

# DBMS-2



① Complete Relational Model

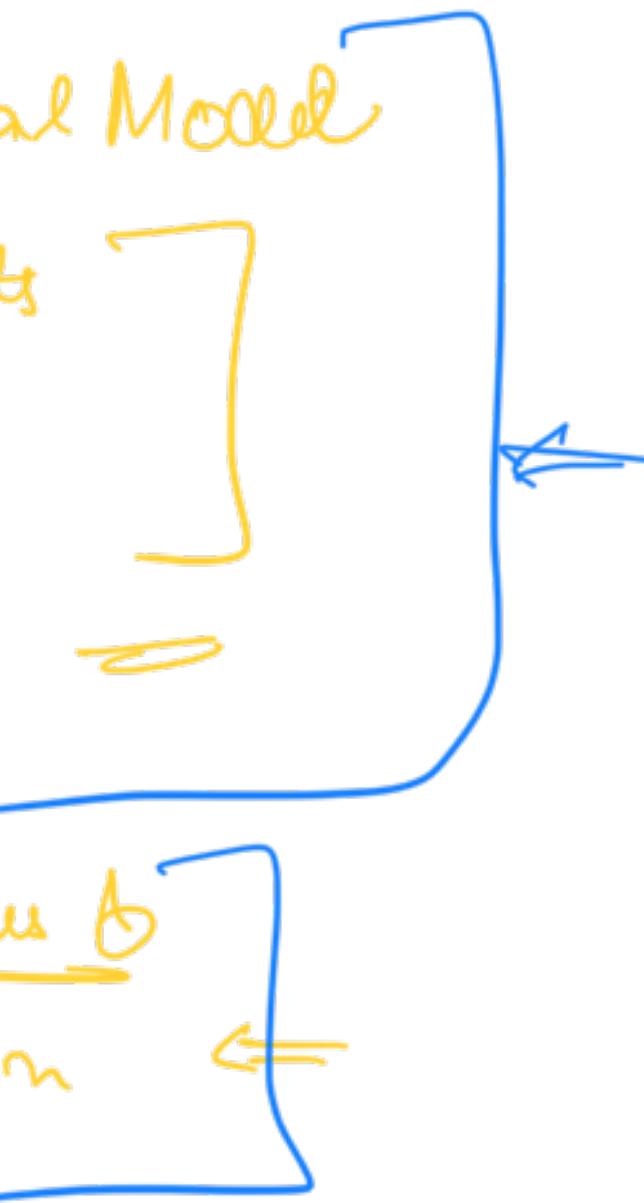
→ Constraints

→ Schema

② ER Diagrams

③ Functional Dependencies

Normalization



Next Class : Transaction & ACID Properties

Next to Next: SQL

~~DBs~~ Relational Database Constraints

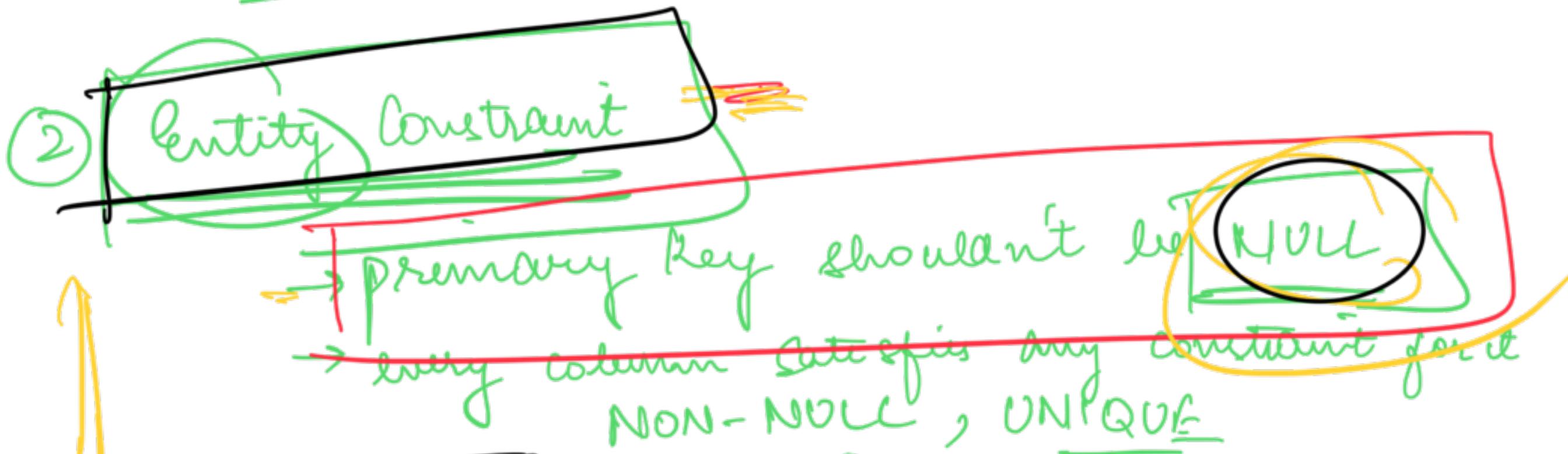
↳ DB Checks

① Domain Constraint

Data Type (Type of data for a particular column)  
→ In & connect data type

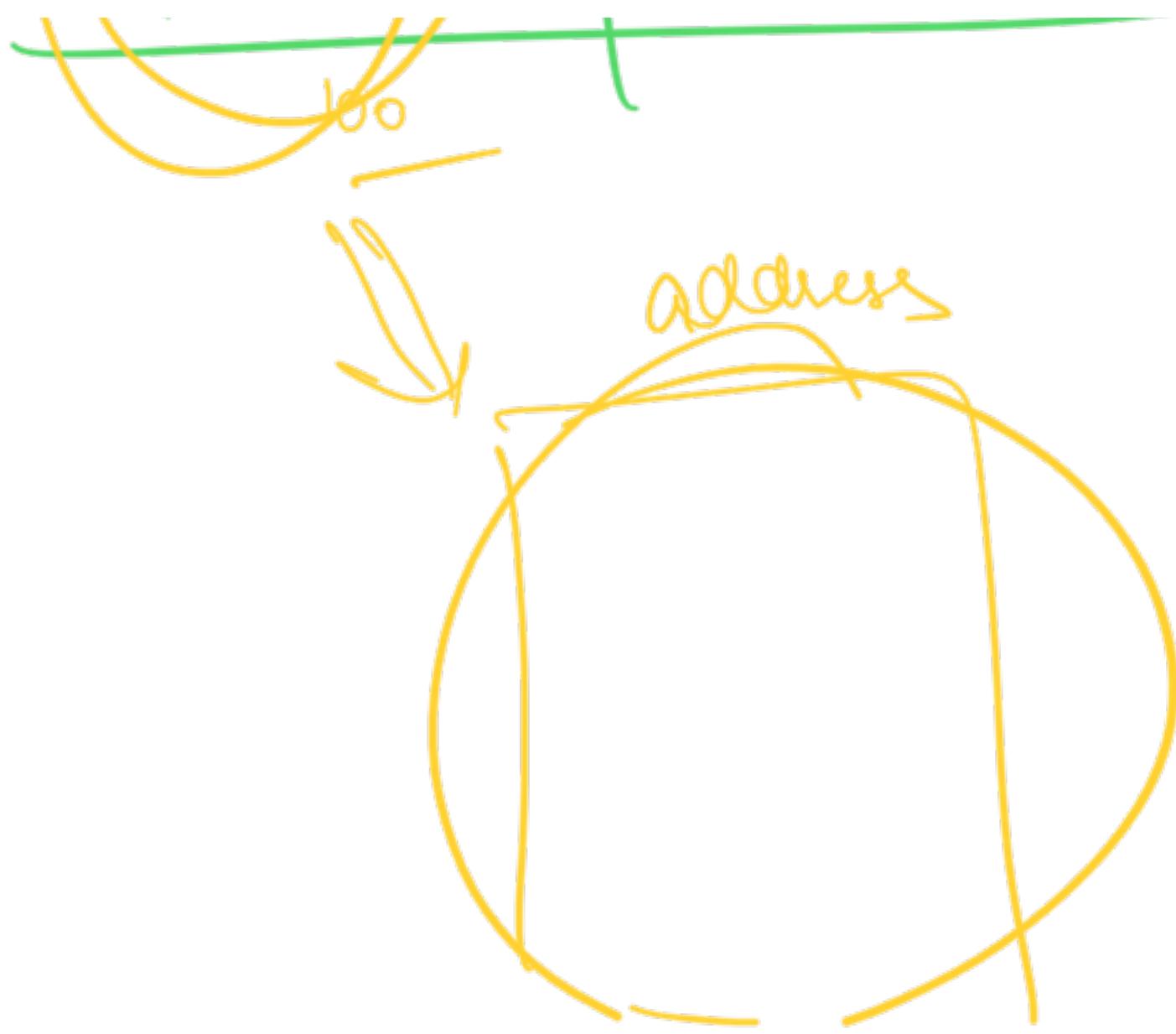
⇒ Value must be of type

UI



②.1 Key Constraint → Keys are unique values

id	address
1	
2	
3	
4	



Students

id	Name	address	batch_id
1	John Doe	123 Main St	1

batches

id	Name
1	Aug 21 Intake
2	Sept 11 Intake

①

3

cp II Semester

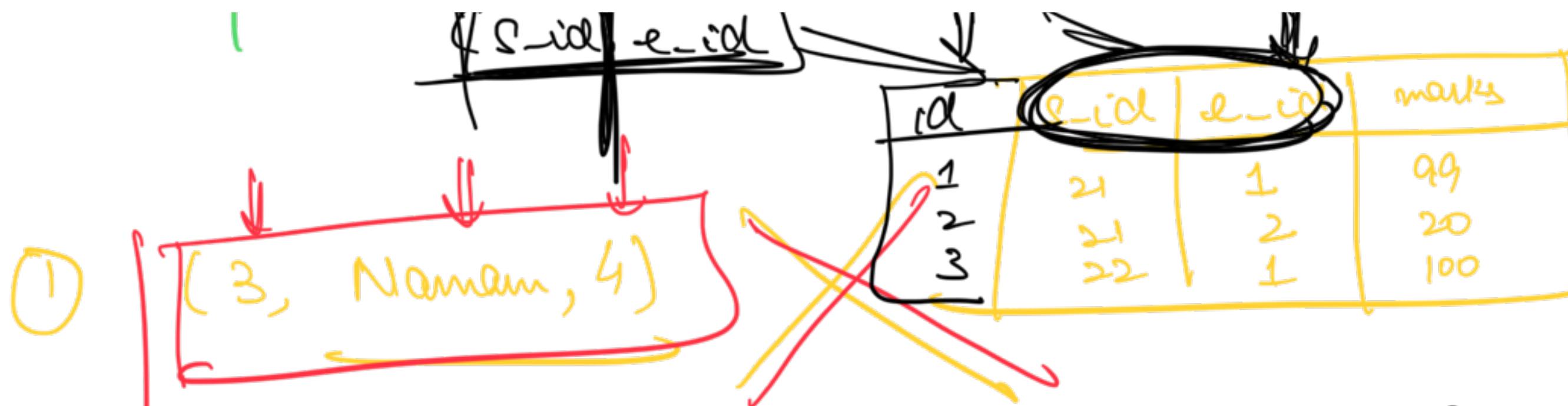
### ③ Referential Integrity

Students

id	name	batch_id
1	Ashish	2

batches

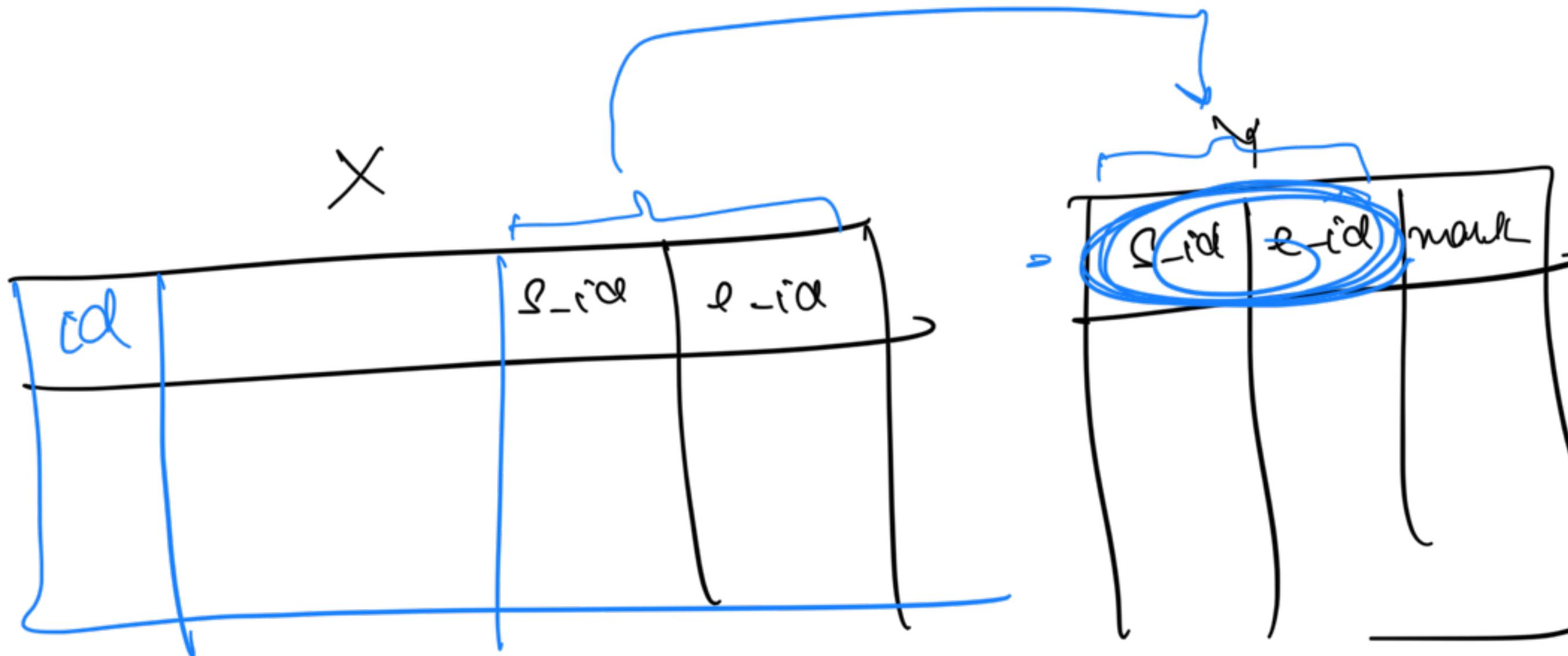
id	Name
1	Apr 22 Intake
2	May 22 Intake
3	May 22 Adv



Can't insert a row if corresponding fk doesn't exist in other table

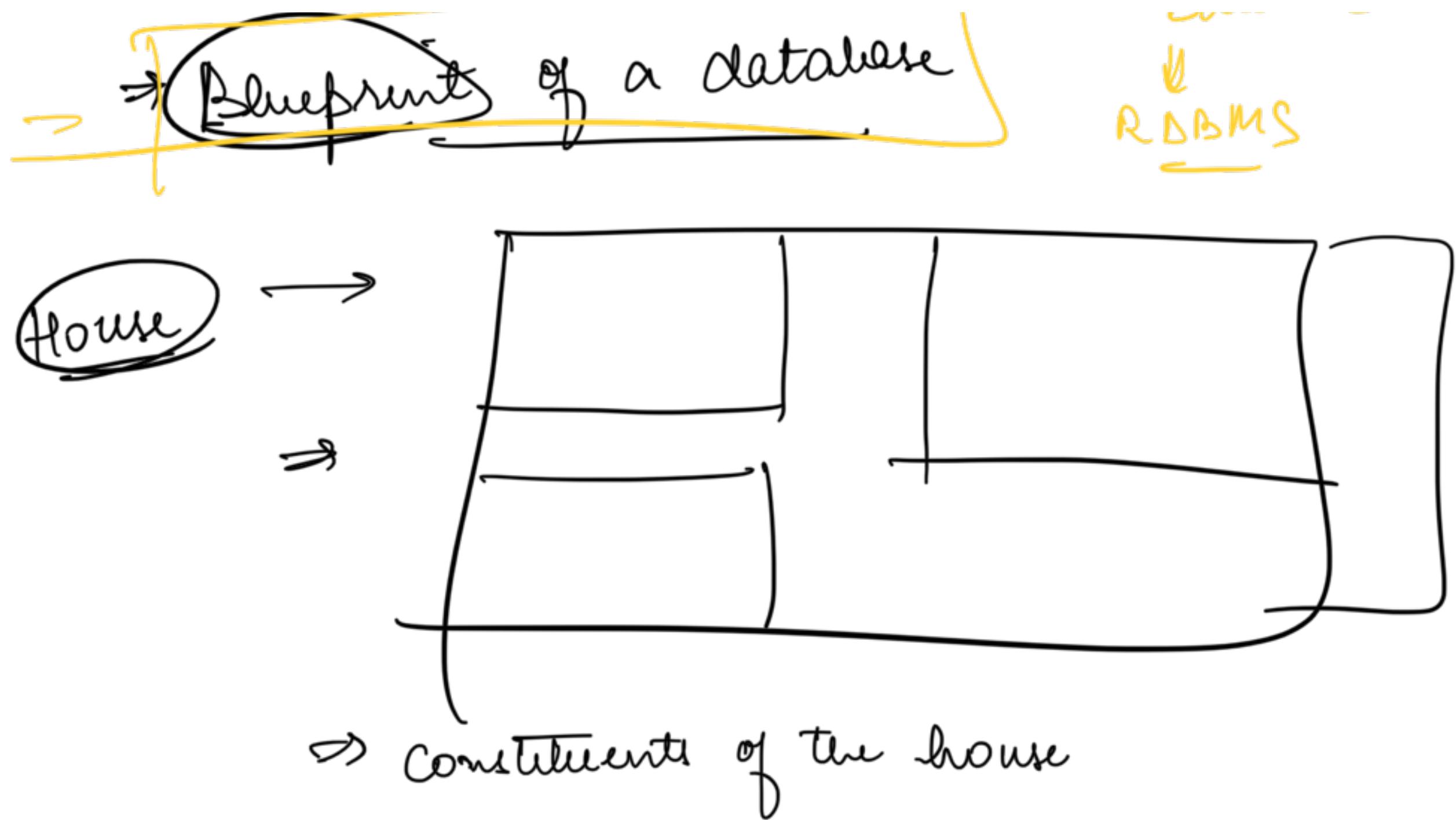
② Can't delete a row in the table being referred if there are still referring instances

Schema



Schema

→ Constituents of  
a Database



➔ Schema is going to contain all the tables that will be there in the DB

→ It also contains columns in every table

→ pk, fk of every table

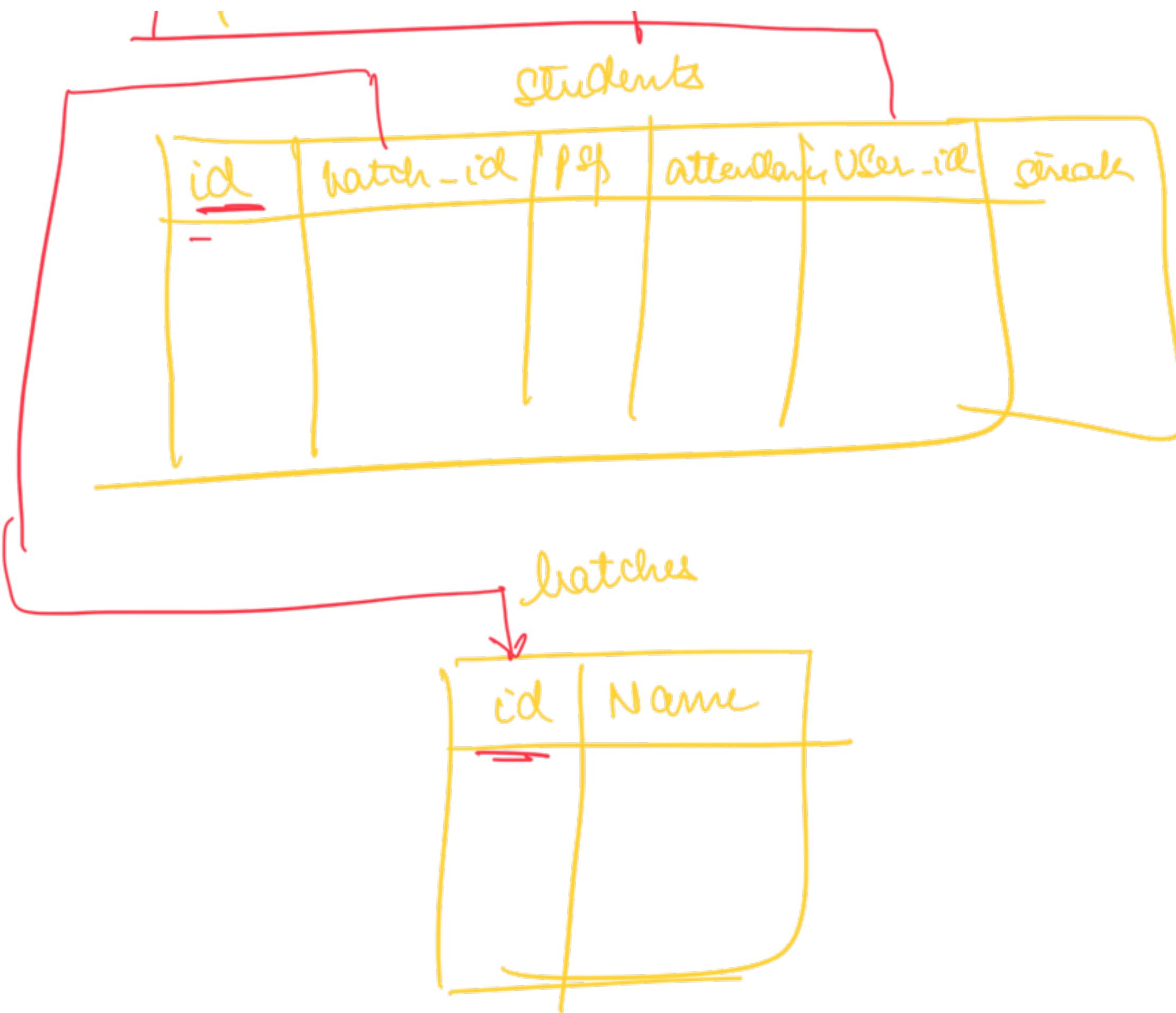
Scalene's DB

The diagram shows a table structure with the following columns:

- id (Primary Key)
- name
- ph No
- email
- grad year
- univ
- yoe

Red annotations are present:

- A red circle highlights the "id" column.
- Red diagonal lines cross out the entire row for the "name" column.
- A red circle highlights the "email" column.
- A red arrow points from the "Users" label above the table to the "id" column.
- A red arrow points from the "Users" label to the "yoe" column.



Step 1  $\Rightarrow$  Create all tables

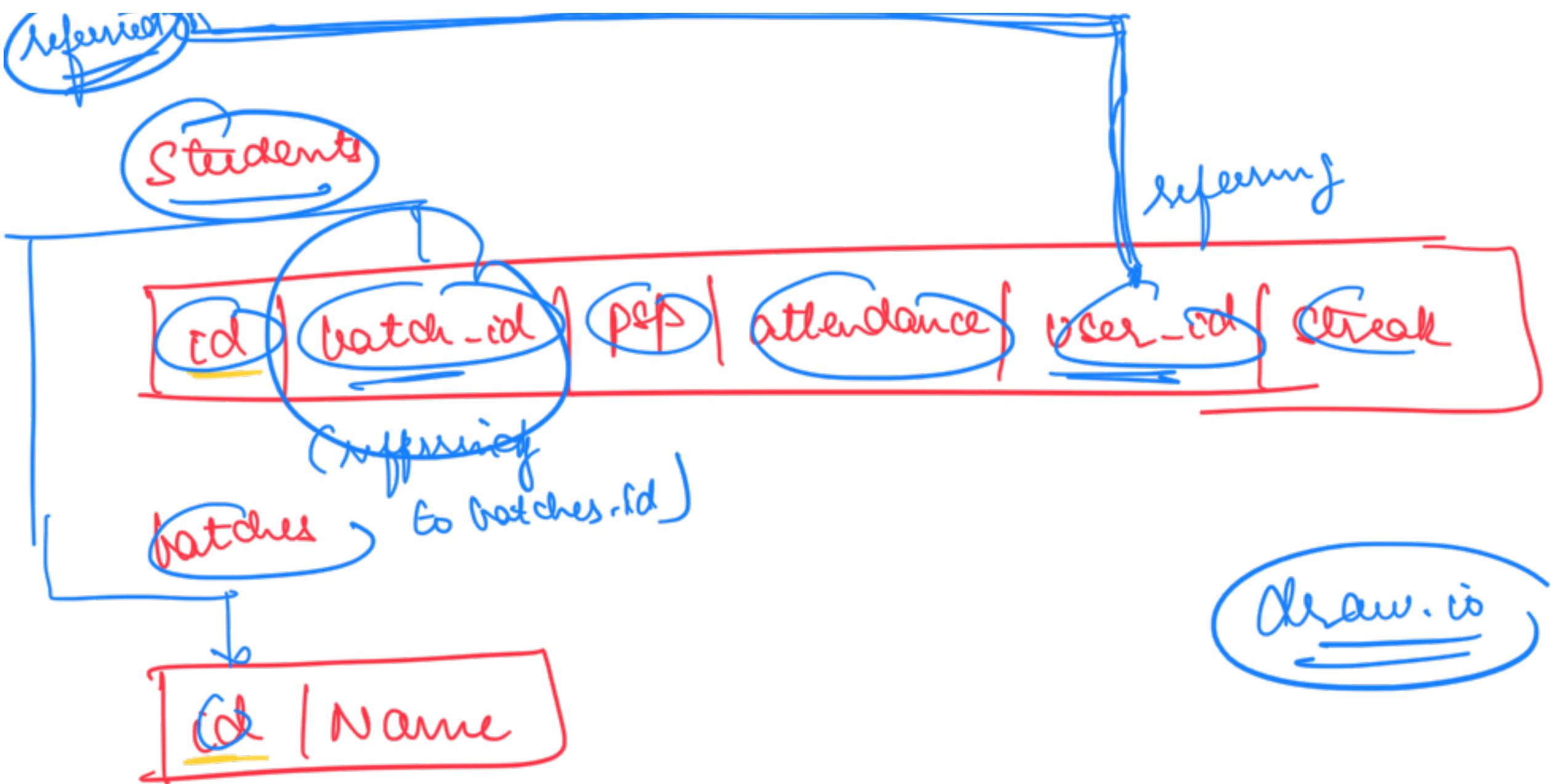
Step 2  $\Rightarrow$  Create fk

~~sq CD~~

Users

UNIQUE

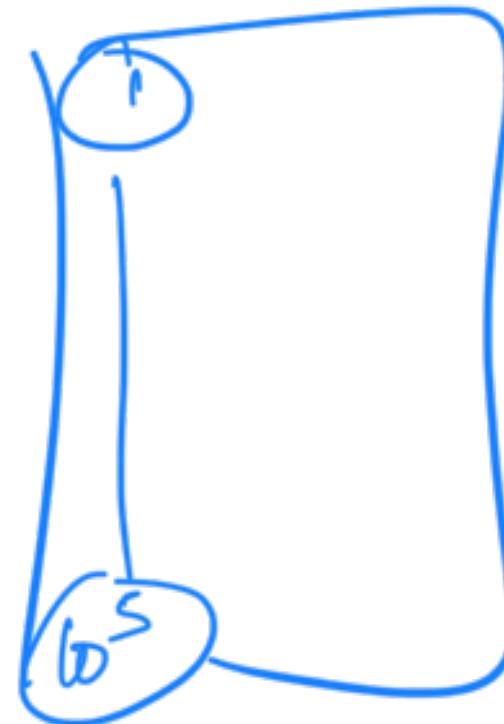
<u>id</u>	<u>Name</u>	<u>perNo</u>	<u>Email</u>	<u>grad year</u>	<u>uni</u>	<u>voe</u>
→						



Users

1:1

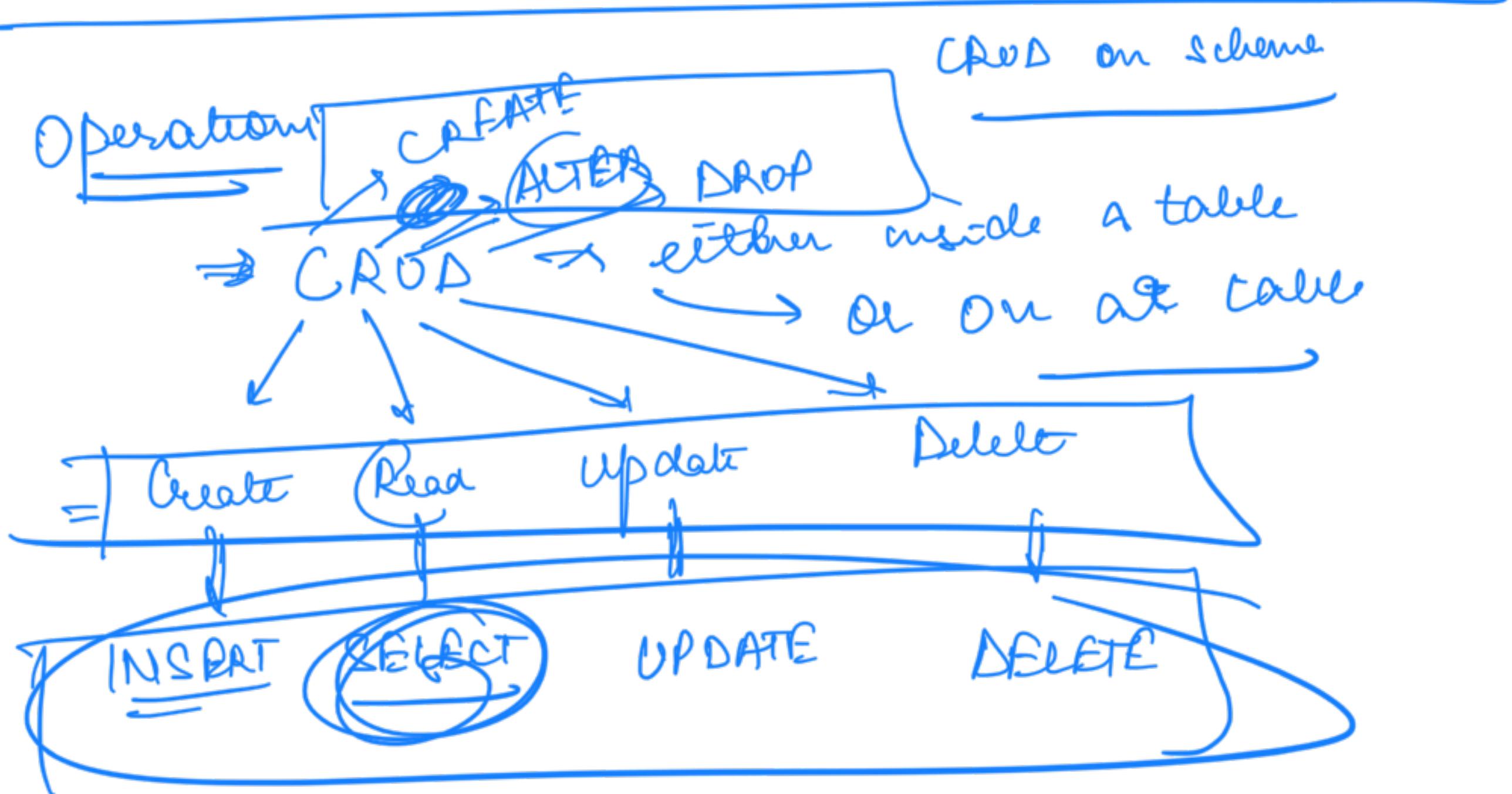
Courses

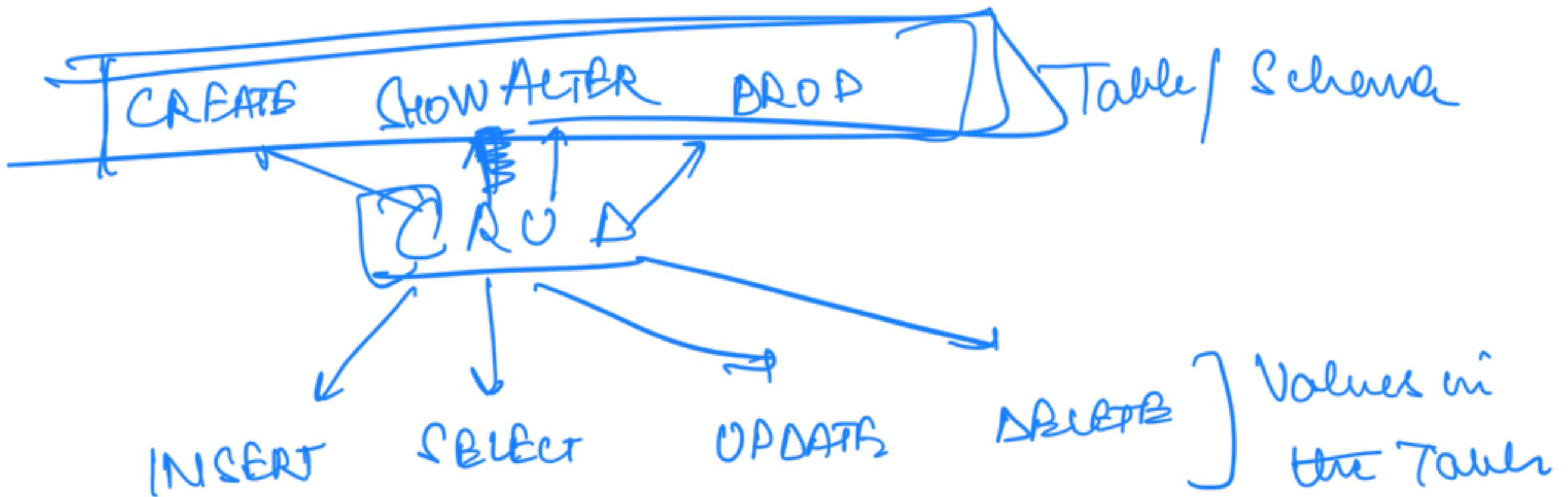


FK: Column(s) in a table which is referring  
to primary key of another table

PK: Column(s) in a table using which you  
can identify any row

(UNIQUE)





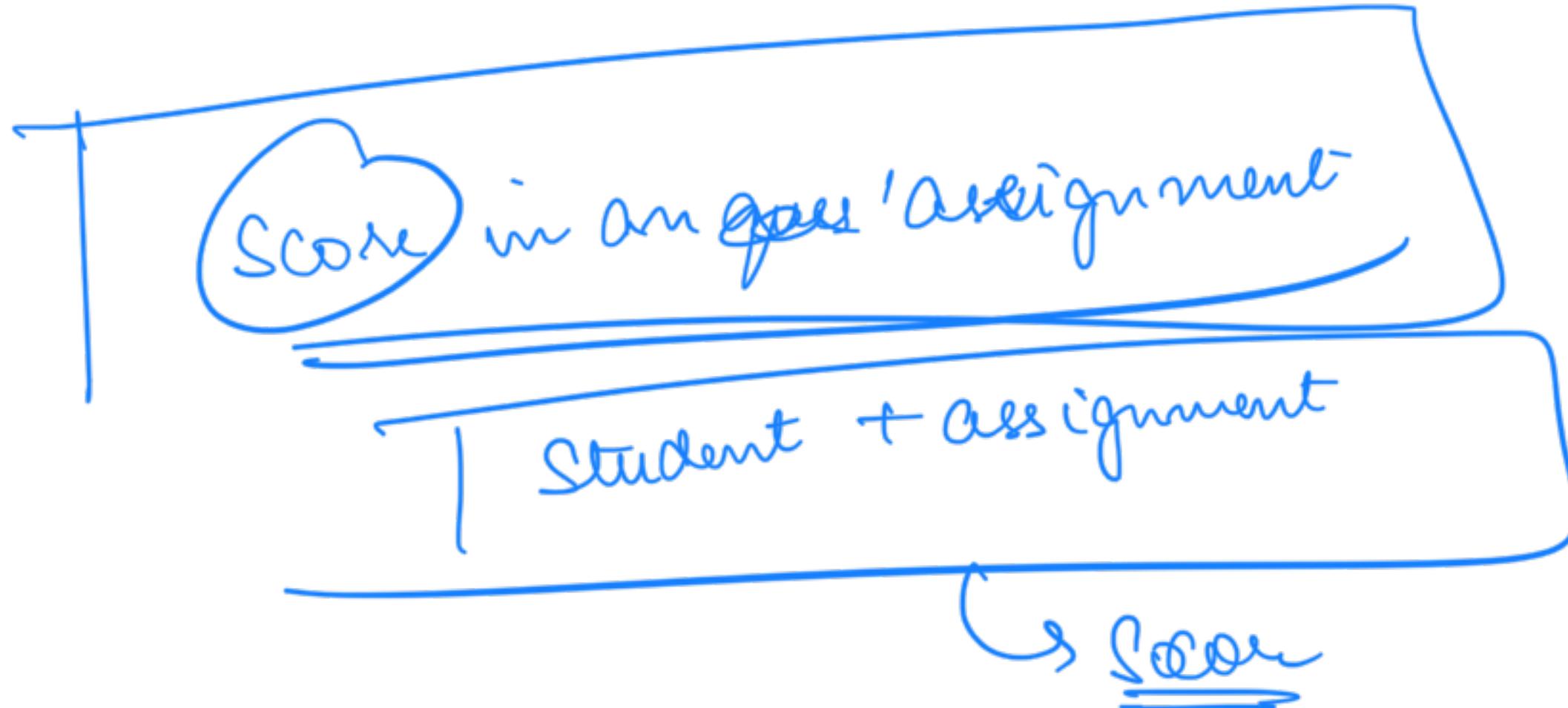
# ER Model / Diagram

## ~~Entities~~ Relationship Model

¶

E: Entities: Anything for which I have to store info

R: Relationship: Association b/w entities



## Case Study : Design ER Diagram of Scoler

① There are several batches at Scoler.

Every batch will have ID, Name and  
current ~~in true slot~~

② Each batch will have multiple classes

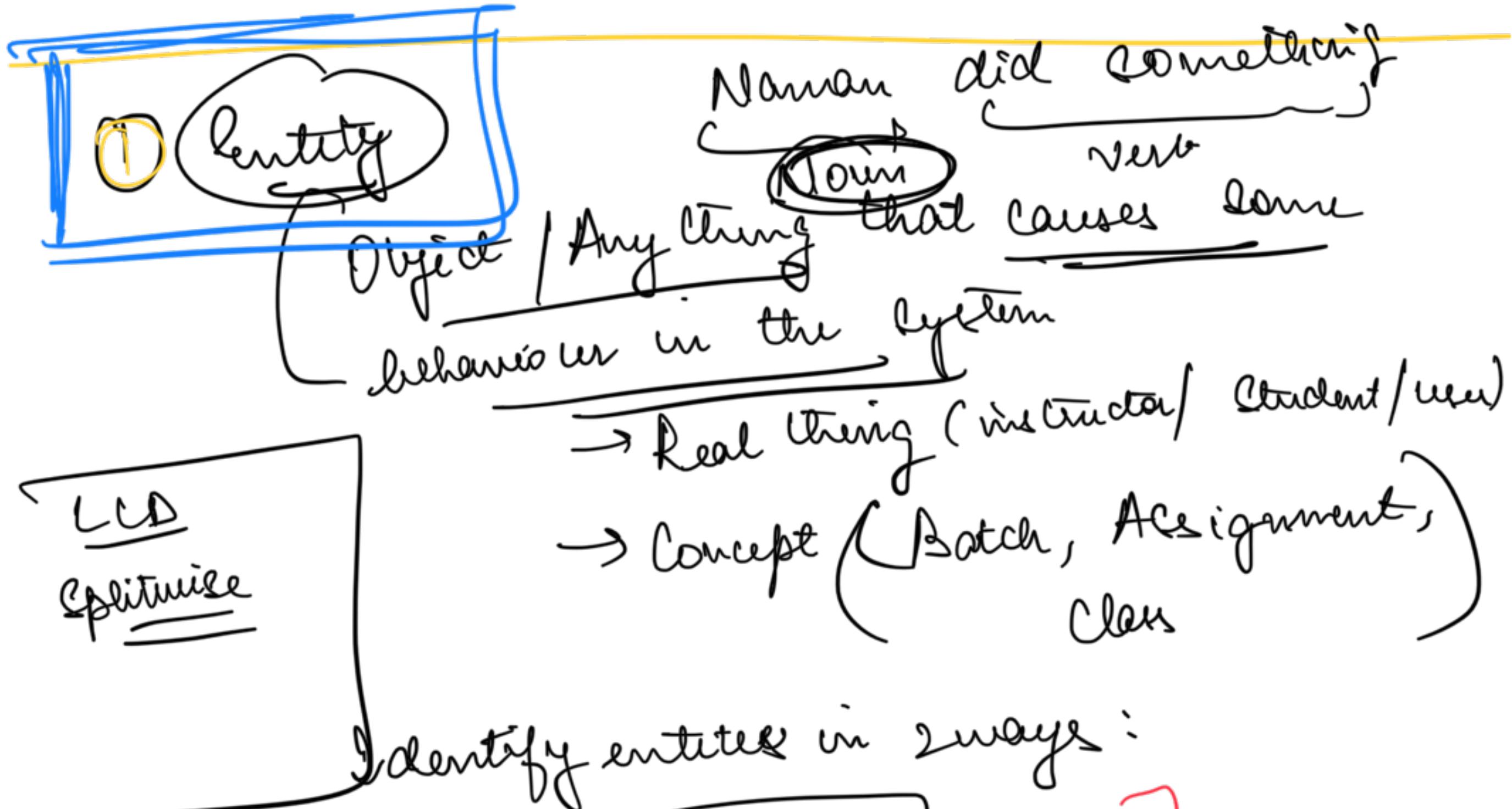
Every class I need to close ID, Name,  
instructor

③ One class can be attended by multiple

→ batches

- ④ Every student has Name, ID, grad year,  
→ uni, email, phone Number
- ⑤ Every student has a student buddy
- 
- ⑥ A student can move from batch to  
another if when they pause a course.
- for every student, for every batch they  
were in, we need to store their batch  
entry date

① Every Student has a mentor Mento<sup>z</sup> has a name and VDE.





→ Every entity has attributes

→ Represented using Rectangle with name  
wide rectangle

ER ⇒ Relation

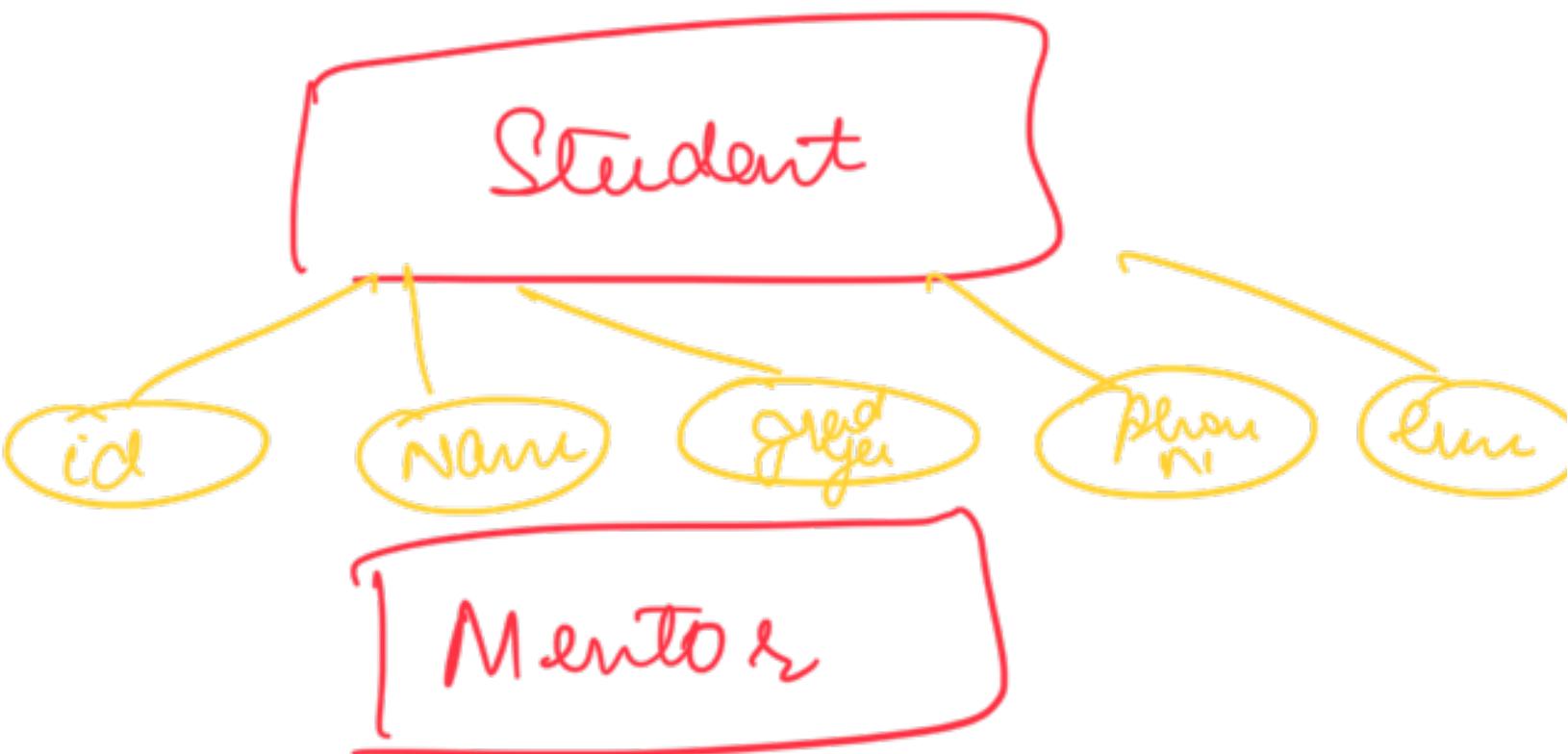
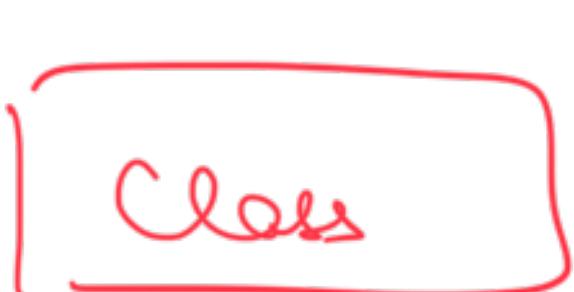
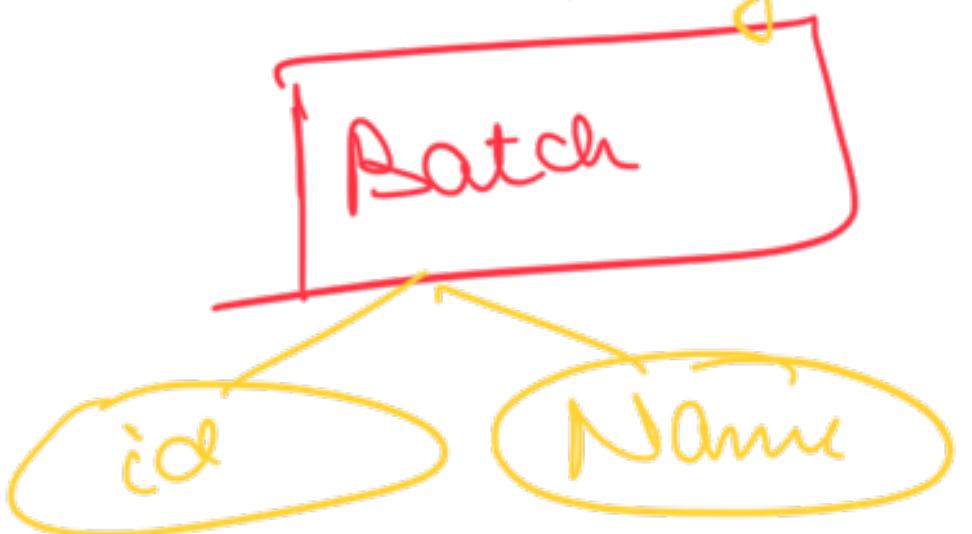
Entity ⇒ Table

attribute ⇒ Column.

... → A relation is represented by

## ② Attributes of entities

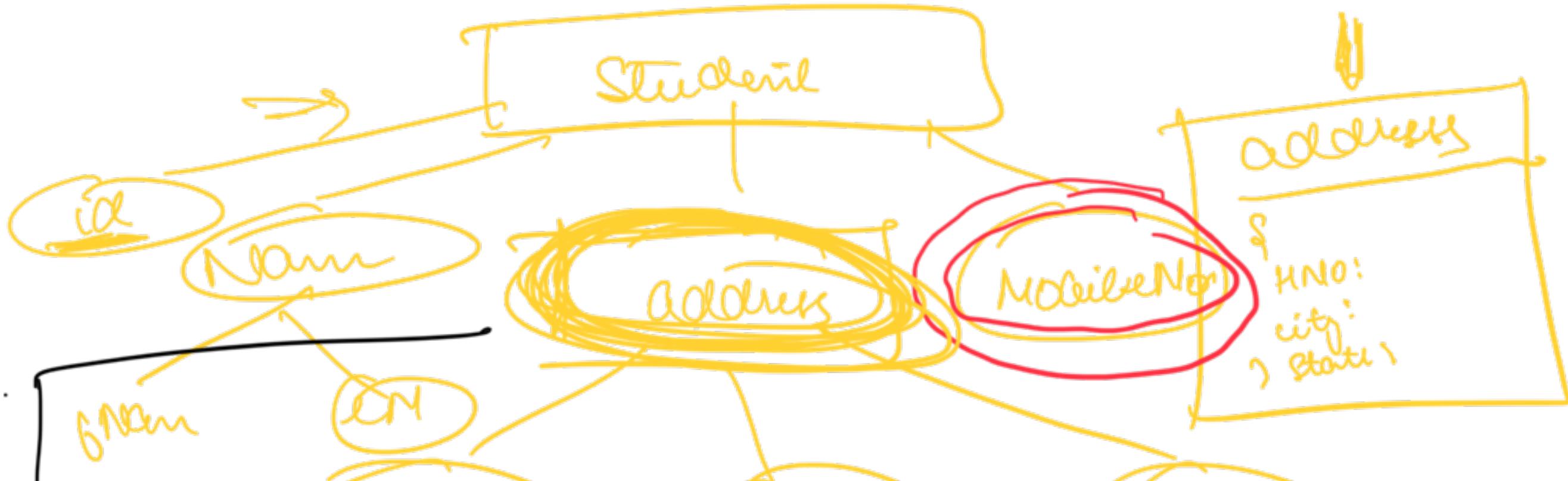
- info about entity
- every attr might have a range of values



## Multiple types of attr in ER Model

① Simple: Can't be divided further  
→ Atomic Value

② Composite: Which have further parts  
(like address)



houseNo

City

State

↓

addresses

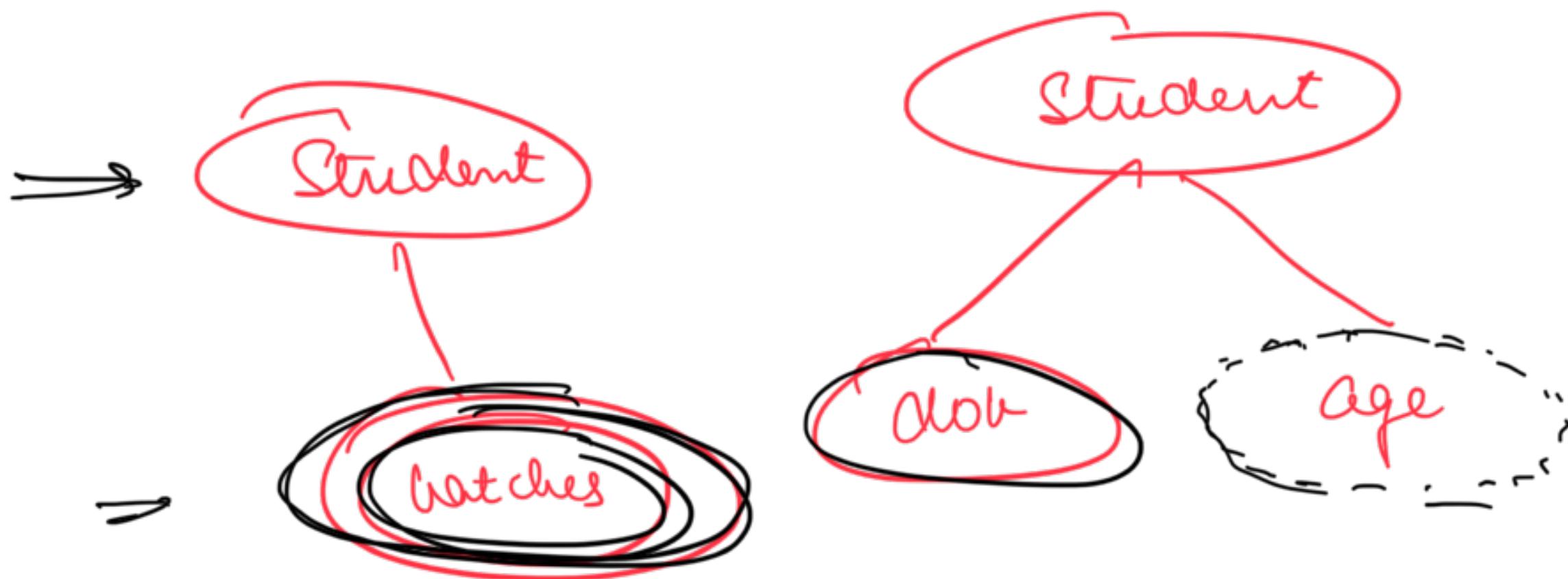
<u>id</u>	<u>hNo</u>	<u>city</u>	<u>state</u>

③ Multivalued Attr

→ Lists, Arrays

→ P . . . more than one value

→ Can be www  
of same thing



#### ④ Derived attr:

- Attri whose value can be found from other attr I have

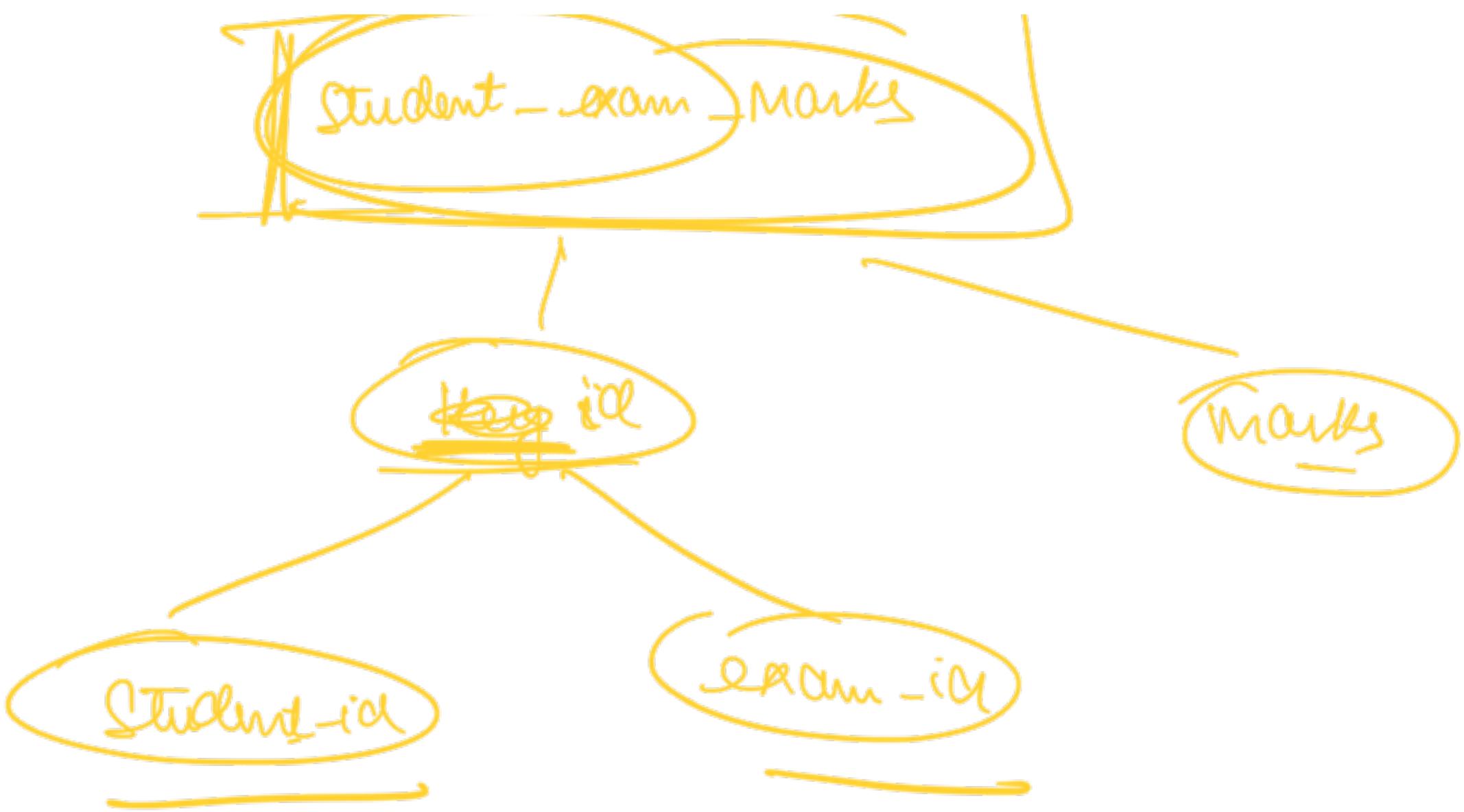
→ Rep using dotted oval

### ③ Keys

- Uniquely identifies a record
- Underline the ovals that are part of

The Key

- ↳ One attribute
- ↳ Multiple attributes



## ④ Relationships

→ Nothing but Conn connection b/w 2 entities.  
 ...  
 ...  
 ...

→ Up as a ~~unum~~



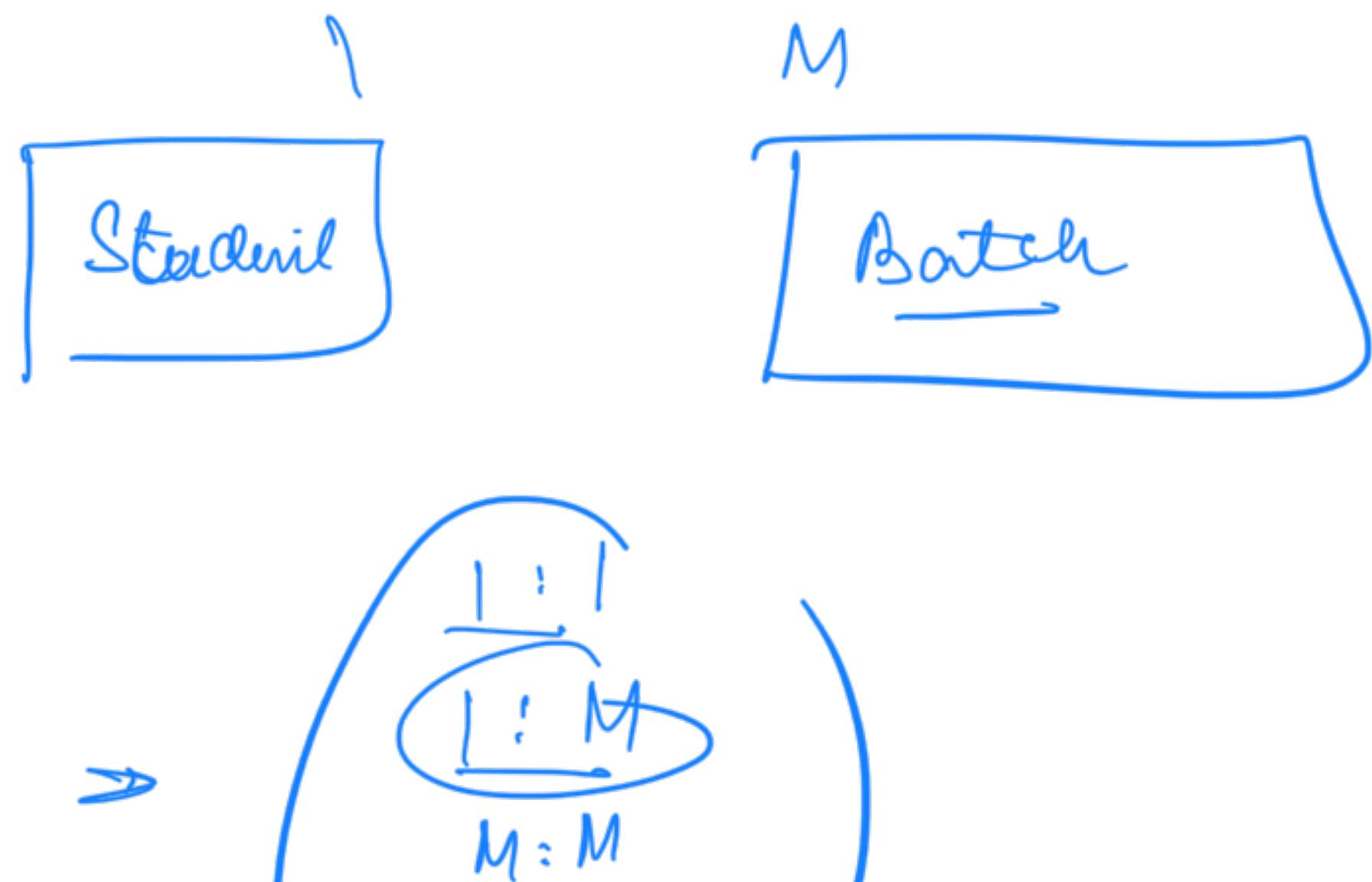
→ Cardinality of a rel<sup>n</sup>

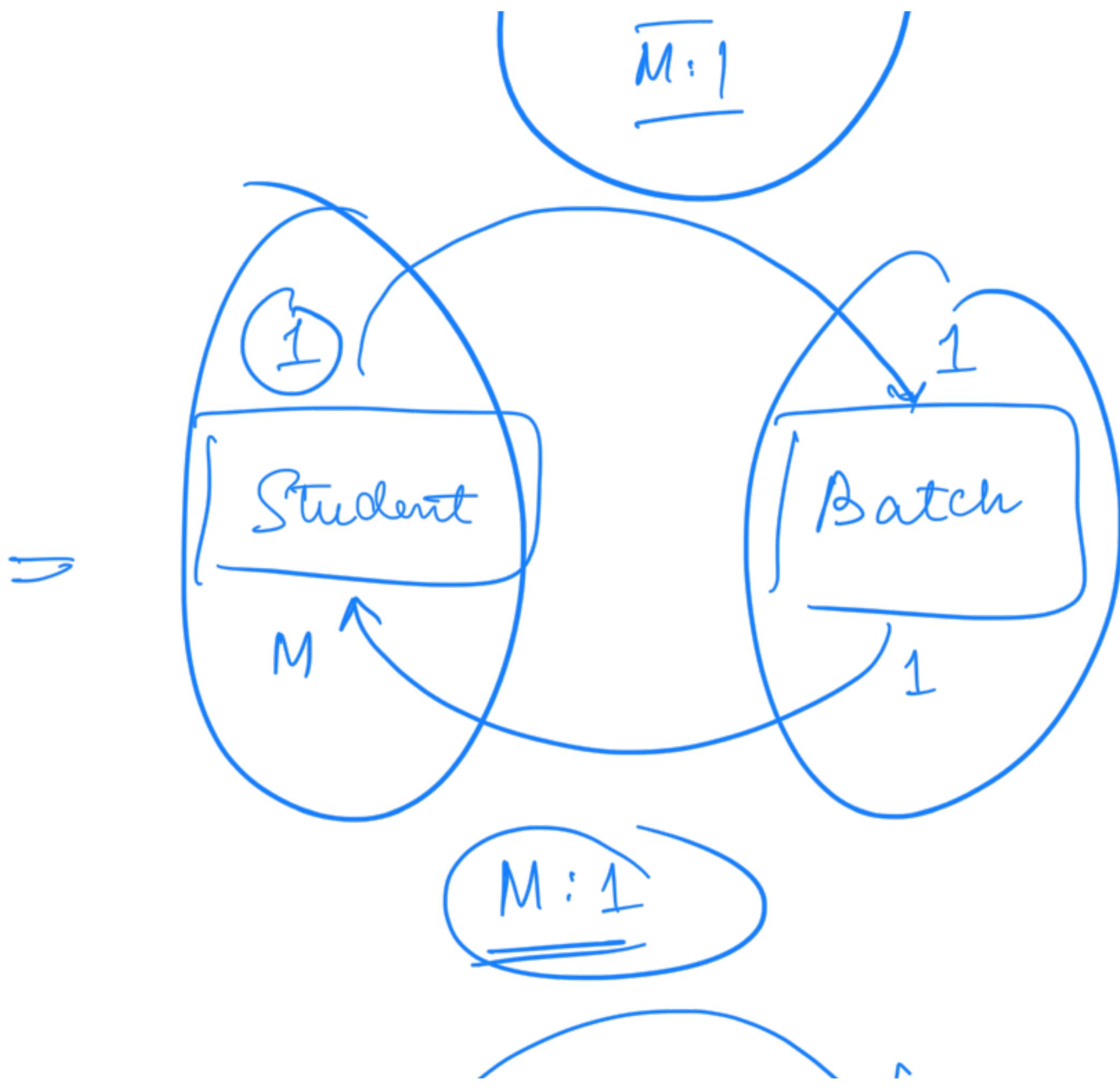
How many  
of  
left side

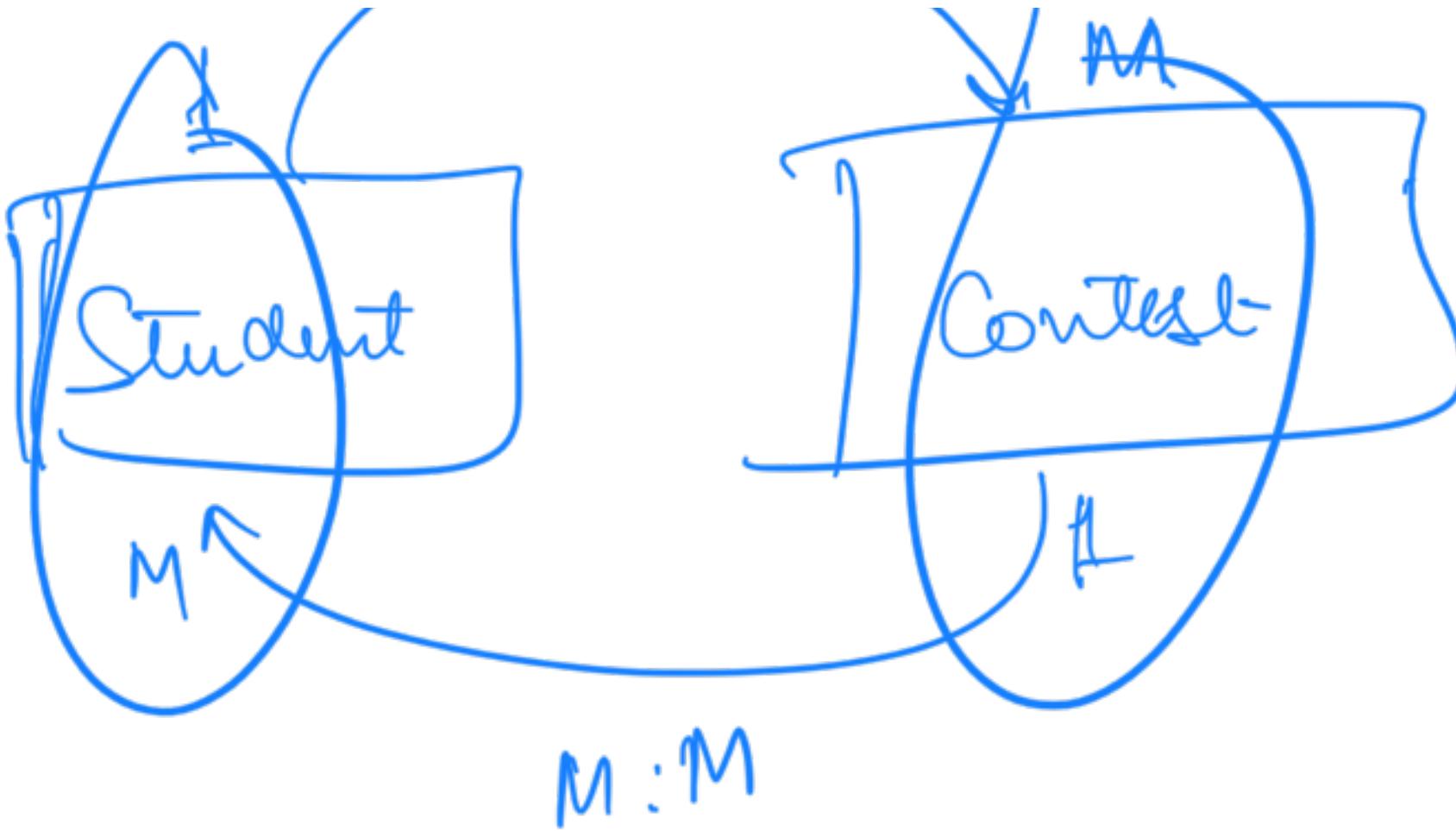
How Many  
of  
Right side

1 : 1  
1 : m →

$m : 1$   
 $m : m$

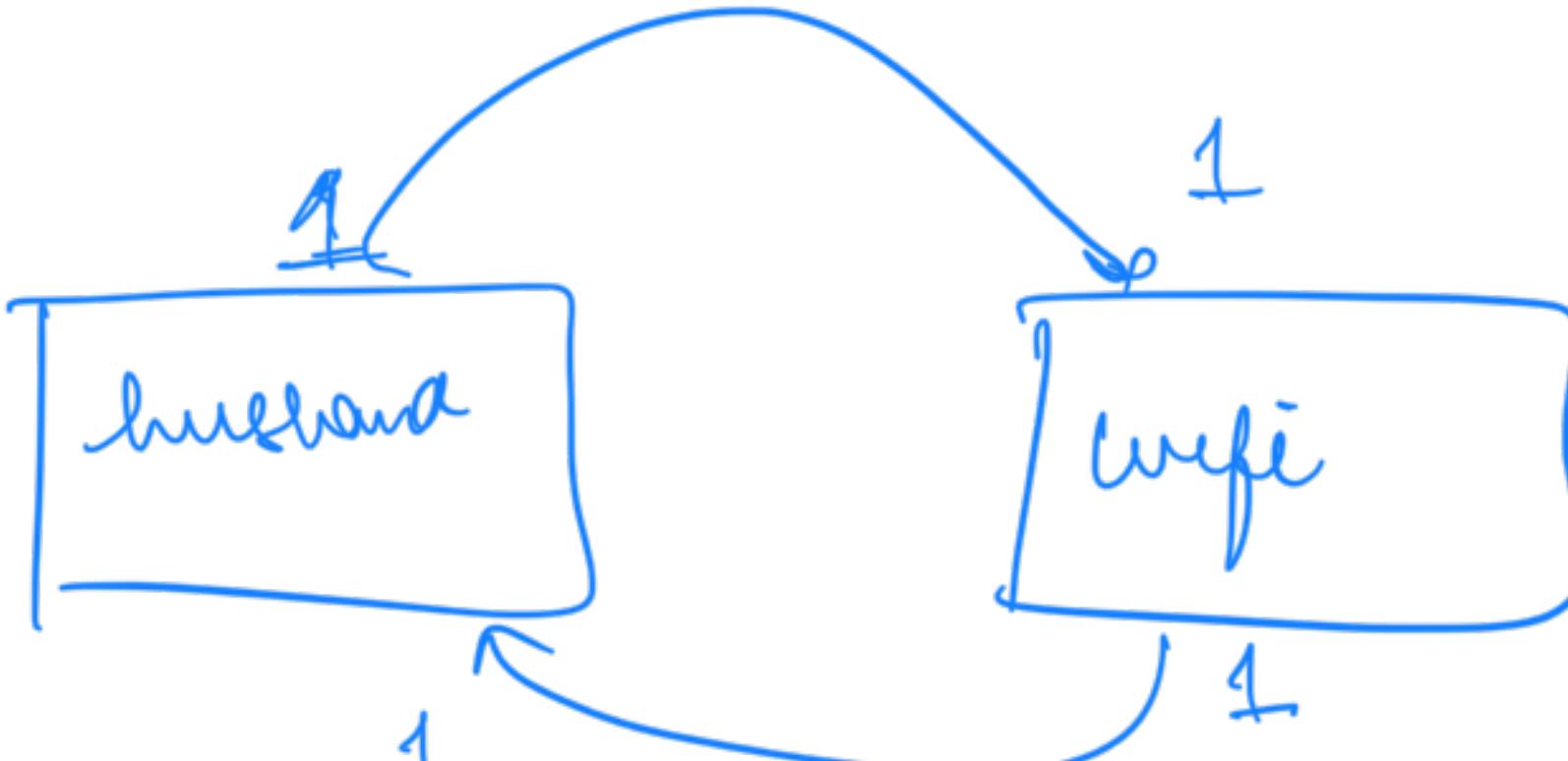




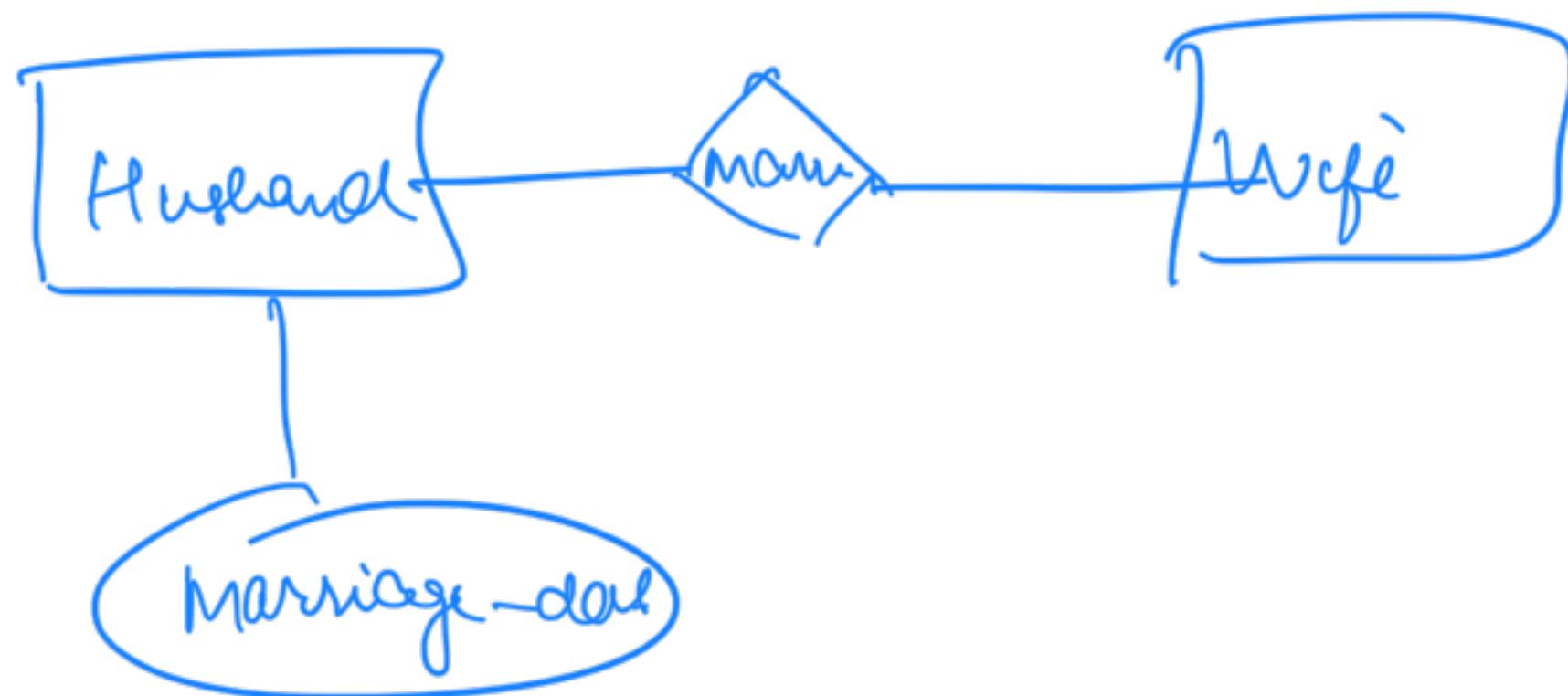


if the rel<sup>2</sup> is 1:1  $\Rightarrow$  put the attr on any of entity

you can put that attr on any of the involved entity



1:1 : fk can be on ANY ONE



1:M

or

M:1

→ Put the attr on M side

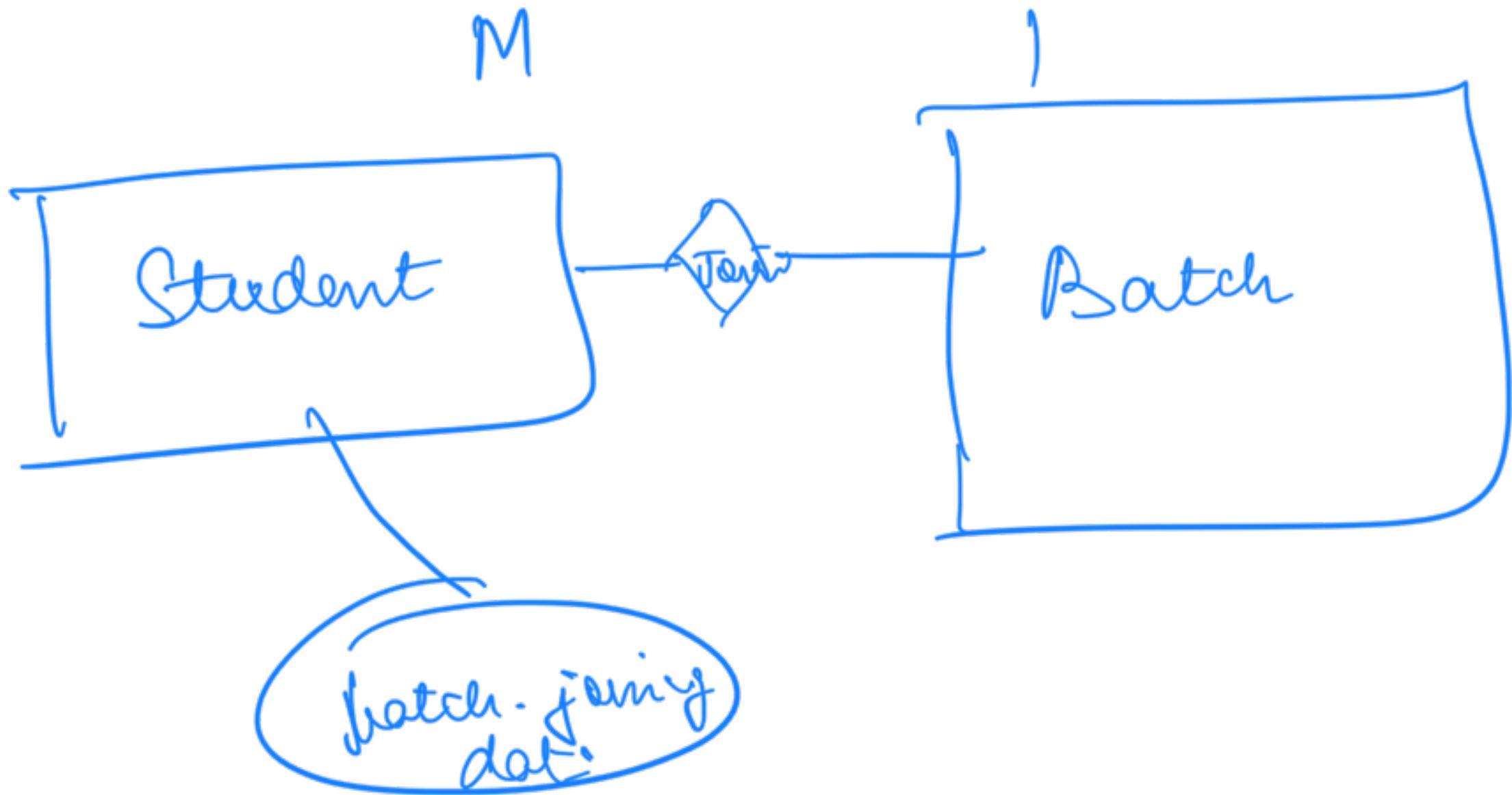
M:1

Students

id	name	email	batch_id

batcher

id	name	<del>student_ids</del>



M : M

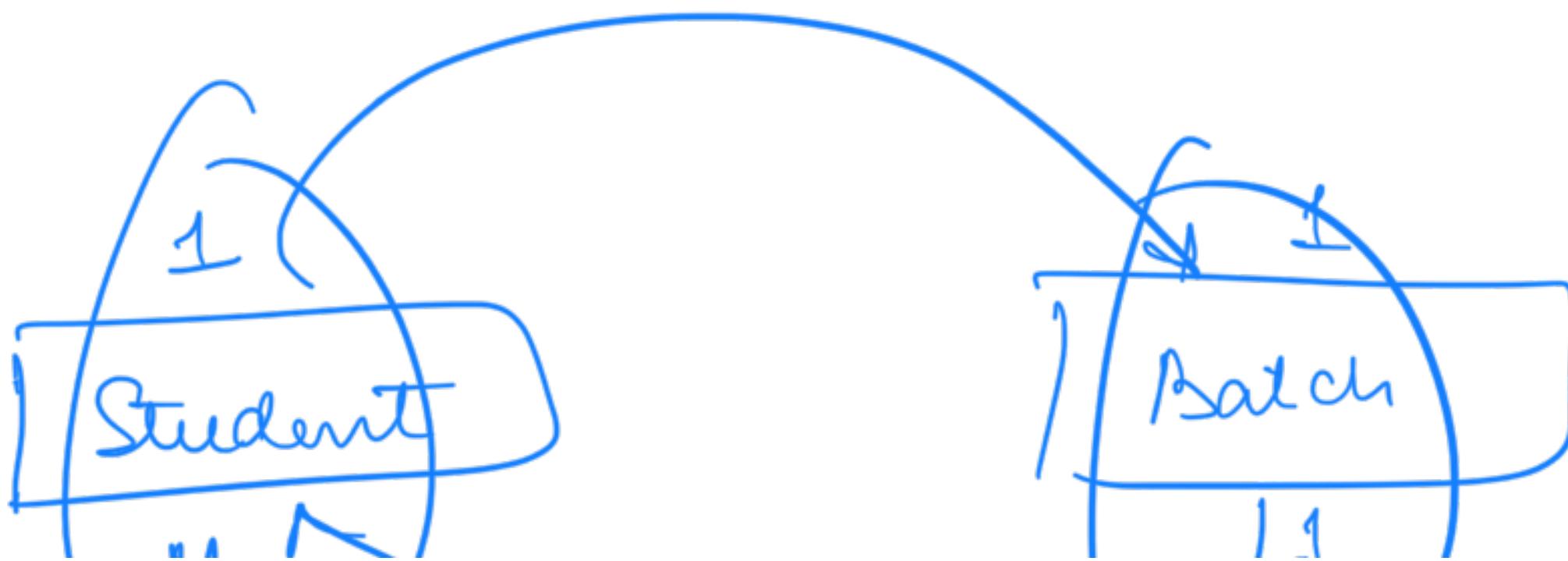
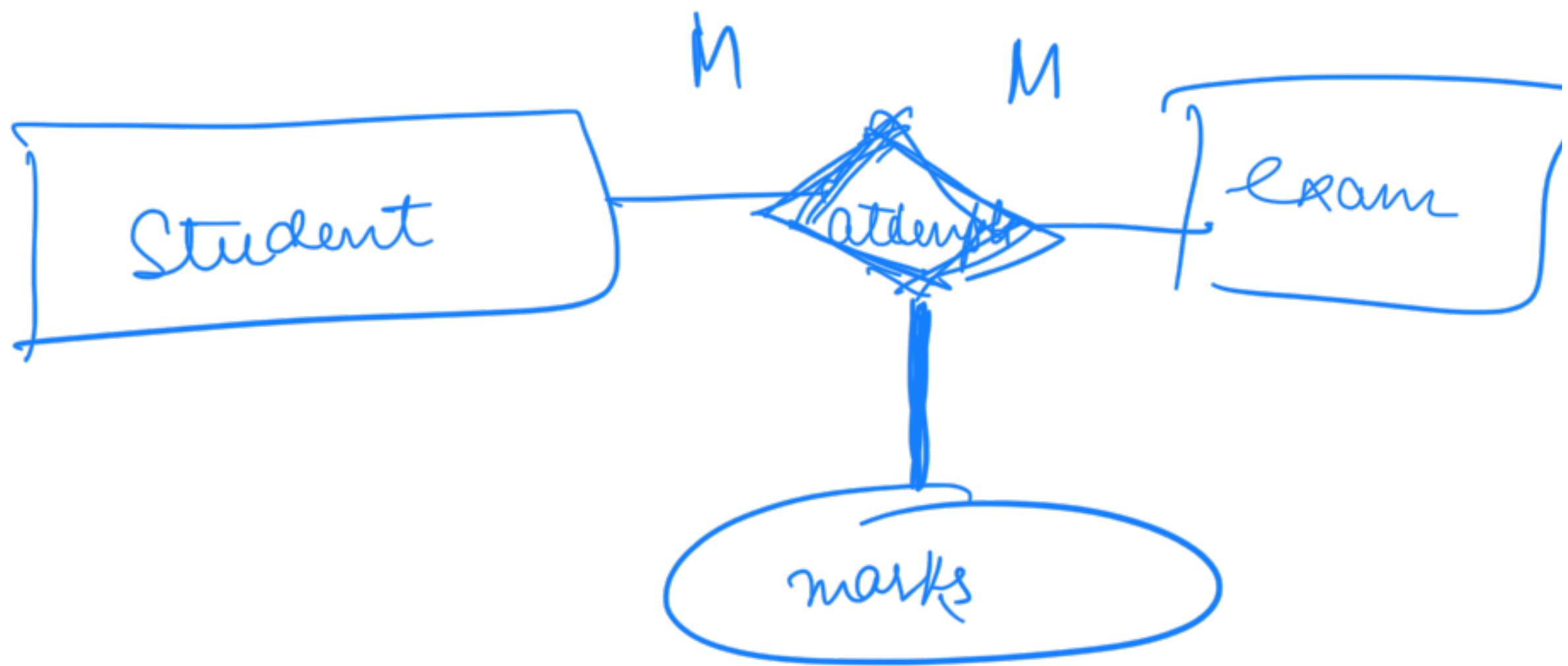
Students

id	Name	email	phNo	<del>exam ids</del>

Exams

id	Name	duration	<del>st- ples</del>

student	ex_id	Mark



M

U

M : 1

Students

id	Name	email	batch

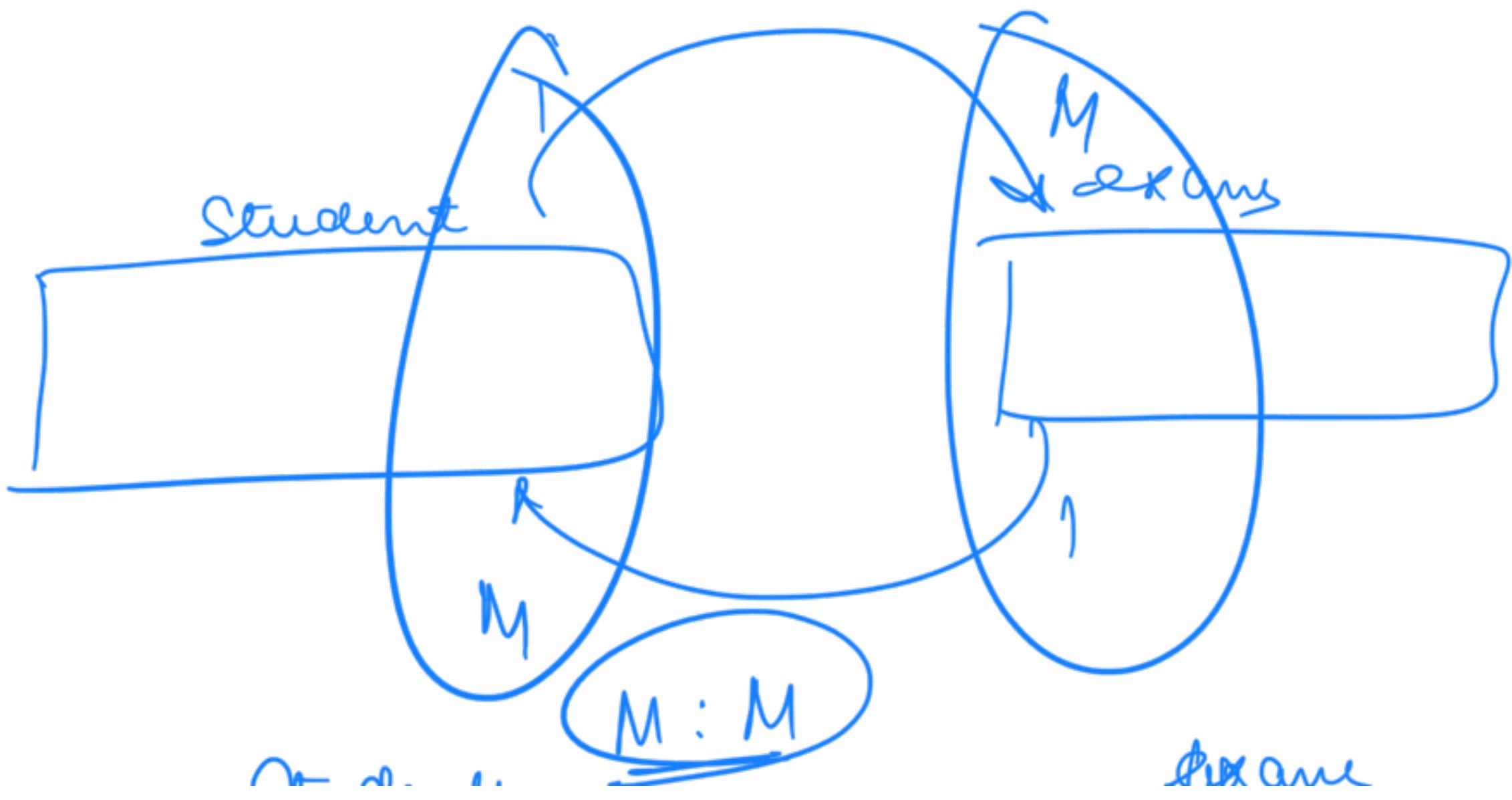
batches

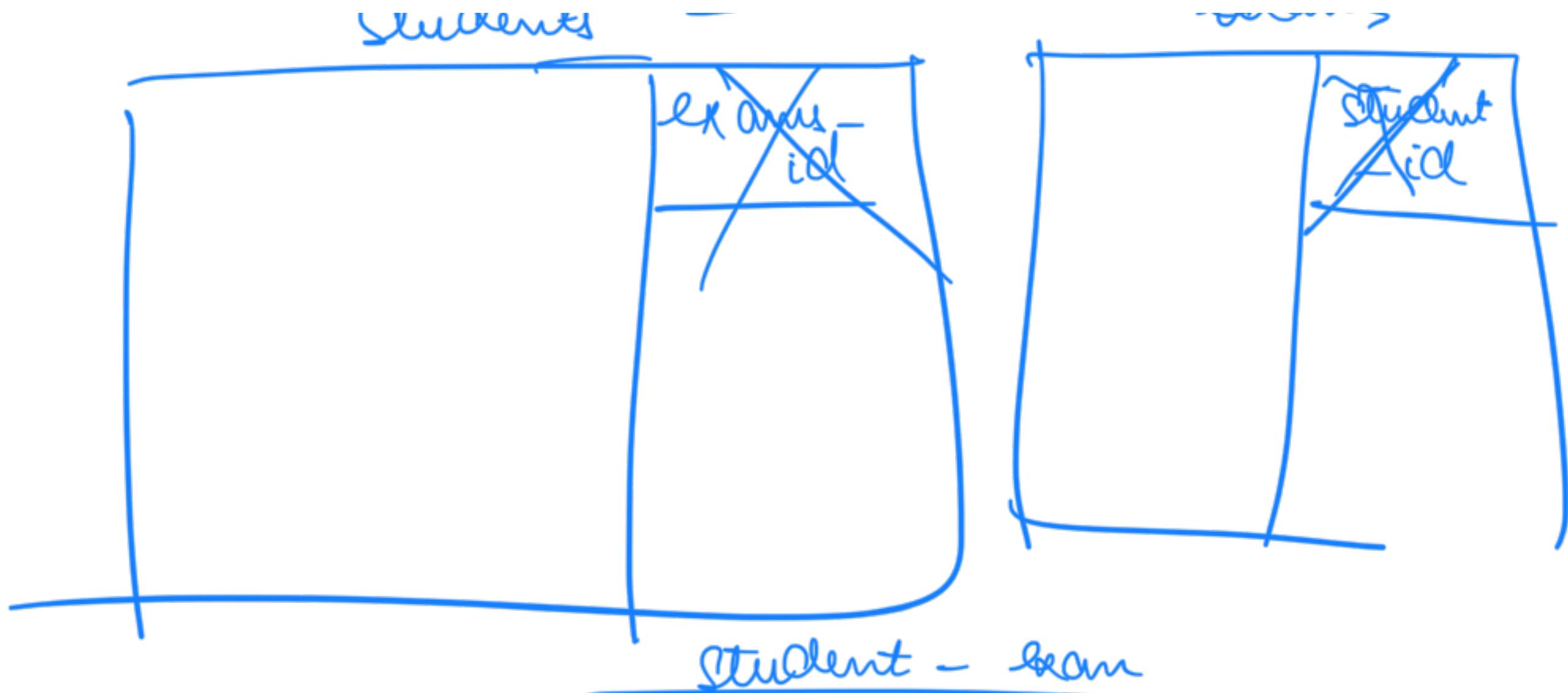
id	name	st id

Normalization

Transaction

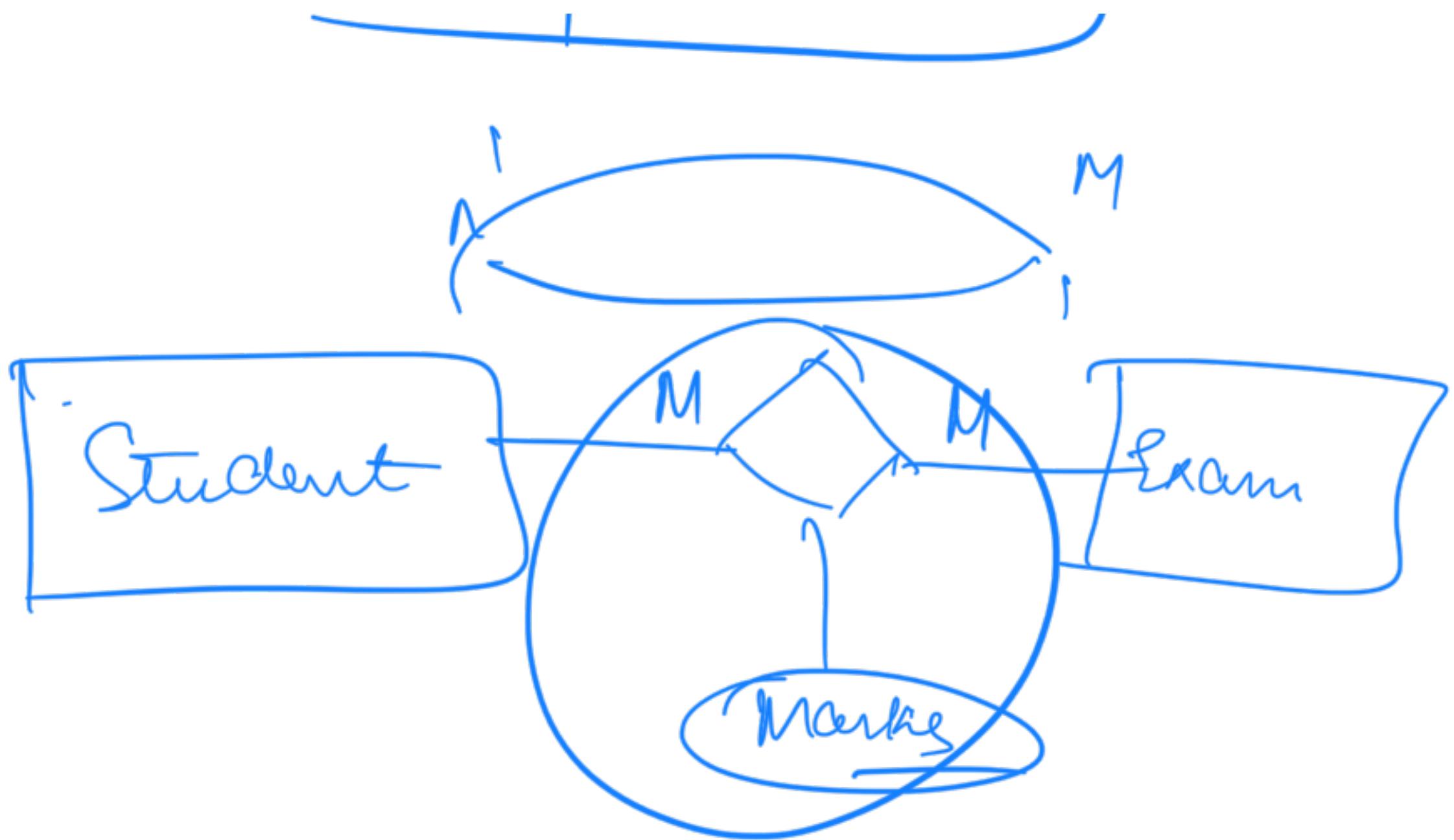
~~Access SQL 1, 2~~

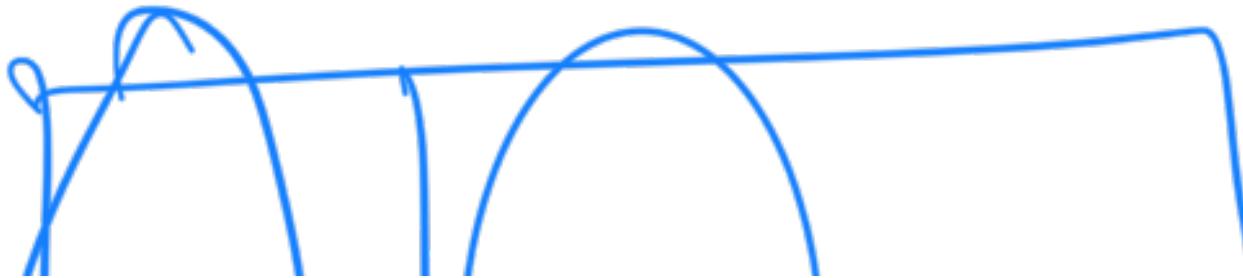
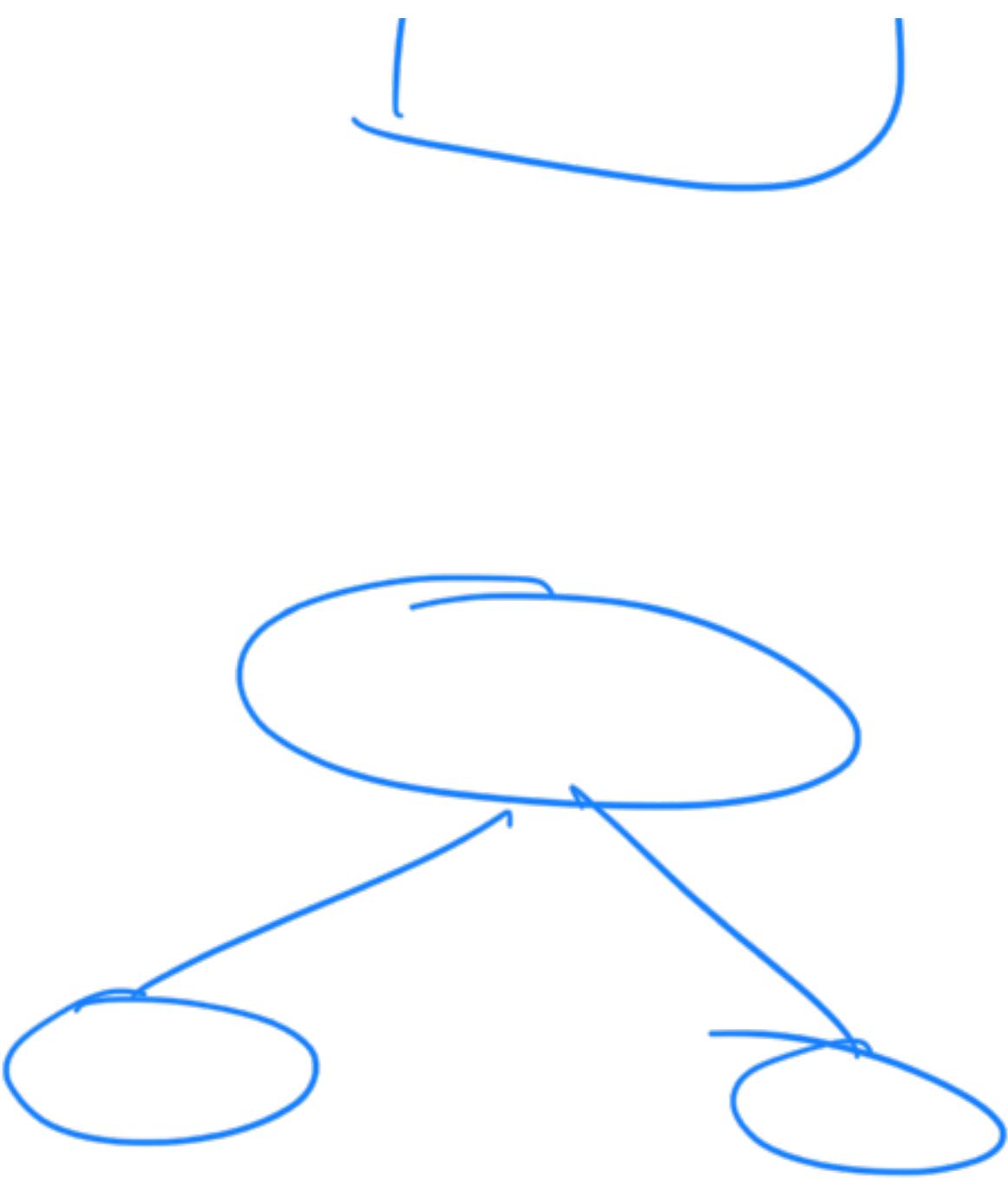
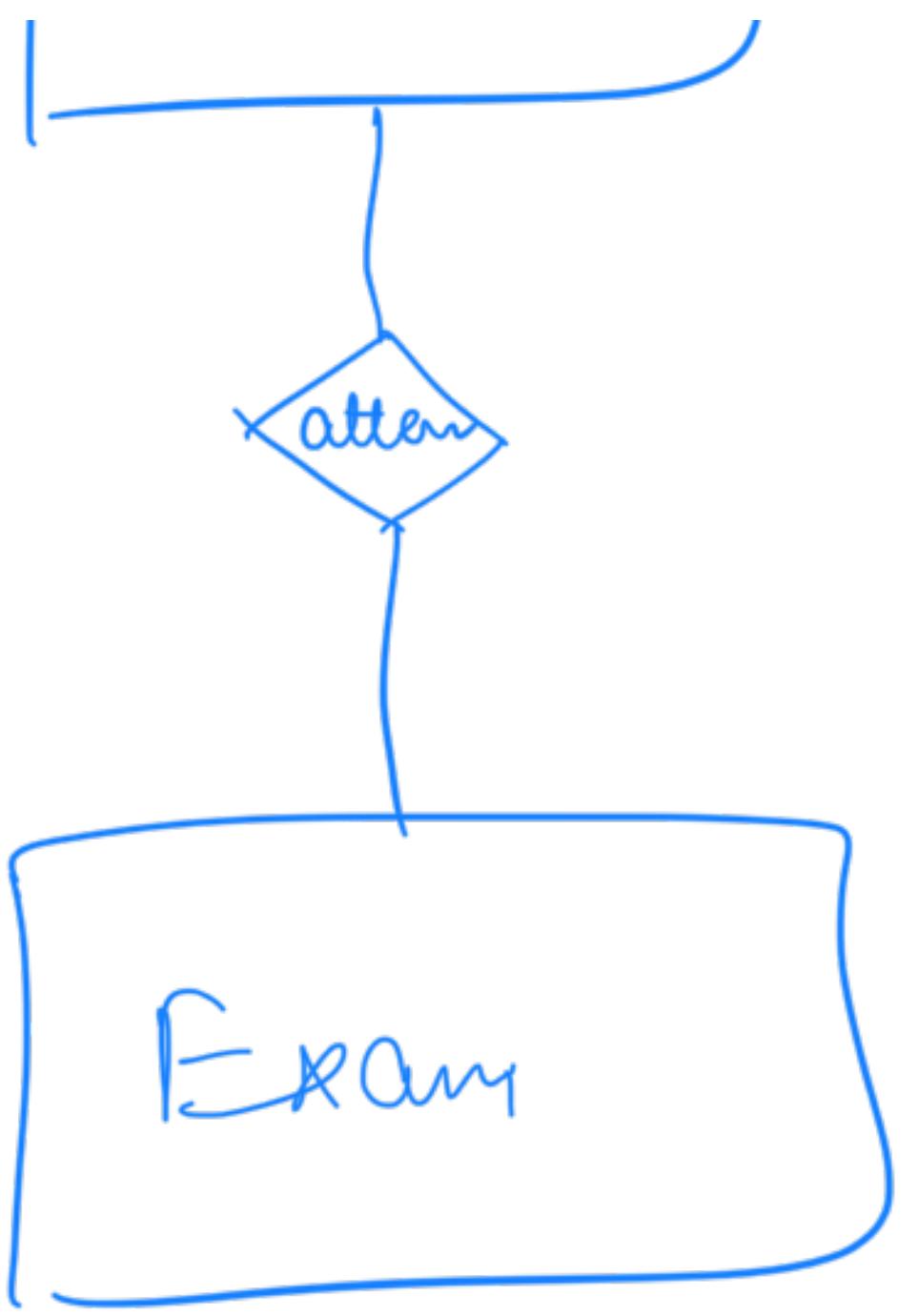




Student - Exam

st_id	exam_id	marks







RM = Schema Design

func → ER Diagram