2) Schema design

⇒ How to decide what tables are needed

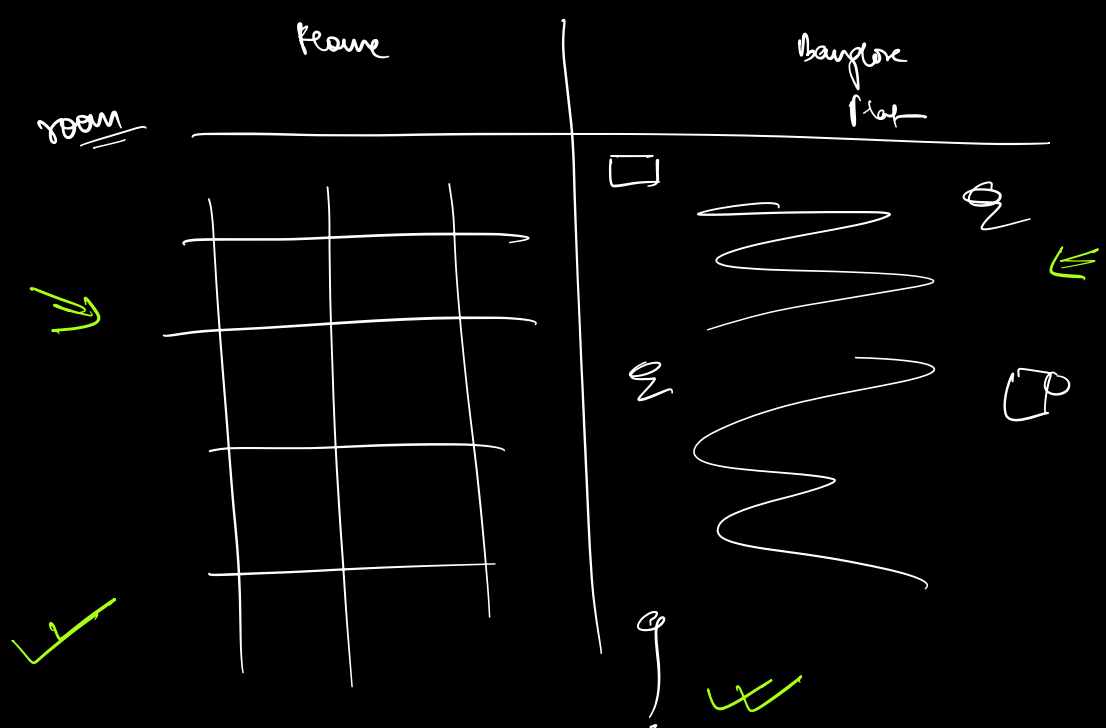
⇒ How to model cardinality

⇒ [3) Normalization]

LL Design ⇒ Classes diagram] UML
Use case diagrams]

API design

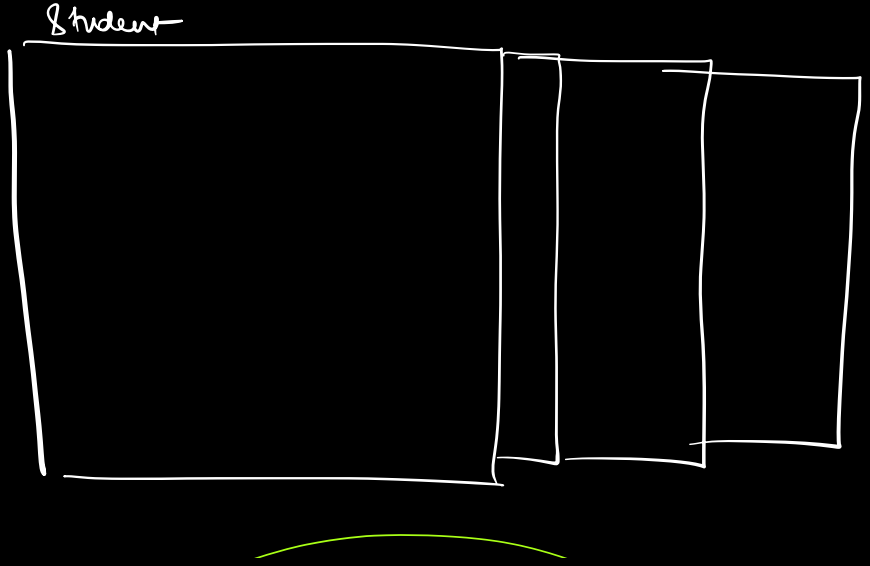
DB design (Schema design

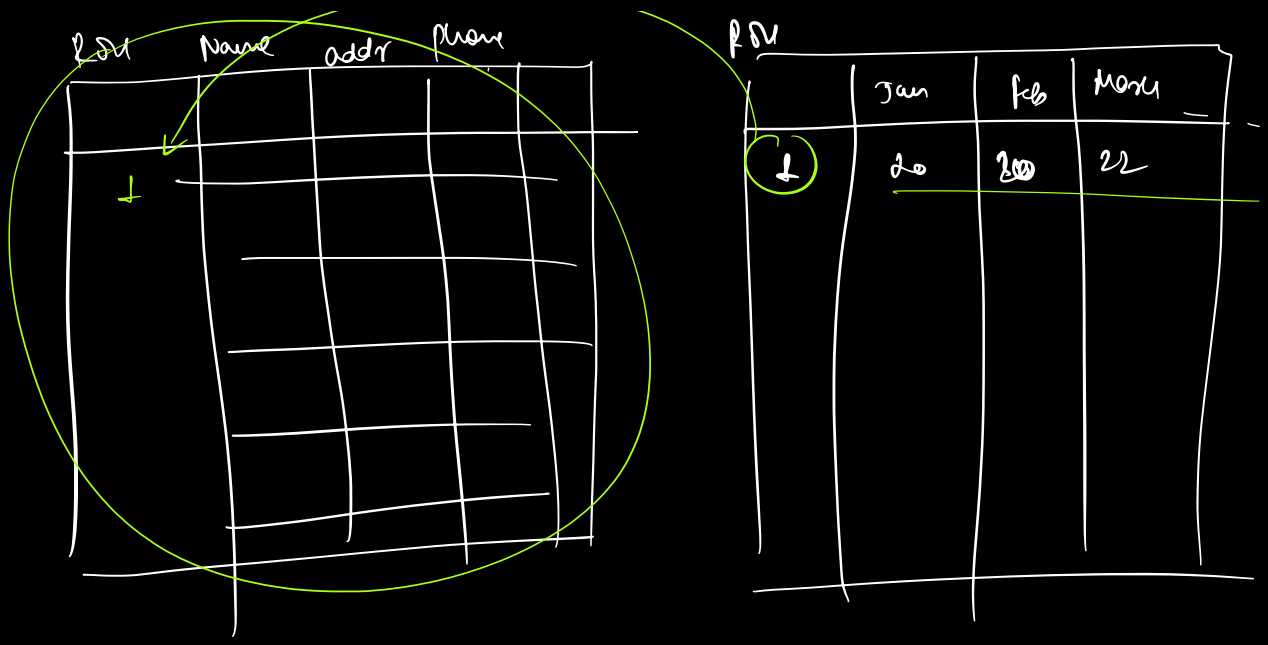


DB Schema ⇒ pictorial rep. of how my structure of DB will be, or how we are going to store data in our DB.

⇒ How DB will be structured

Row	Siddhant	01/07	02/07	03/01
1				
2	Sushil			
3	Gaurav			
4	Madhvi			





⇒ Schema design is a part of design document

⇒ How to approach schema design :-

Case Study ⇒ Scaler

Requirements ⇒ Read / Analyze / Breakdown ⇒ Design docs [Use Case, Class, DB Side]

↓

Coding

What ⟶ ~~How~~

I) There are several batches at Scaler. Each batch has students, name, instructor, start month, timings.

II) Each batch has classes. Each class has, name, instructor, assignments.

III) Every student has name, grad year, university, phoneNo, email, workExp.

IV) Student get mentors. Mentor has name, exp, noOfSessions.

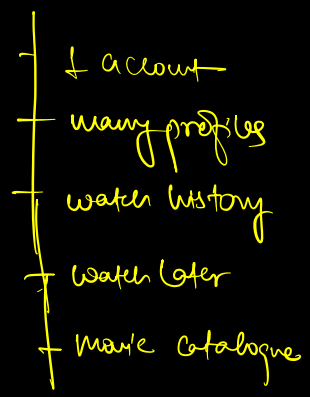
V) A student may move from one batch to other. We would want to know the dates of student joining / leaving a batch.

		Join	Leave
Harshal →	Aug 22 Batch –	22/08/22	22/10/22
	Nov 22 Batch –	28/11/22	—

	Join	Leave
~~oldBatch~~ currentBatch	HAS TO BE PRESENT	NULLABLE
~~newBatch~~	—	—

=) Assgn =) Try to create schema design of Netflix.

⇒ Design



S-1 ⇒ Find out all the entities in the req^n.

Entity ⇒ something for which we would want to store more information

- real life
- conceptual

⇒ Find the NOUNS in req^n

⇒ For every NOUN, check if we need to store more information about it

⇒ If Yes create a table for this entity.

⇒ Batch, Student, Mentor, Instructor, Class, Assignments

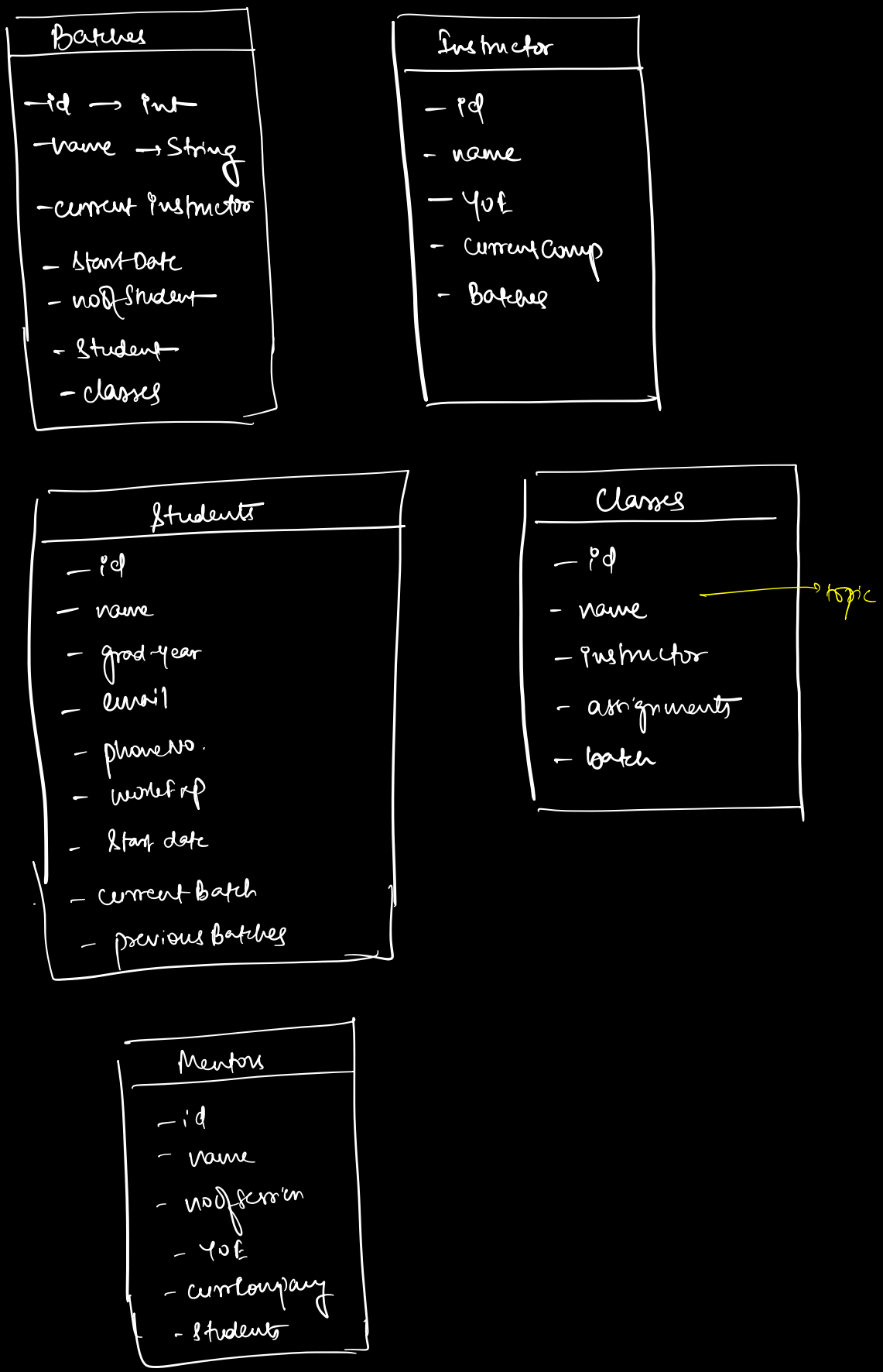
Tables name ⇒ plural of entity name

Batch ⇒ Batches

Student ⇒ Students

& go with company conventions.

Assgn ⇒ [GUID] UUID ⇒ read ⇒ how to generate


Batches
- id → Int
- name → String
- current Instructor
- Start Date
- noOfStudent
- Student
- classes
Instructor
- id
- name
- YOE
- CurrentComp
- Batches
Students
- id
- name
- grad year
- email
- phoneNo.
- workExp
- Start date
- current Batch
- previous Batches
Classes
- id
- name
topic
- Instructor
- assignments
- batch
Mentors
- id
- name
- noOfSession
- YOE
- currCompany
- Student

⇒ Conventions :

↳ Attributes

double ⇐ ✓ 1) CamelCase ⇒ currentInstructor

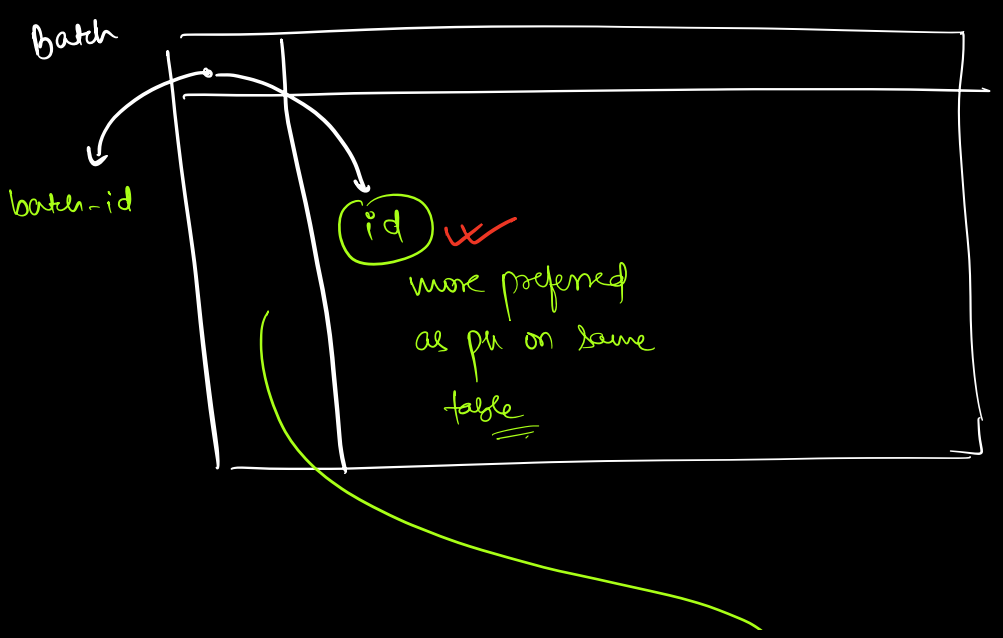
common in SQL ⇐ ✓ 2) underscore ⇒ current_instructor

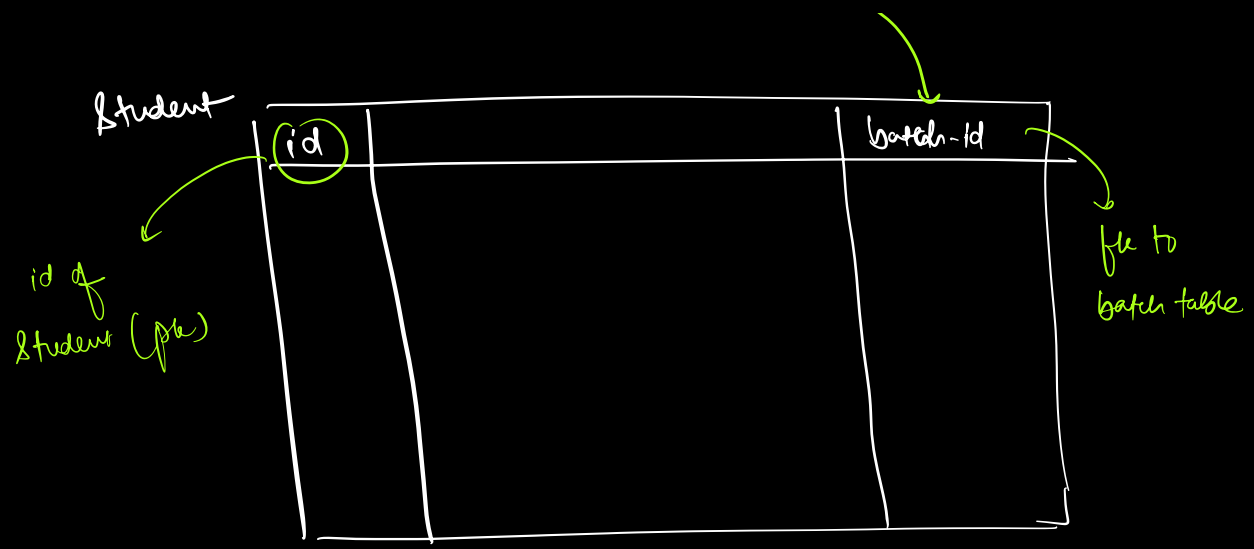
possible but very wrong ⇐ 3) [normal english ⇒ current instructor] ✗

SQL is case insensitive

⇒ Student
Student
STUDENT
] same

↳ IDs





=> Cardinality => rel^n b/w entities.

2 entities can be related to each other

i) 1 : 1

ii) 1 : M

iii) M : 1

iv) M : M

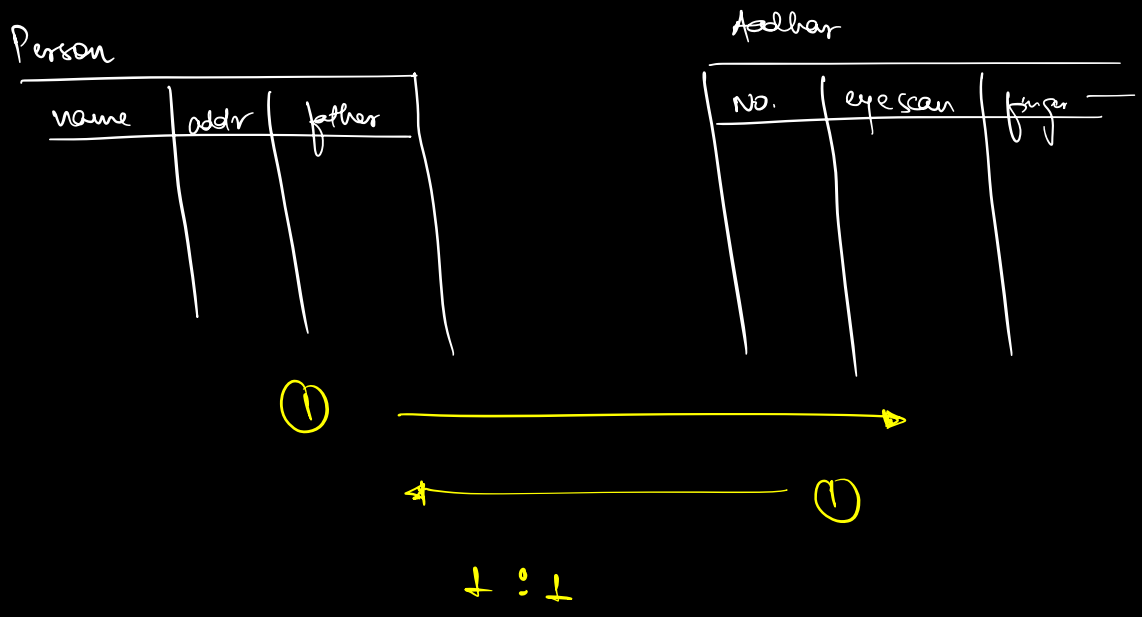
Batch — Instructor

(A) (B)

If two entities A & B are connected to each other, then, cardinality =>

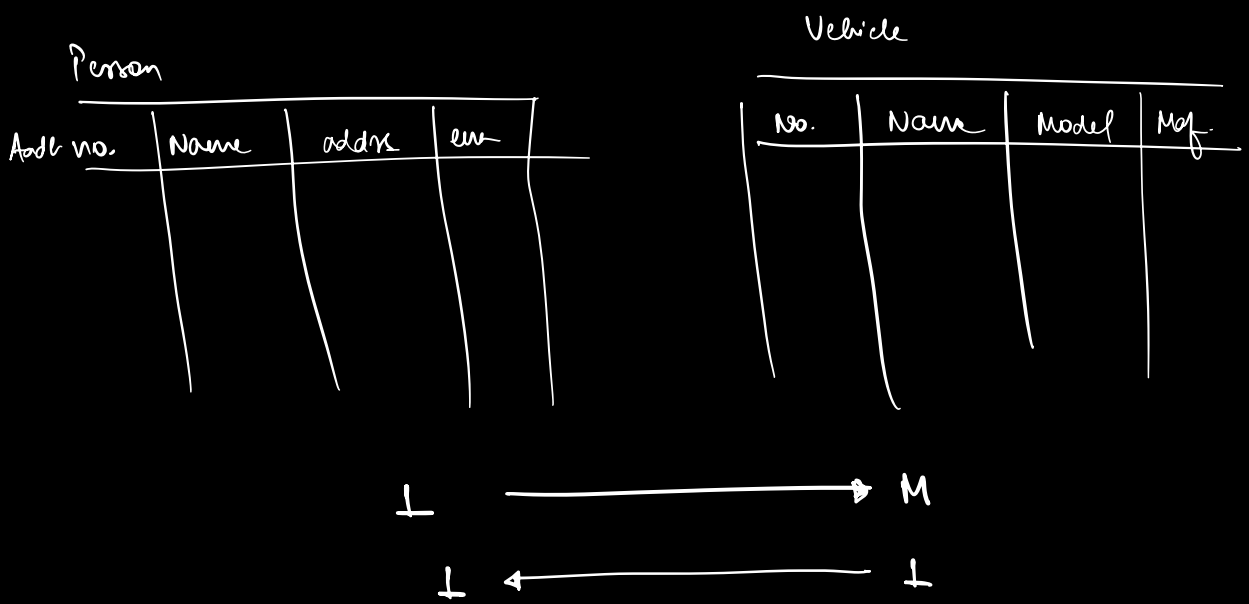
How many of A : How many of B.

(1) 1 : 1


Person
name
addr
father
Aadhar
No.
eye scan
finger
1
1
1
1

1 : 1

(2) 1 : M


Person
Adh no.
Name
addr
em
Vehicle
No.
Name
Model
Mfg.
1
M
1
1

1 : M

iii) M : M

Student

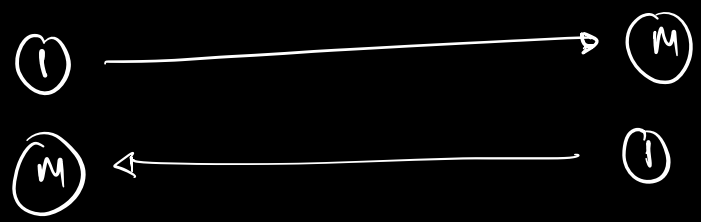
id	name	addr	phone

Courses

id	name	type	fees
	Code BTech Swimming Gym - -		

Rakesh ⇒ BTech / Scaler / Swimming

Gym ⇒ Rakesh, Madhu, Rizwan



M : M

Steps for cardinality :-

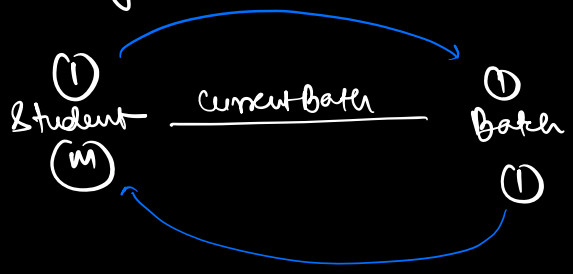
S1 ⇒ Find the relⁿ we are talking about b/w A & B.

A ⇒ Student

B ⇒ Batch

relⁿ ⇒ current-batch

So ⇒ how many A : how many B.

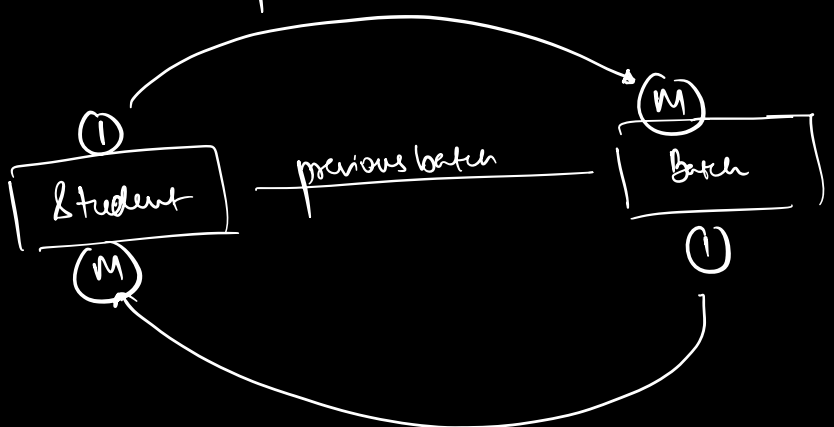


M : 1

eg ⇒ A = Student

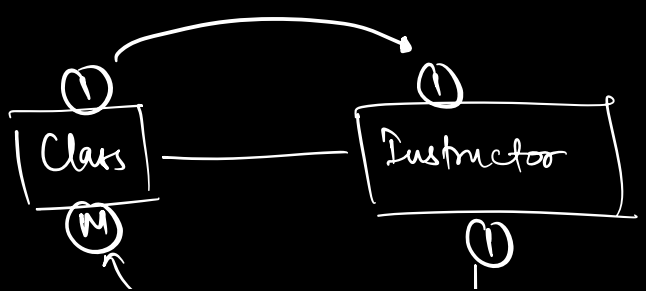
B = Batch

rqm ⇒ previous batches



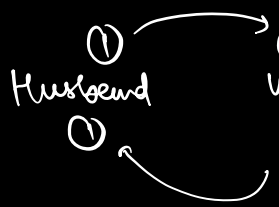
M : M

eg ⇒

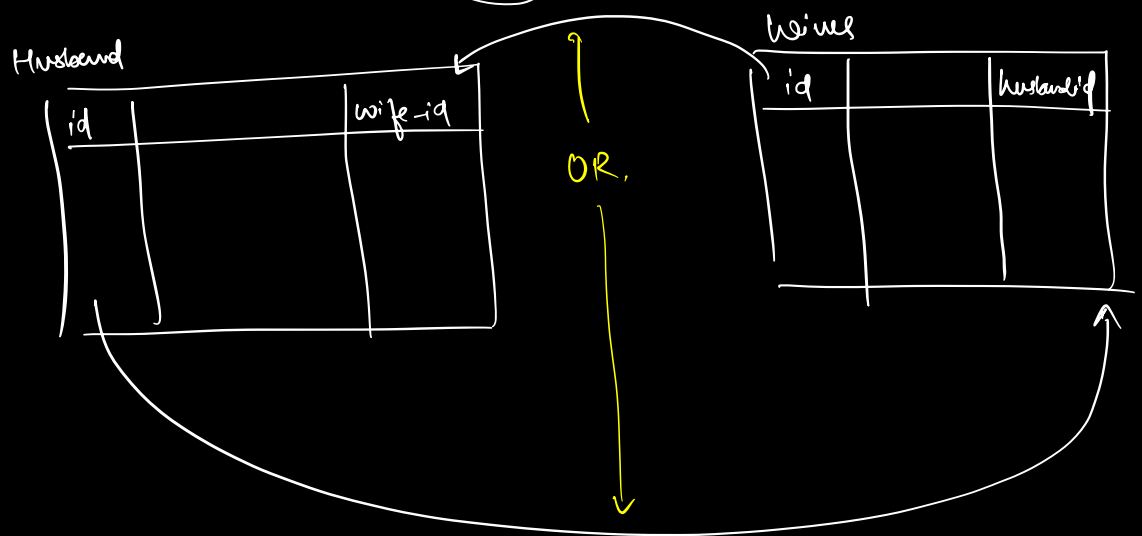


M : 1

⇒ How to represent multiple cardinalities in DB :-



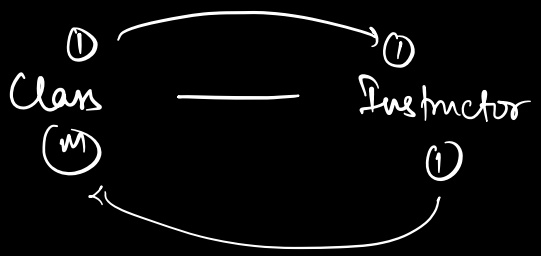
⇒ 1 : 1



1 : 1 ⇒ put the id of one side to other side.
either way is fine.

Husband-id in wife table

or wife-id in husband table.



M : 1

1 : M | M : 1 ⇒ put the id of one side to many side
(1) (M)

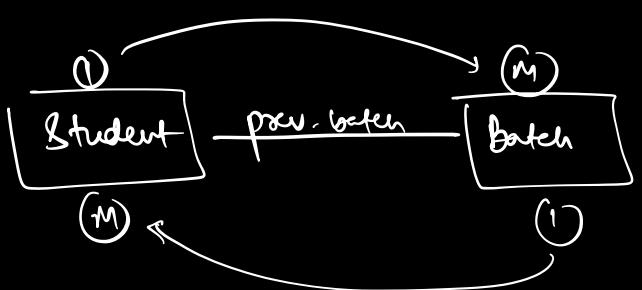
(M) Classes

id	name	date	module	ins_id

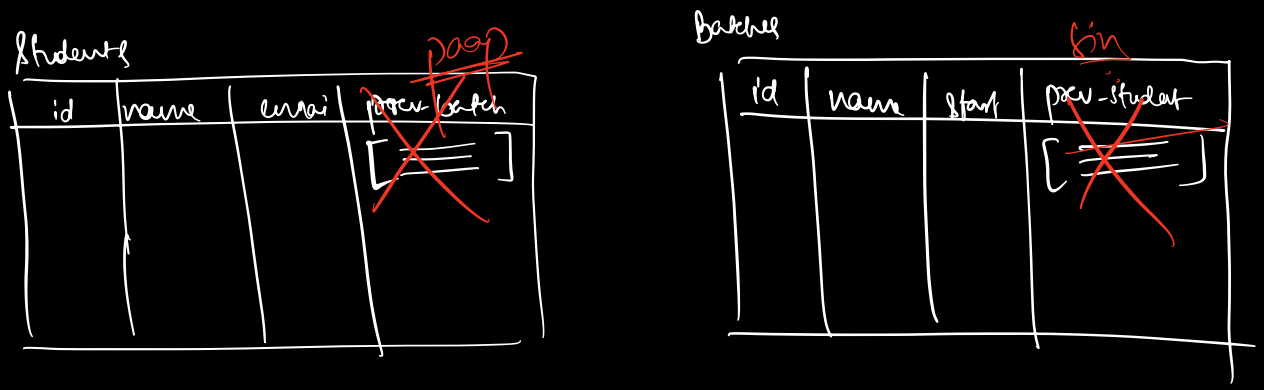
(1) Instructor

id	name	exp	email

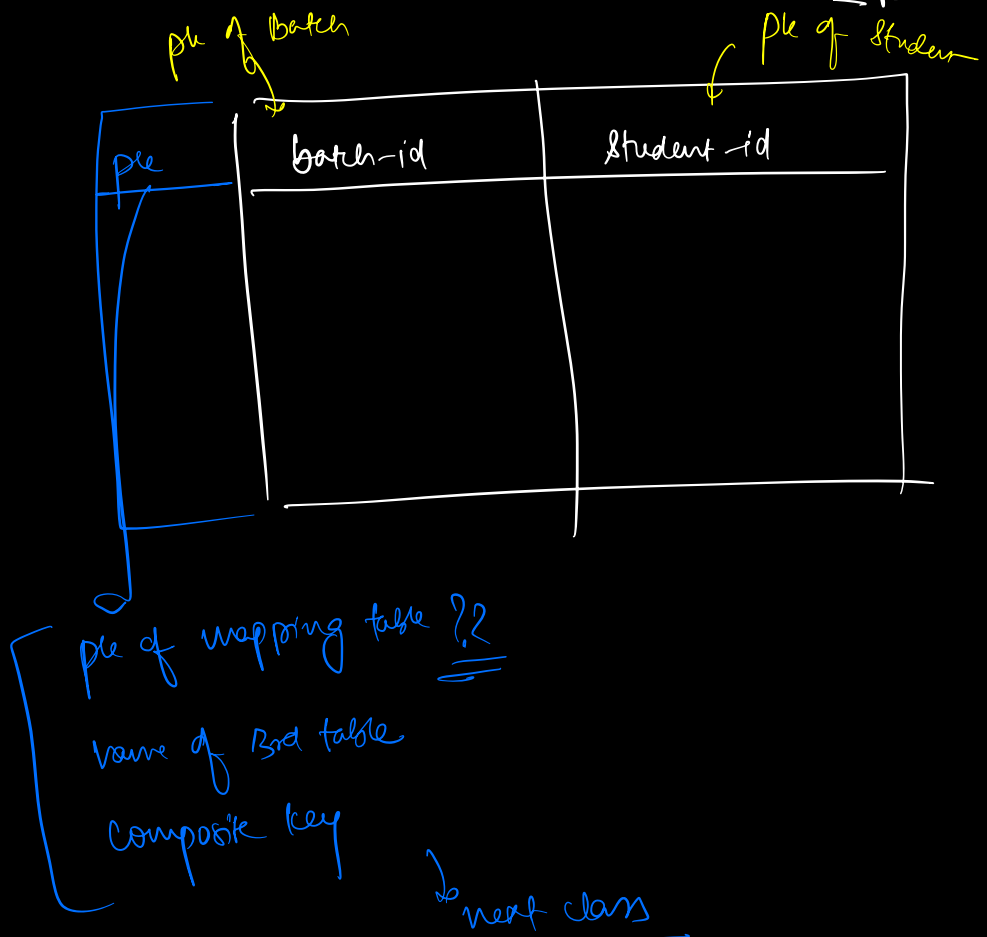
M : M



M : M



=> Create a 3rd table called mapping / lookup



pk of mapping table ??

name of 3rd table

composite key

→ next class