

- 1.) Schema design ✓  
    what / why / how
- 2.) Approaching schema design ✓
- 3.) Finding cardinality ✓  
    ↳ Representing cardinalities

start by 9:05 PM IST

## \* ) SCHEMA DESIGN:

schema → structure of d/b.

# tables

what columns | fk | pk | indexes

pictorial representation of d/b.

Requirement → scales



created tables directly.

PROBLEMS: ??

students

ID	NAME	PHONE-NO	EMAIL-ID
1	Yash	-	-
2	AMAN	-	-
3	TARIQ	-	-

20,000

Need for schema design

→ to restrict scope of changes going fwd.

## \* APPROXIMATION SCHEMA DESIGN:

Eg → scalar

- ✓ 1.) scalar has multiple batches
- ✓ 2.) each batch → name / curr-instruc / start-month
- ✓ 3.) each batch has many students
- ✓ 4.) each batch can have many classes
- ✓ 5.) each class → name / date-time / instructor
- ✓ 6.) for every student → name / gradyear / univ / email / phone-no
- ✓ 7.) every student has a buddy (who is also a student)
- ✓ 8.) a student can move from batch1 → batch2
- ✓ 9.) for each batch student attends, store start-date of that batch
- ✓ 10.) every student has mentor
- ✓ 11.) each mentor stores name / curr-company
- ✓ 12.) store info on all mentor-sessions
  - \* : time / duration / stud / mentor / stud-rating / men-rating
- ✓ 13.) for each batch → store whether DML / Academy.

## \* STEPS:

s1) CREATE TABLES: / Identify tables.

→ Find all NOUNS in query. → store data.

→ tables in schema design.

- 1.) batch
  - 2.) student
  - 3.) class
  - ~~= 4.) instructor~~
  - 5.) Mentor →
  - 6.) MentorSession.
- 

Naming conventions : (tables)

batch  
✓ batches v/s

Plural ✓

Batches / BATCHES / batches

snake case  
mentor - sessions

column - names → singular + snake - case

s2.) In your tables:

- ✓ 1.) Add 'id' Pk → 2.2
- ✓ 2.) Add all other attributes (NON Relational)

why 'id' ??  
→ int / long int

→ NAMING CONVENTION ON Pk:

- ① id  
✓ ② batch-id → advantage

JOINS  
→ using / Natural JOINS.

① batches

batch-id	name	start-month
----------	------	-------------

② instructors

instructor-id	name	email	avg-Rating	curr-company
---------------	------	-------	------------	--------------

③ student

student-id	name	email	Phone-No	Grad-Yr	univ
------------	------	-------	----------	---------	------

④ Class

class-id name date-time

⑤ Mentor

mentor-id name wife-company

⑥ mentor-sessions

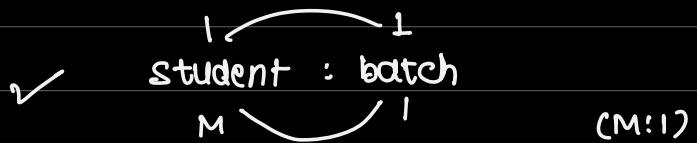
mentor-session-id time duration s-Rating m-Rating

As of now → Relations are Missing.

Q3.) Representing Relations:

CARDINALITY: degree of Relation b/w 2 tables.

How many of (x) : How many of (y)



TYPES:

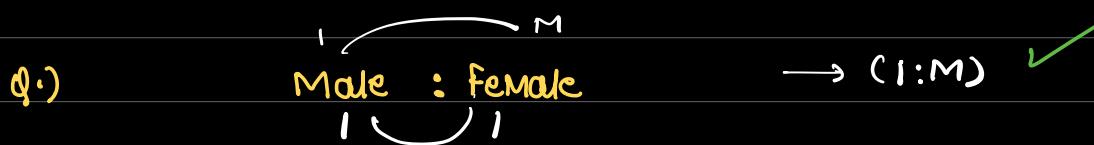
1:1

1:M

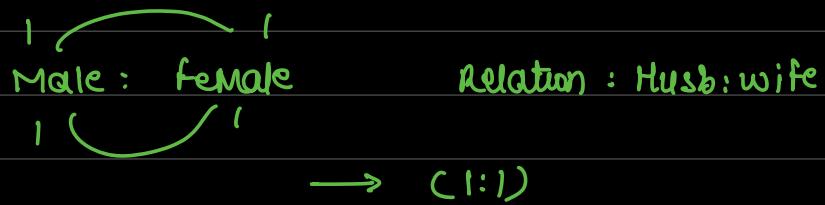
M:1

M:M

father      daughter



what is cardinality ??



→ DIRECTLY calculated → XXX

\*\*\*

cardinality: Always find Relation b/w ② tables  
before going for cardinality.

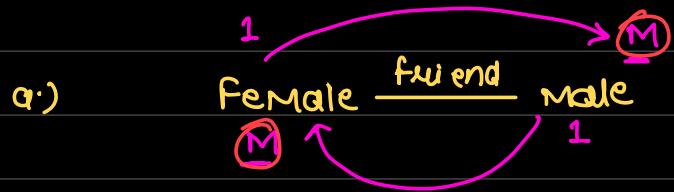
\* How to find card. b/w ② tables -

# steps:

s0.) find out ② entities / tables

≡ s1.) find out relation for which you need  
to find cardinality.

s2.) 2 step process -

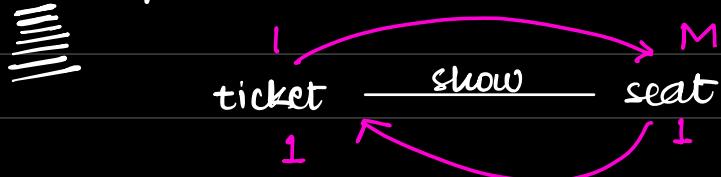


b.)  $(M : M)$  if  $(M)$  present  $\rightarrow$  put  $(M)$  on that side.

$\rightarrow \checkmark$

### # EXAMPLES:

In BookMyShow example



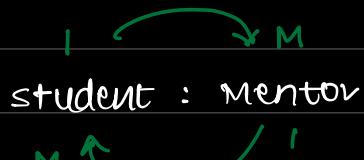
$(1 : M)$

# Husband  $\leftrightarrow$  wife



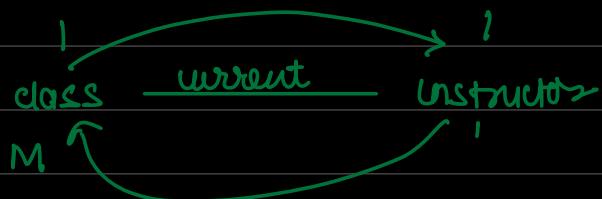
1:1

# student  $\leftrightarrow$  Mentor at scalar



(M:M)

# class  $\leftrightarrow$  wkt instructor



(M:1)

break till 10:40 PM IST

\* REPRESENTING CARDINALITIES:

1) 1:1

Husband	wives	(1:1)
id name wife-id	id name husb-id	

solution → Put 'id' of any ① side on other side



Mentor

$\text{id name student-ids CI}$        $\times$       NOT CORRECT

student

$\text{id name Mentor-id}$        $\checkmark$       correct way.

solution: Put id of (1) side onto (m) side  
 $\checkmark$

iii)  $M:M$

orders in Amazon

order

$\text{id Name}$

$1 \rightarrow M$

$O:P$

$M \uparrow 1$

Product

$\text{id Name}$

$\rightarrow (M:M)$

i) order

$\text{id Name Product-ids CI}$        $\times$       NOT CORRECT

2.) Product

id name order-ids [] x NOT CORRECT

\*\*) Solution: (M:M) → Create a Mapping table

order-products → (Newtable)

order-id product-id

Revision-

1:1 any id on any side  
1:M / M:1 id of (1) on (m) side  
M:M Mapping table }  
                        {

\*) SPECIAL CASES while Representing:

① 1:M / M:1

student

id NAME Mentor-id

Mentor

Mentor-id Name

scalars :  
              := 20M Rows  
              2 CL.

70% →

Sparse Relation

(too many Nulls)

1 Yash NULL

2	xyz	<u>Null</u>	✓	PROBLEM →
3	abc	Null		① TOO Many Nulls.
4	MNC	Null		loosing on space
5	cde	1		② memory wastage
6	OPQ	2		

Solution → create a mapping table

student-mentor

student-id      mentor-id

✓

x Nulls here

1:1 ✓

Husband: wife

sparse relation → Any cardinality.

②  
==

special case:

(@ Any cardinality)

Husband			wife	
id	Name	wife-id	id	Name
		→		

Req → Marriage-date

✓ → husband

# first-meet-date → husband

:

#	husband	.	.			
	id	Name	wife-id	marr-date	f-meet-id	....
→				≡	≡	
				↓	↓	

attribute of relation b/w  
Hus. <>> wife

Solution → ✓ create New Mapping table

Husband - wives

h-id	w-id	f-Meet-date	m-date	....
		==	==	

- (S1) identify ↴
- (S2) create ↴ (NON Relations)
- (S3) Represent Relations → Relation - ==

S3.)

1.) batch has instructor

batch : instructor → (M:1)

id of cust on batch side

# batches

batch-id, name, start-month, instructor-id

- ② Each batch has students

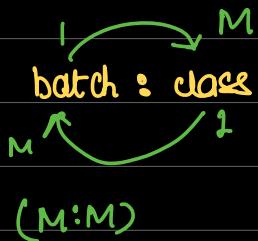
batch : student  $\rightarrow$  (1:M)

id of batch on student

student

batch-id

- ③ Each batch many classes (NOT live, overall)



solution  $\rightarrow$  Mapping table

New table:

batch-classes

batch-id      class-id

# Homework:

$\rightarrow$  Complete all tables with their relations -