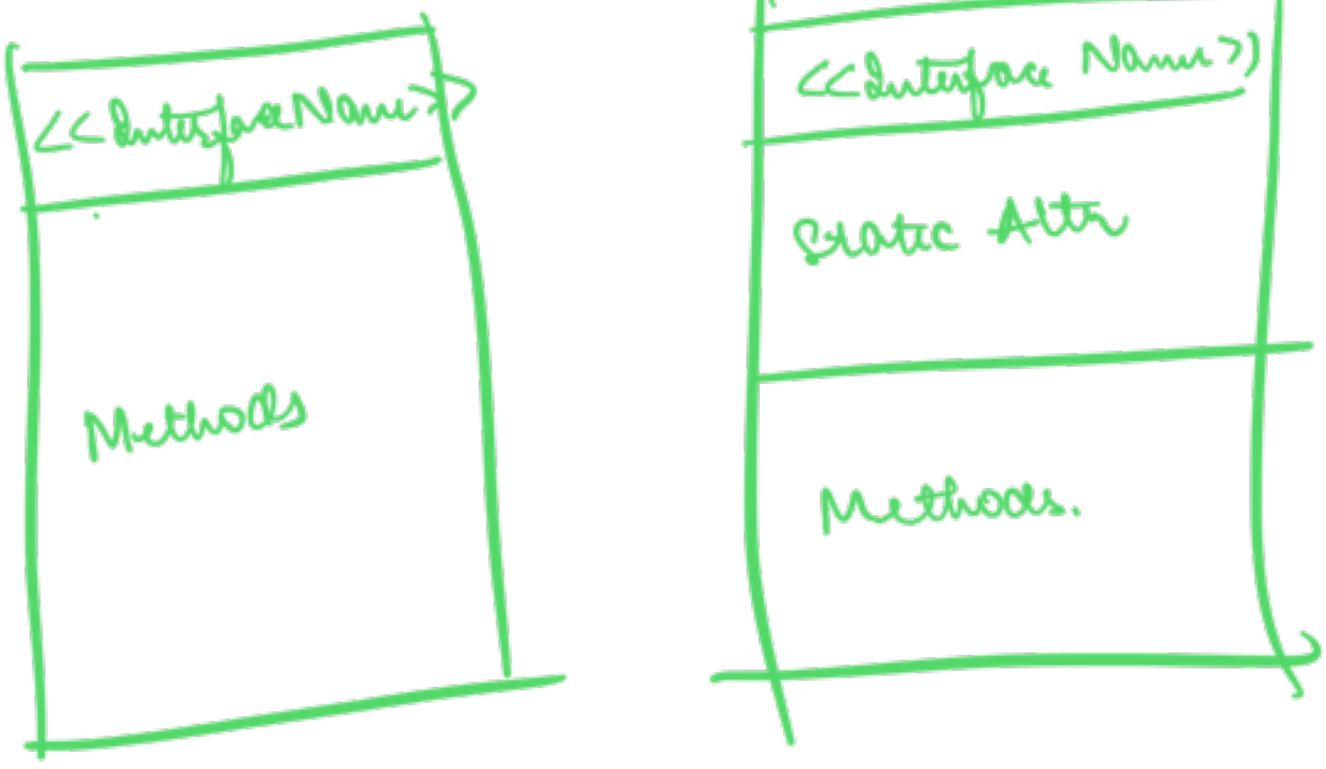
public : +

private : —

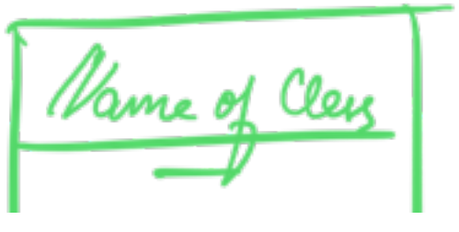
protected : #

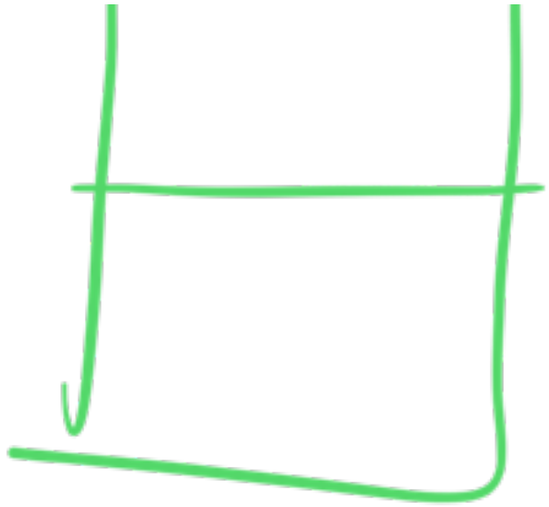
default : ~

Interface



Abstract Class





A IS-A B

Relationships

IS-A → Inheritance

IS-A → Implement

Association

HAS-A → Composition

HAS-A → Aggregation

A extends B {}

A implements B {}

Bird implements Runnable

Association

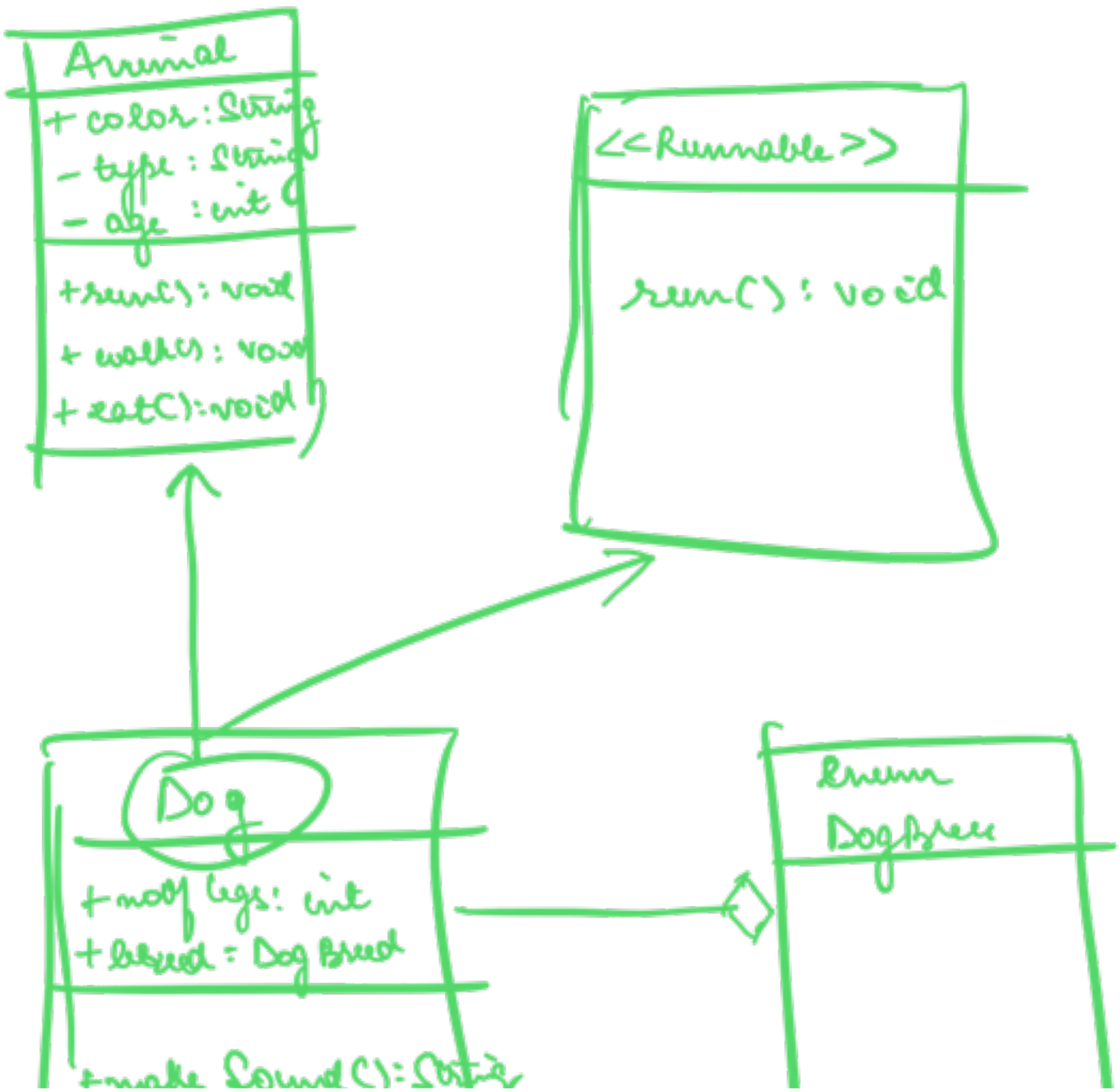
class A {

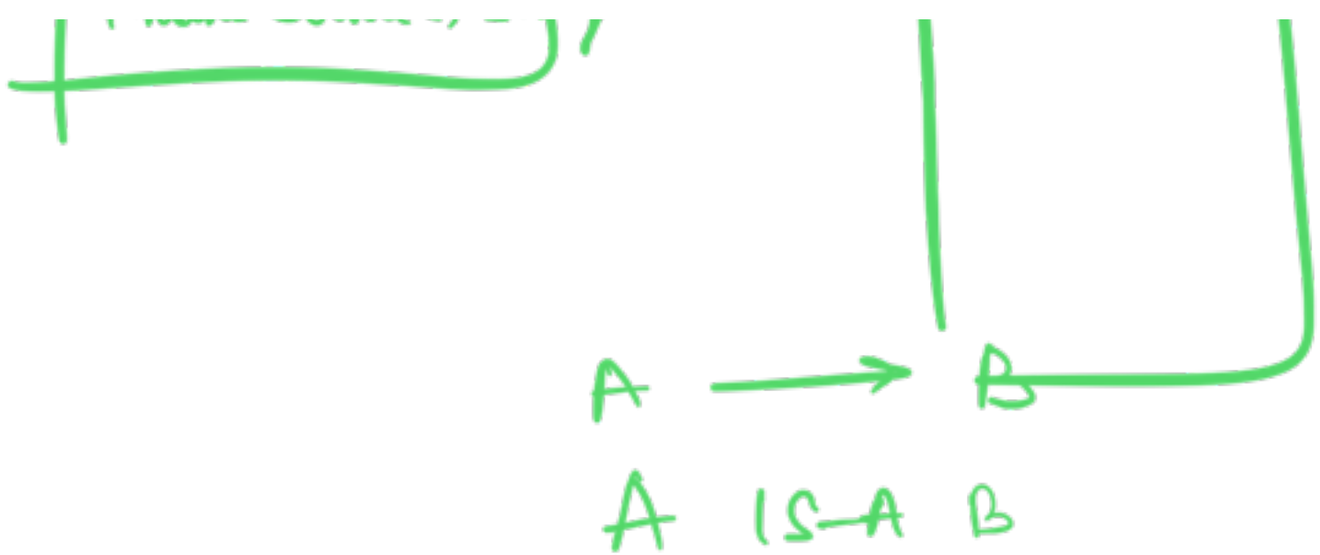
B b;

}

A HAS-A B

A ——◇—— B





A IS-A B

10:30 PM

Schema Design

1. IS-A
2. HAS-A (Association)
   - ↳ Aggregation ⇒

↳ Composition

A ⟶ B Aggregation

A is collecting objects of B

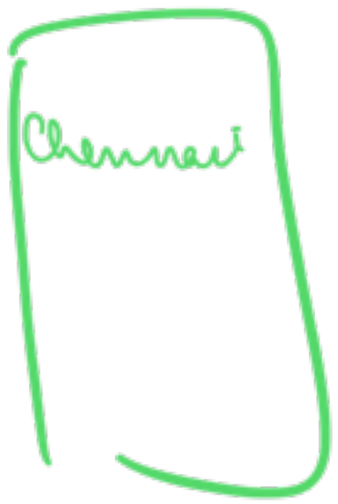
⇒ Objects of B are independent of A

Aggregation

A is associated to B

If the objects of B have a well defined
use in isolation

User — City



Composition

A

B

A ——▷ B

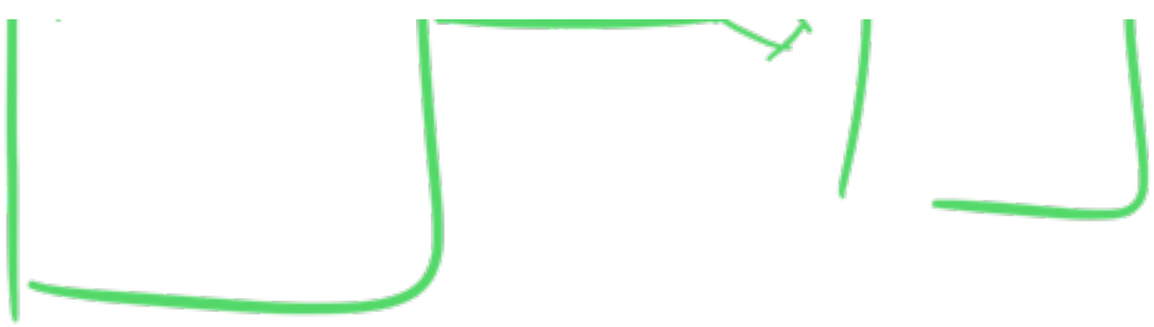
① Composition
② Aggregation

whether B will exist without A or not

House

Composition

Room



House 9

Room room;

House() {

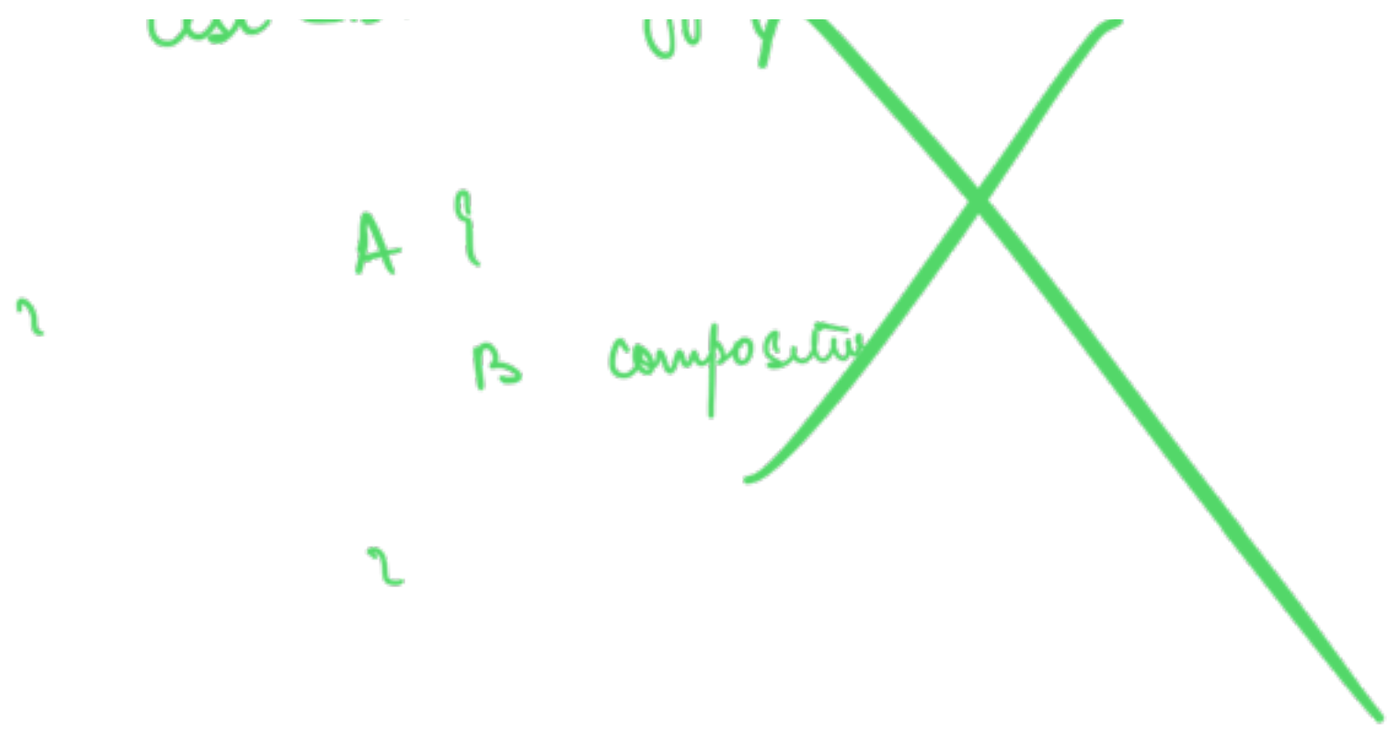
this . room = new Room()

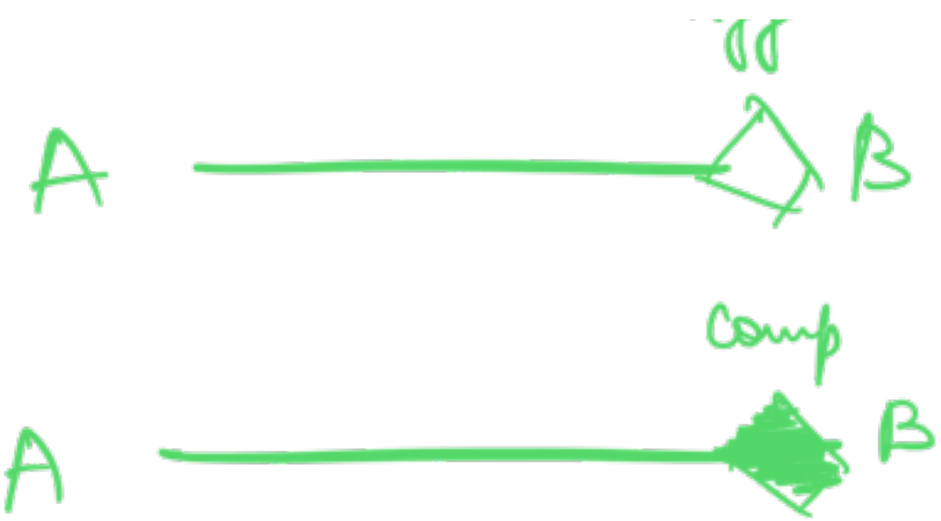
}

}

A {

1 : 1 <B> aggregation


A
B



Schema Design

Batch {

List < Student >

}

Movie Show {

List <Ticket>

}

Ticket {

Movie

}

Schema Design

→ SQL DB

→ Identify all entities

→ Identify the cardinality of rel^n b/w

diff entities

→ Associate mapping b/w diff entities

Case Study

≈ ① Find all the entities in the system

Associations ② for every entity create a table ≈

③ If an entity has other entity as an attribute

→ ⓐ Find the correct way to store that association in table

For other attributes, create a new column for every attr.

Case Study

① There are students in Scaler. Every student should be having Name, mNo, password.

```
class Student {
    String Name,
    String mNo,
    String password
    Long id;
}
```

Students

id	name	mNo	password

A

As

Str. of Name

EXACTLY ONCE

⇒ Every Student can opt to at most

1 Batch

⇒ Every Batch has a Name.

Students

sid	Name	mNo	pass	BatchName
1				ABCD
2				ABCD
3				ABCD
4				

XYZ

Inconsistent

class Student {

id
Name
mno
pass
batchName

1. SRP
2. Inconsistency in update
3. Redundancy
4. A batch can only be created if atleast one student exists
5. Unnormalized

1NF → 2NF → 3NF → BCNF

Pros
# of tables is less

Students

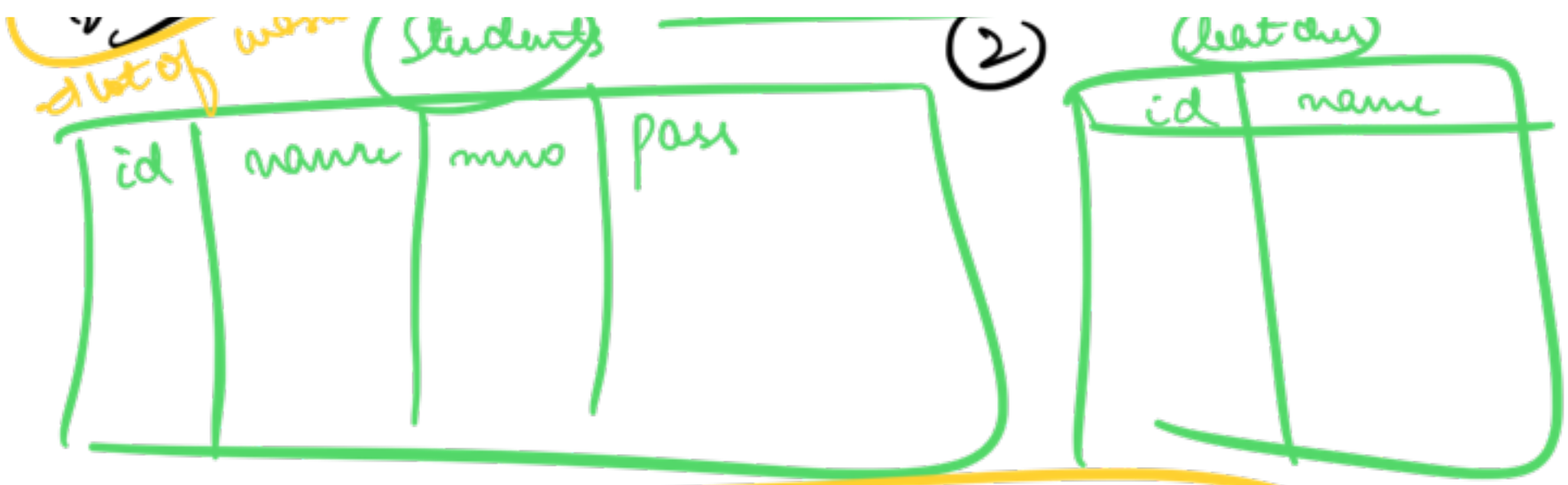
id	Name	mno	pass	batch_id
				NULL
				NULL

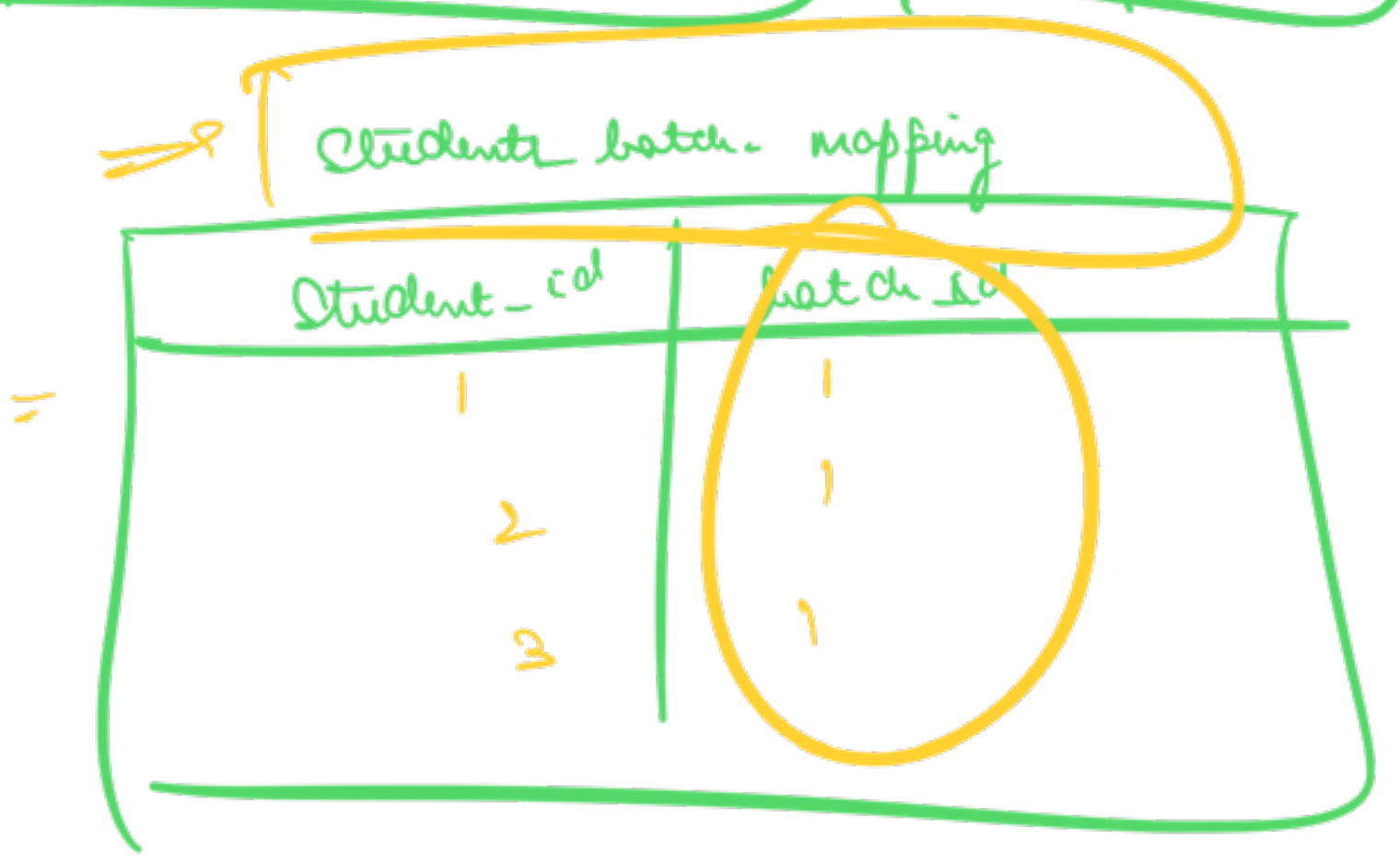
batches

id	name

1M users are in 1 batch
1 batch has 1M users

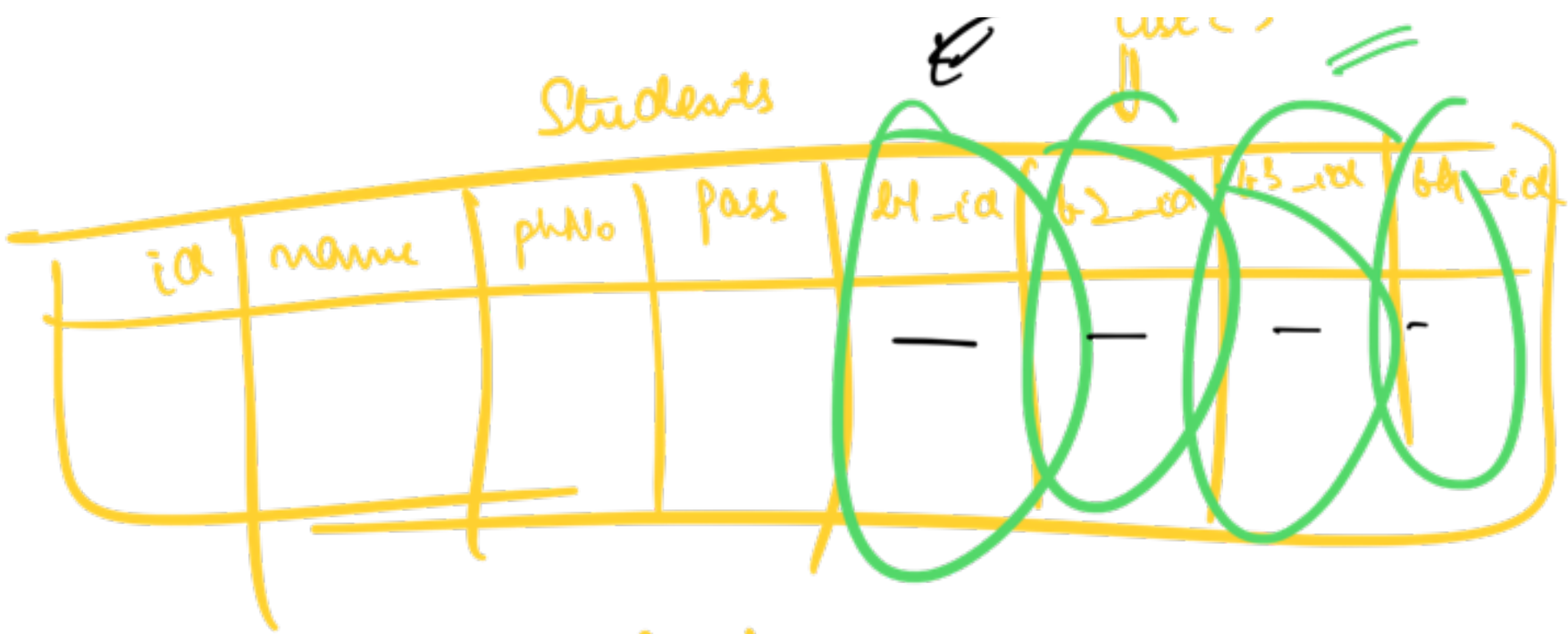
→ Sparse table





(V3) Every Student can belong to EXACTLY 4 Batches.

Handled in DB side



Query : Get all Students belonging to Batch 1

It associated to more than 1

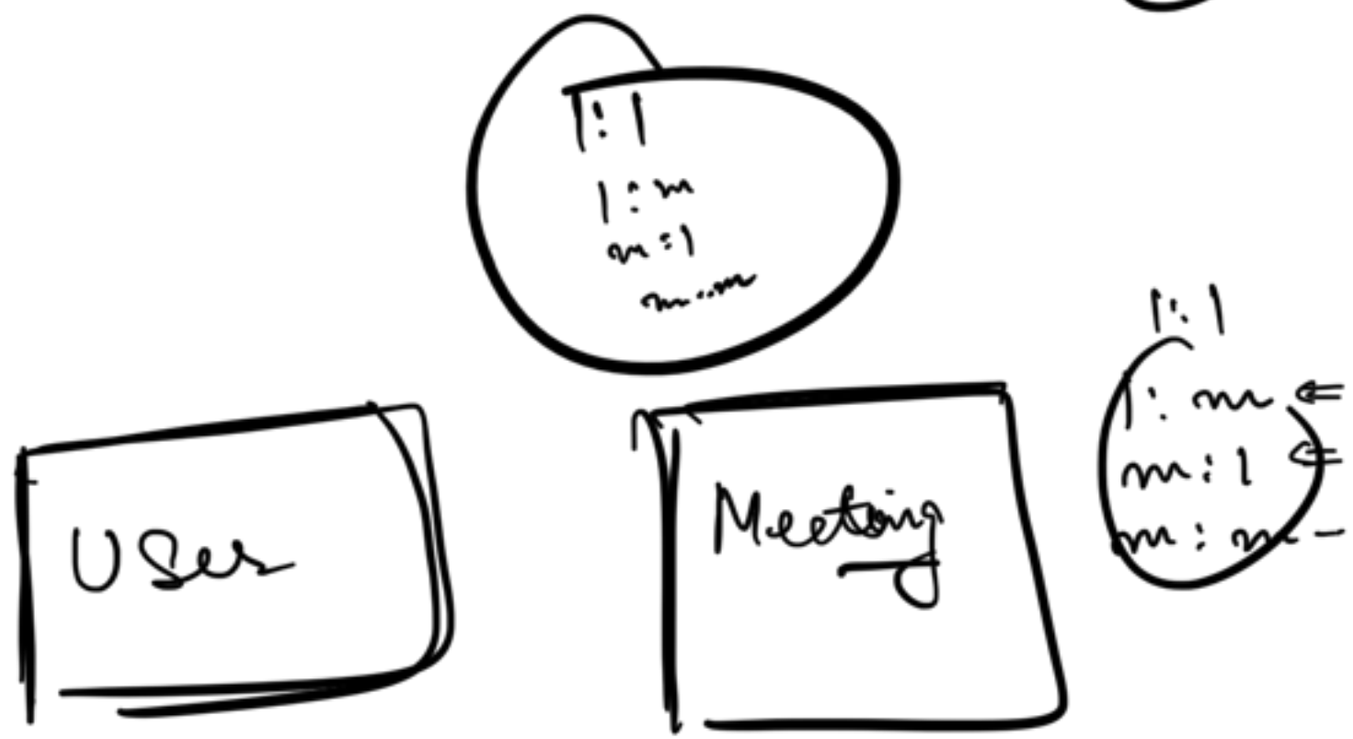
→ Mapping table

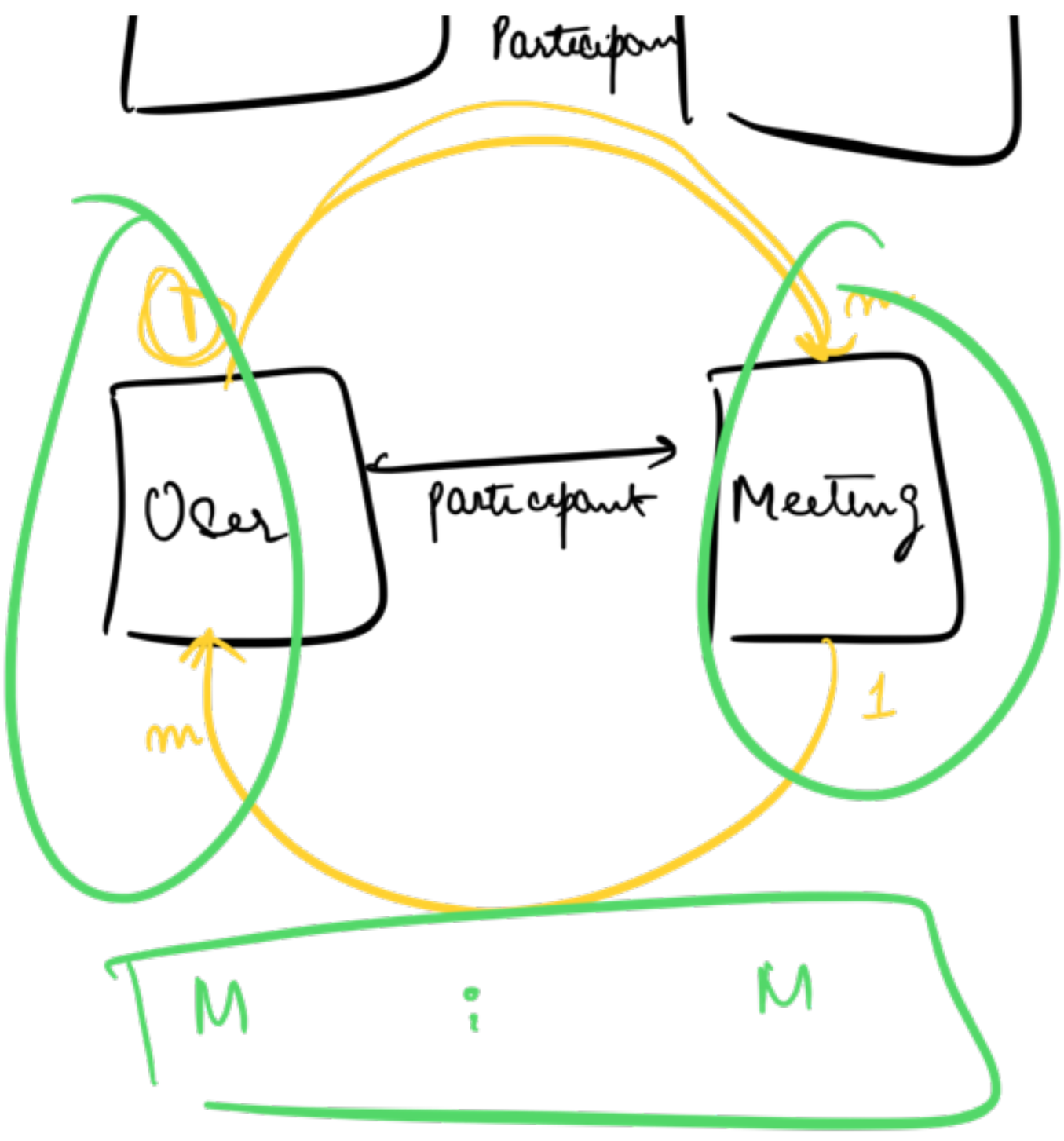
Students

hotdus

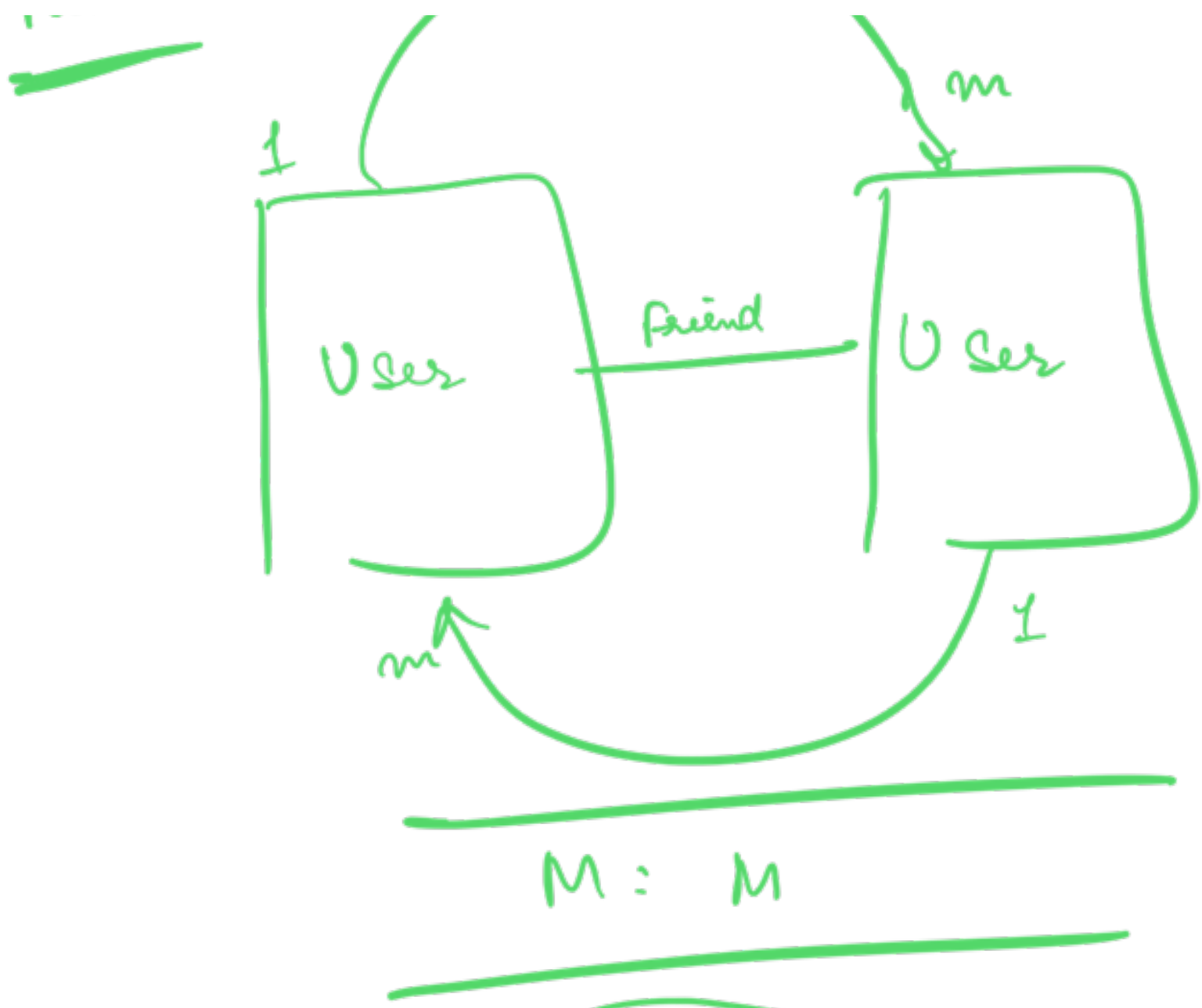
Student_Rated_Mapping

Google

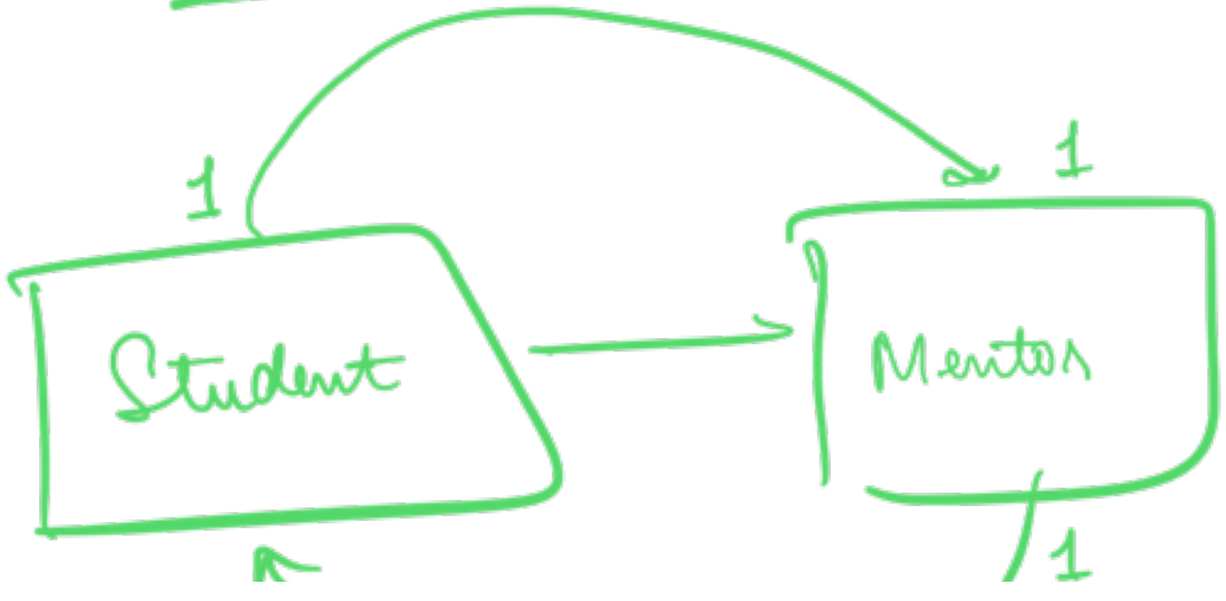




Facebook



M : M



M

M : 1

1

1

Student

Batch

M

1

M : 1

1

m

Aadhaar No

Bank Ac

1 — 1

1 : 1

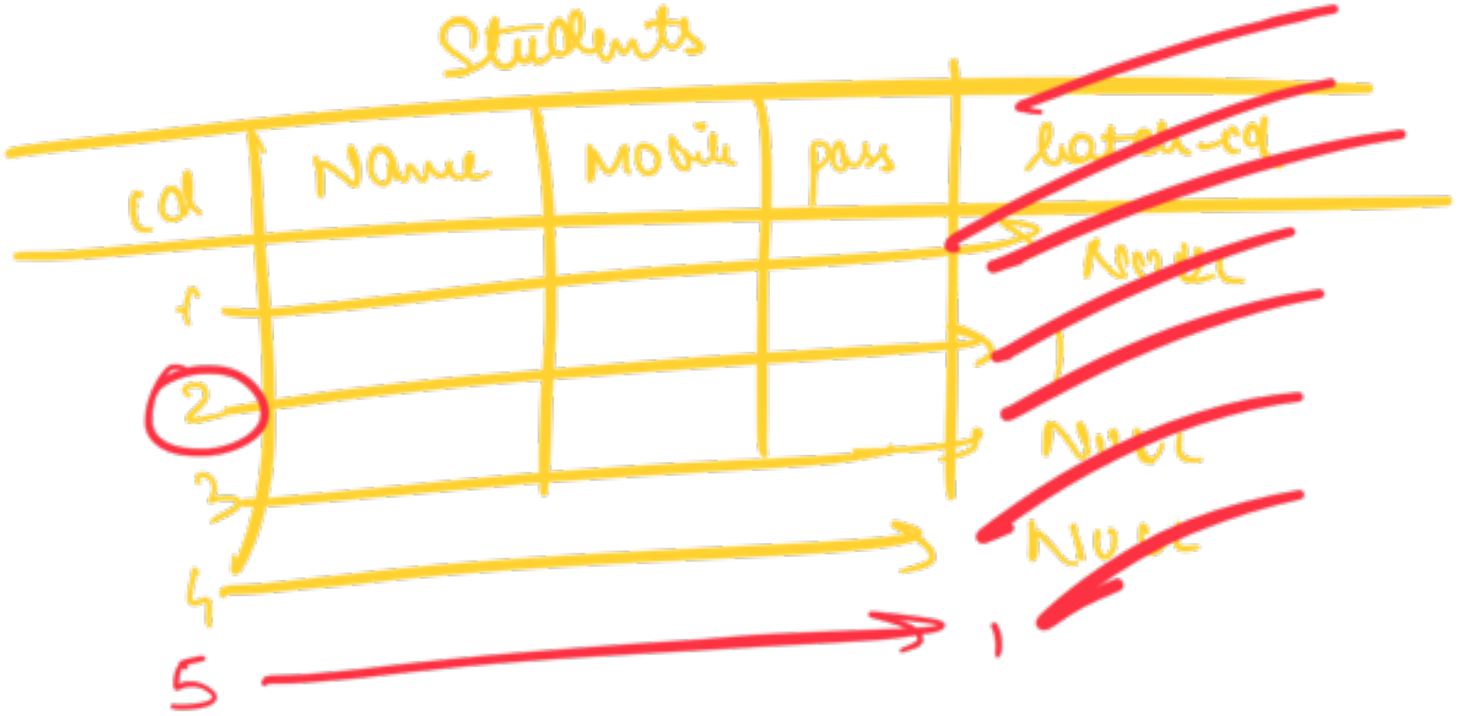
Type	How To Rep
M:M	Separate Mapping Table
M:1 OR 1:M	Separate Mapping Table
M:1 OR 1:M	Put an ID of 1 side in the table of M side
1:1	Put ID of any side on any side

almost once NULL

Student — 1 — 1 — Batch


M
1
M : 1


1 : 1
Student
Batch


Students
Id
Name
Mobile
pass
batch-id
1
2
3
4
5
1
Student_batch_mapping

2	1
5	1

user_user - friend - mapping

user - id	user_id	comm - date

User_batch_mapping

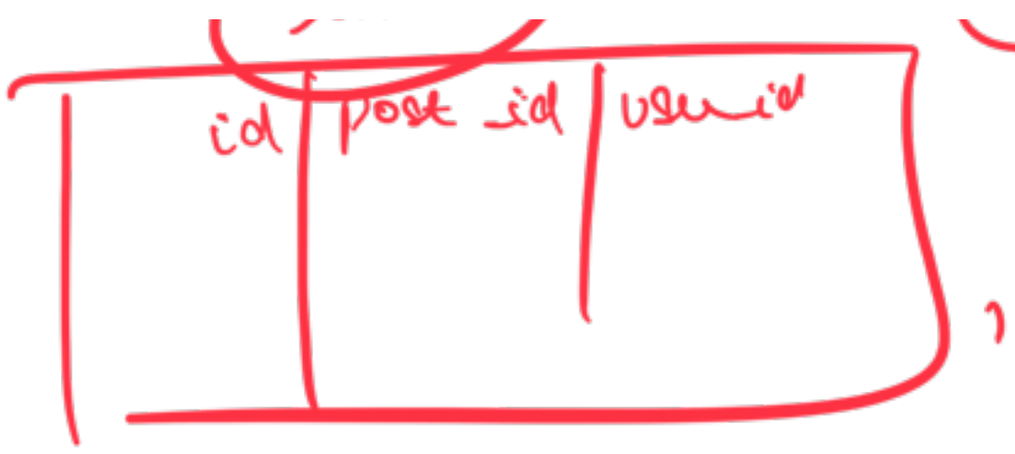
St_id	b_id	p&p	attends

posts

users

likes

likes



PSP ⟷ Batch ⟷ Student

Every Student can be a part of Multiple Batch

Every Batch will have multiple contest

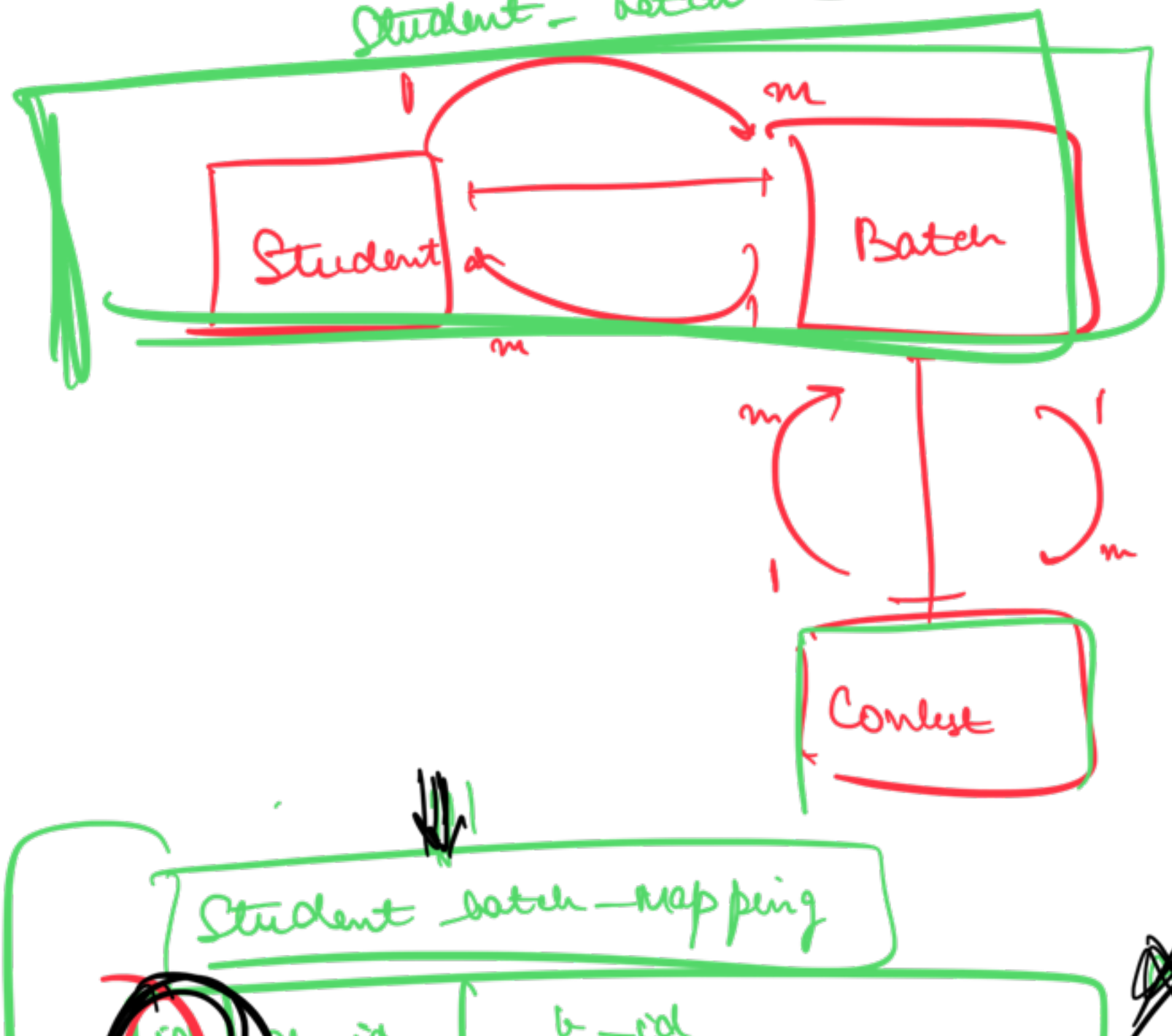
Store score of every student in every contest in every batch.

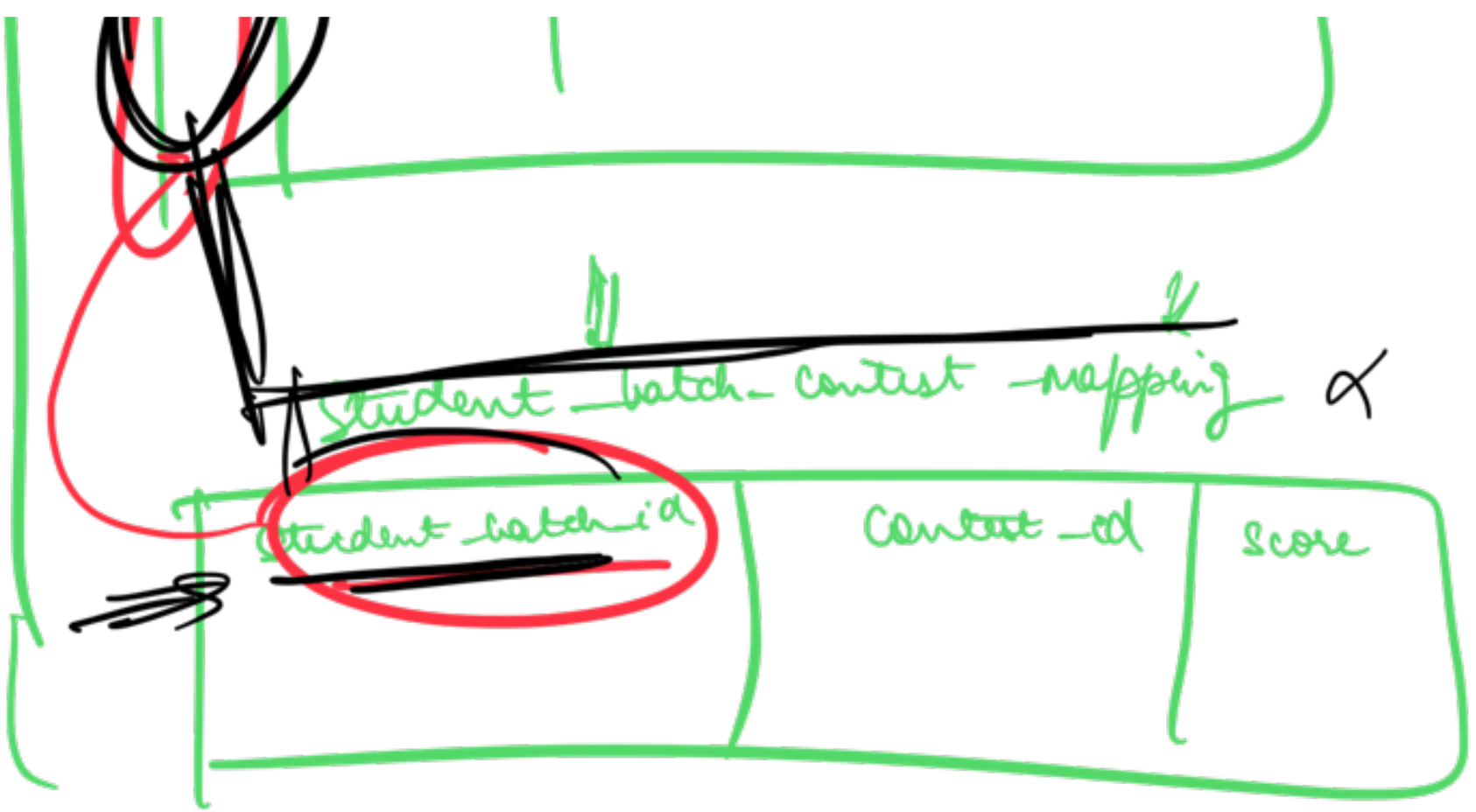
Stud → Batch → Contest

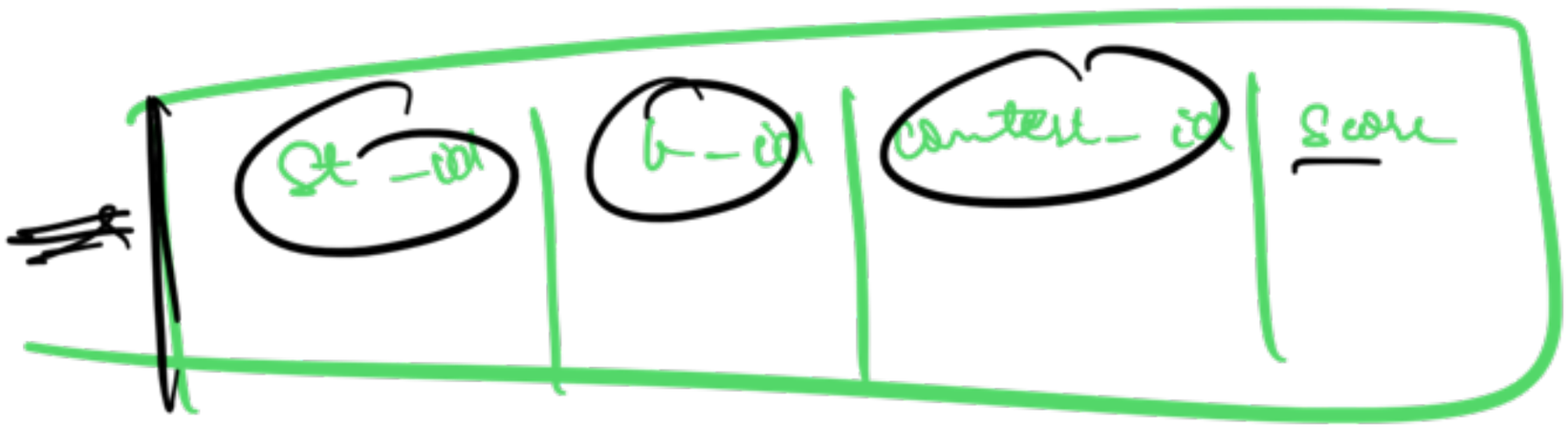
1 → 1 → 15 → 40

1 → 3 → 13 → 100

Student_batch







Outcomes

(1) How to find cardinality

② For every Cardinality how to put arrow

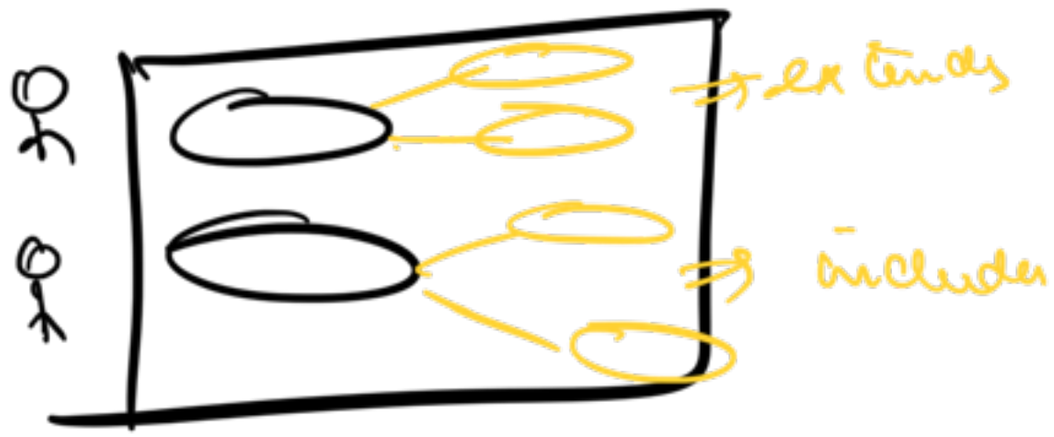
③ Types of assoc^{n}

① Aggregate == collect

② Composition == create

④ Class Diagram

⑤ Use Case Diagram



Assign

Draw in draw.io

→ Draw use case diagram and class diagram of PayTM.