

Schema Design I

Agenda

- ① What is schema design
- ② How to approach schema design
- ③ Conventions
- ④ Representing cardinalities
- ⑤ Deciding Primary Key → When should PK has multiple values
- ⑥ Sparse table
- ⑦ Relations with attributes

What is schema design, → solving how should something be done → considering the constraints



The structure of the database

→ what all tables

↳ What all columns in the table

↳ PK

↳ FK

↳ Indexes

In other words,

it is pictorial rep of how the DB is structured

Design → pictorial rep considering all the constraints.

→ Before tables etc are created, they create a design document.

→ schema

→ class diagram

→ architecture diag.

How to approach Schema design

Design Scaler.

Requirements

- All the entities about which I need to store data?
- ① Scaler will have multiple **batches**. About each batch we have to store their name, start month, current **instructor**.
 - ② Each batch of Scaler will have multiple **students**.
 - ③ Each batch has multiple **classes**.
 - ④ For each class we have to store the name of the class, date and time of the class, instructor of the class.
 - ⑤ For every student we store their names, grad year, university name, email, phone number.
 - ⑥ Every student has a buddy who is also a student.
 - ⑦ A student may move from one batch to another.
 - ⑧ For each batch a student goes to we have to store the start date of that batch.
 - ⑨ Every student also has a **mentor**. For every mentor we store their name and current company.
 - ⑩ We have to store information about all **mentor sessions** (time, duration, student, mentor, study setting).
 - ⑪ For every batch we have to store if its an Academy batch or DML batch.

STEPS

STEP 1 → Creating the tables.

→ How to identify tables

- ① Find all the nouns that are there in the table
- ② for each noun ask if you have to store data about that entity in your dB.
- ③ if yes : create a table
if no: move on

Conventions about names.

- ① Name of the table should be plural
Because it is storing multiple things
mentorSession
MentorSession
'mentor_sessions' (snake case)
- ② Name of the col is a singular → snake case.

* Representation of the schema doesn't matter

what matters is whether you have

all tables need to satisfy requirements.



Schema

batches



instructor



students



classes



mentors



mentor_sessions



Step 2 — In the table that you just created, add the id(primary key), all attributes about the entity.

→ Dont care about FK

→ Dont care about relation to other entities.

Expectations with Primary Key.

① It should rarely change.

(why → update index
→ re-sort data on disc)

② It should ideally be a data type easy to work with

(why → ① sort
② less size)

⇒ Have a separate column integer/ big integer called as 'id' as a PK.

18
10

Convention w.r.t naming key column.

① calls it 'id'

② Call it {table_name}-id / {entity}-id

eg book-id

batches

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

instructor

instructor-id	name	email	avg-rating
---------------	------	-------	------------

students

student-id	name	grad-year	uni-name	email	phone-no
------------	------	-----------	----------	-------	----------

classes

class-id	name	scheduled-time
----------	------	----------------

mentors

mentor-id	name	current-company
-----------	------	-----------------

mentor_sessions

mentor-session-id	time	duration	stud-settng
-------------------	------	----------	-------------

What is missing? Relations.

Step 3 → Represent relations

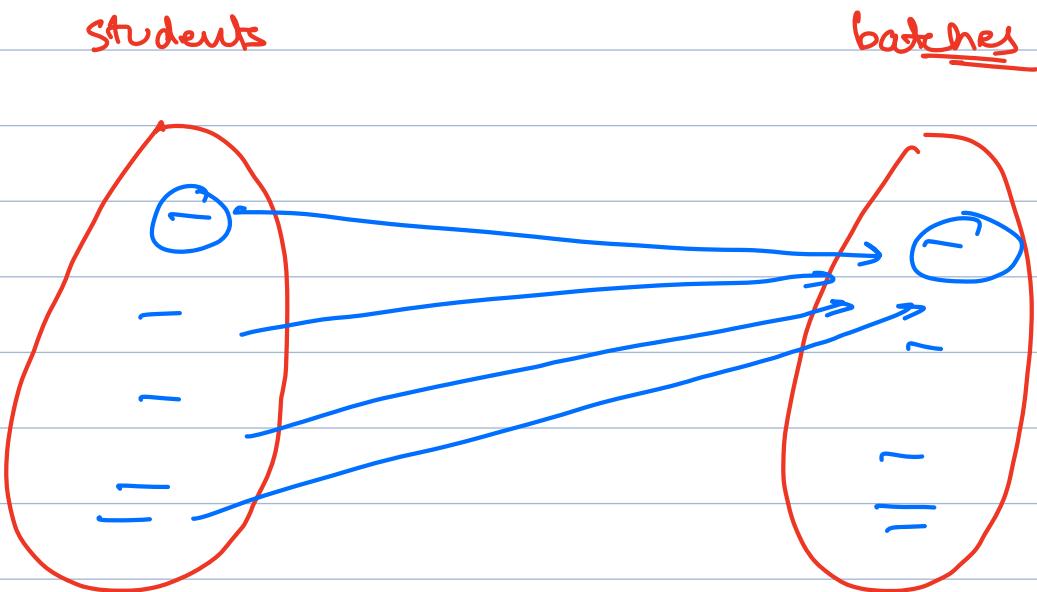
How to represent relations.

Cardinality → how many of one's is related to how many of others.

e.g. student \Rightarrow batch

Cardinality → how many students are related to how many batches and vice versa,

- Possibilities.
- $\rightarrow 1:1$
 - $\rightarrow 1:m$
 - $\rightarrow m:1$
 - $\rightarrow m:m$

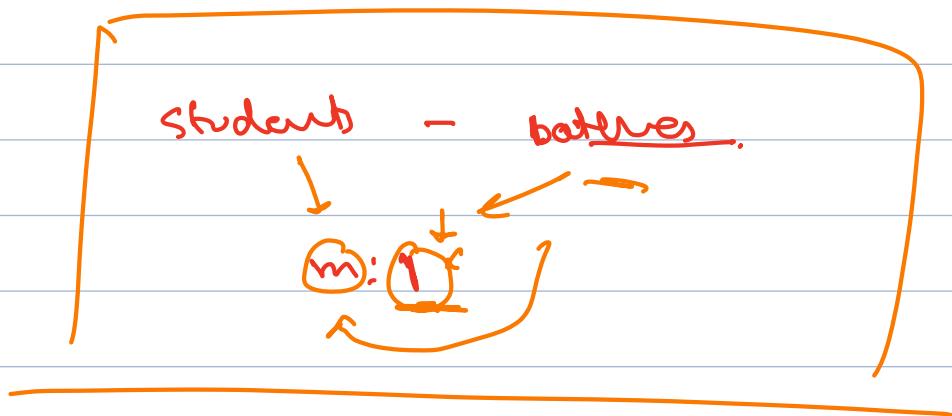


① → an entity can be associated to at most 1 instance.

[0, 1]

② → an entity can be associated (even if something) to more than one instance.

[0, 1, 2, ..., ∞]



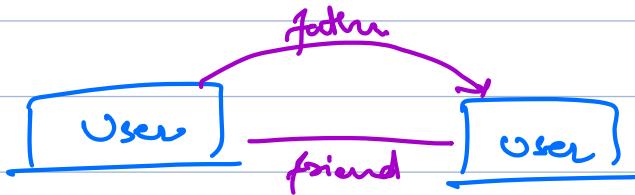
How to represent cardinalities in relations relations



① Find the relation b/w the entities ..

→ cardinality depends on entities & their relation.

→ b/w 2 entities there can be multiple type of rel". Each type of rel" can have a diff cardinality.



- (a) find 2 entities b/w which you have to find cardinalit.
- (b) find what rel" b/w them you have to find cardinality.

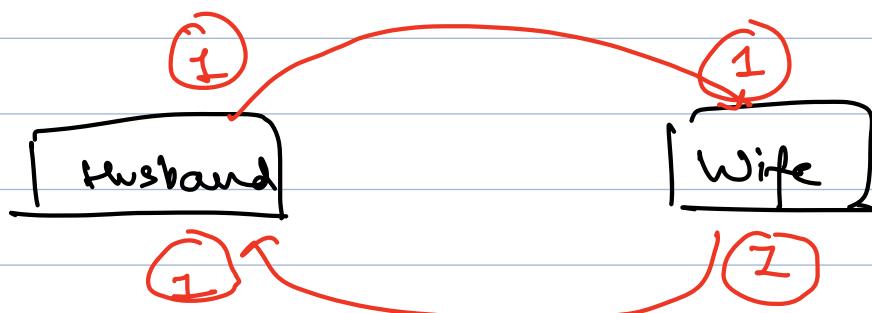
② → 2 step approach.



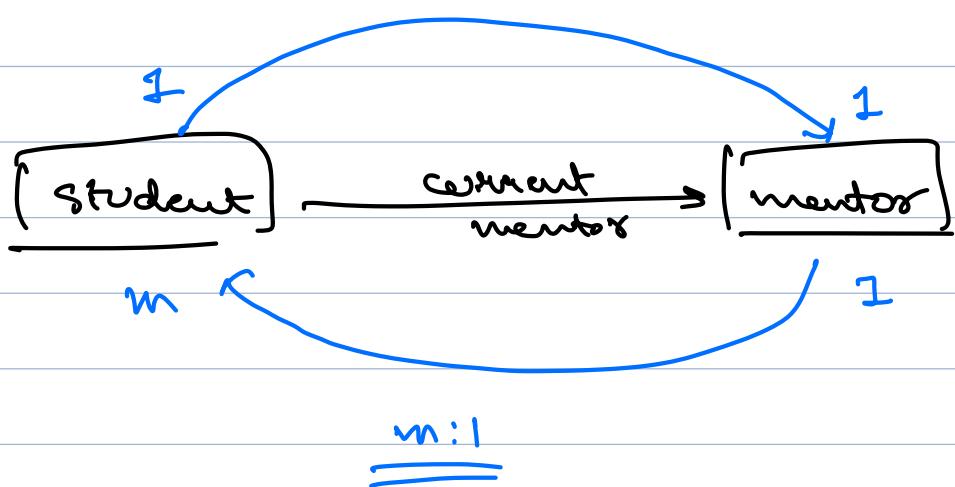
- ① go left to right putting 1 on left side
- ② go right to left putting 'm' on right side
- ③ if there is 'm' on any side, put 'm' on that side in final ans.



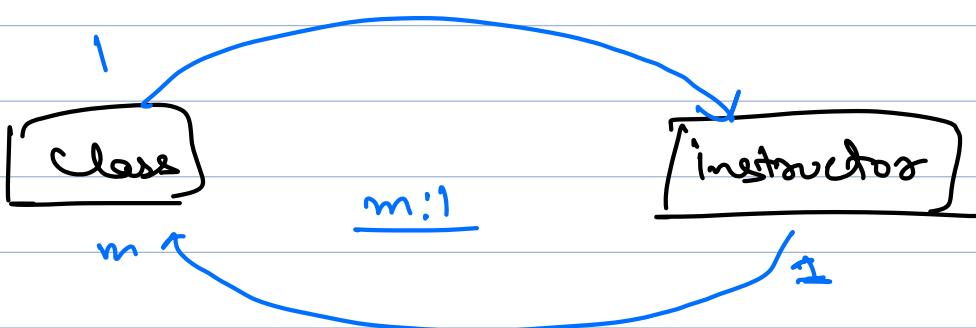
1 : m



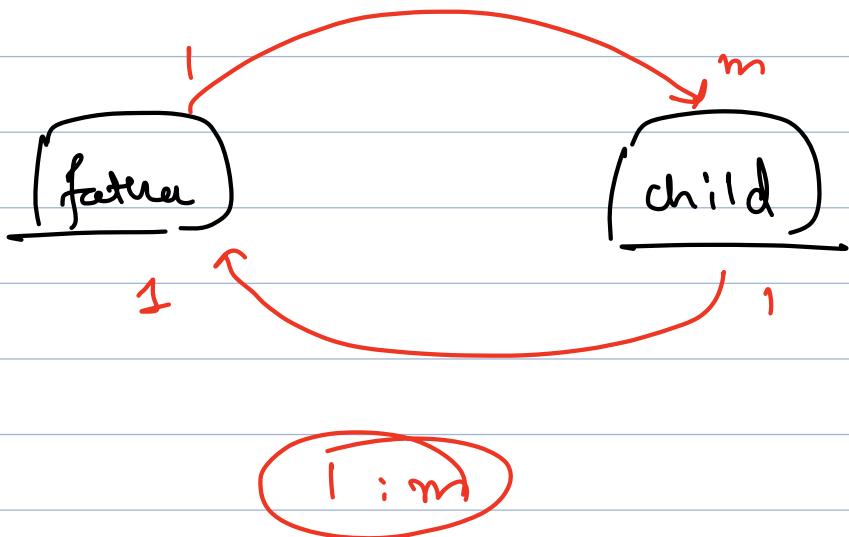
1 : 1



m : 1



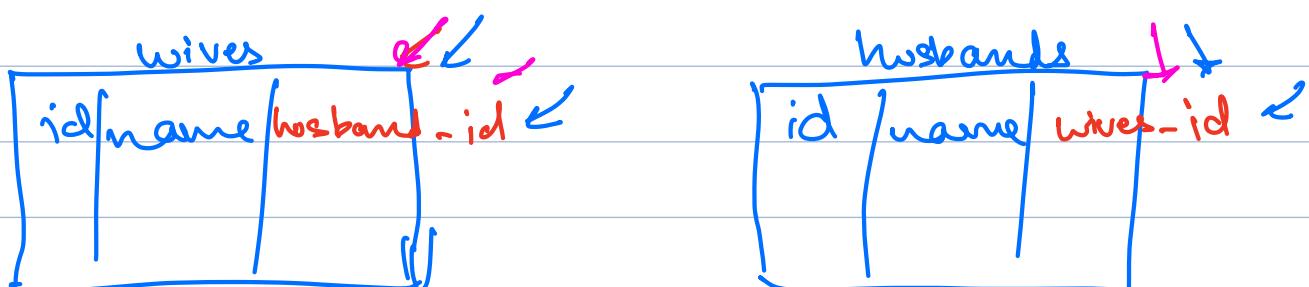
m : 1



Break → 08:32 AM.

How to represent diff cardinalities.

1:1



① husband_id in wives table

OK

② wives_id in husband table

Solve → id of any one side to another side

I can also put on both tables but the problem \rightarrow redundancy
 \downarrow

same thing stored
at two diff places
 \downarrow

Storage is wasted
 \downarrow

lead to update anomaly.

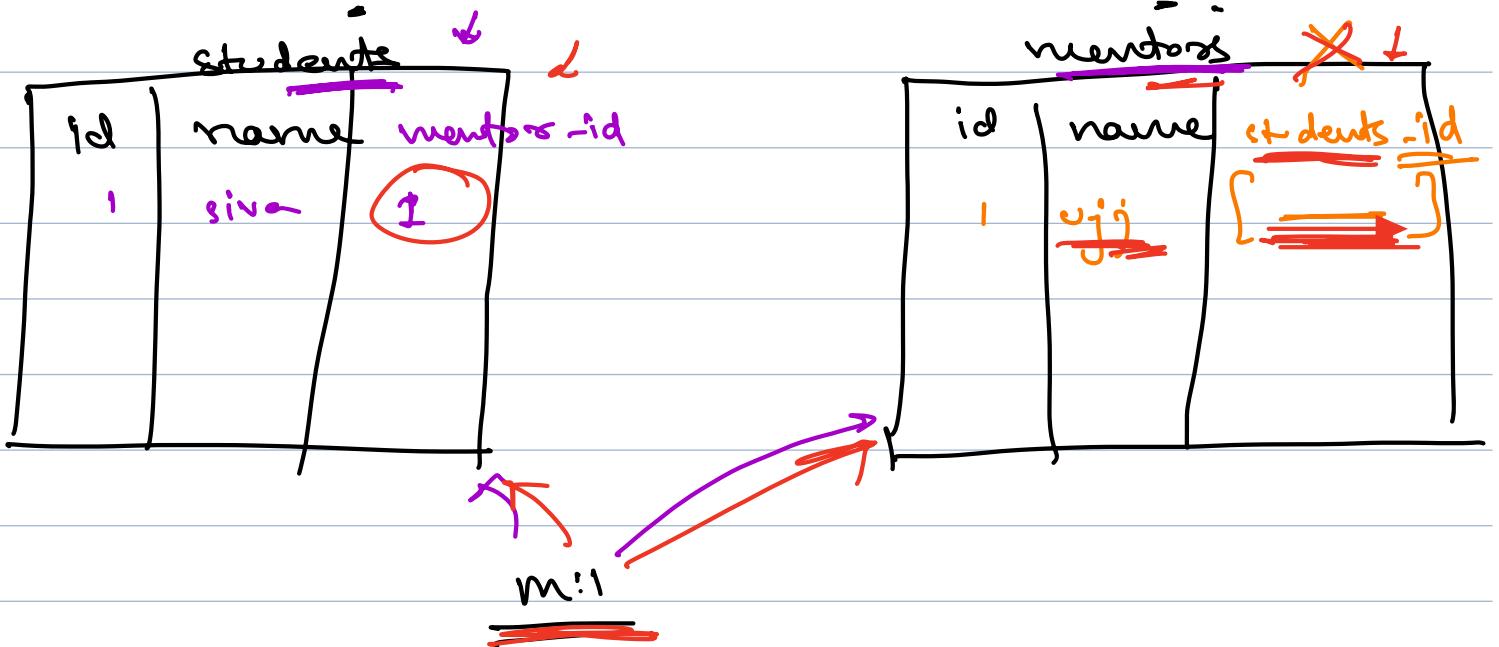
\downarrow
diff data at diff places.
 \downarrow

inconsistency.

1:m
or
m:1

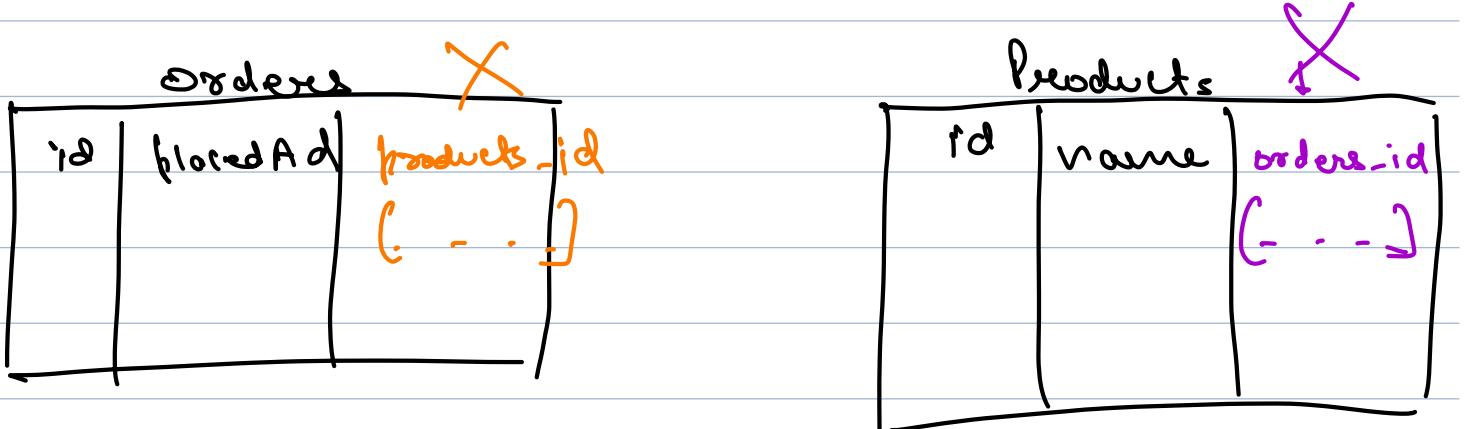
{ A:B \rightarrow 1:m }
B:A \rightarrow m:1 }

If we learn one of them, we will learn the other as well.



Solv Put id of 1 side on m side.

m:m



m:m

Diagram of a mapping table for the many-to-many relationship between Orders and Products.

order-products table:

order-id	product-id
1	1
1	2
1	3
2	2
2	3

Mapping table
or
lookup table.

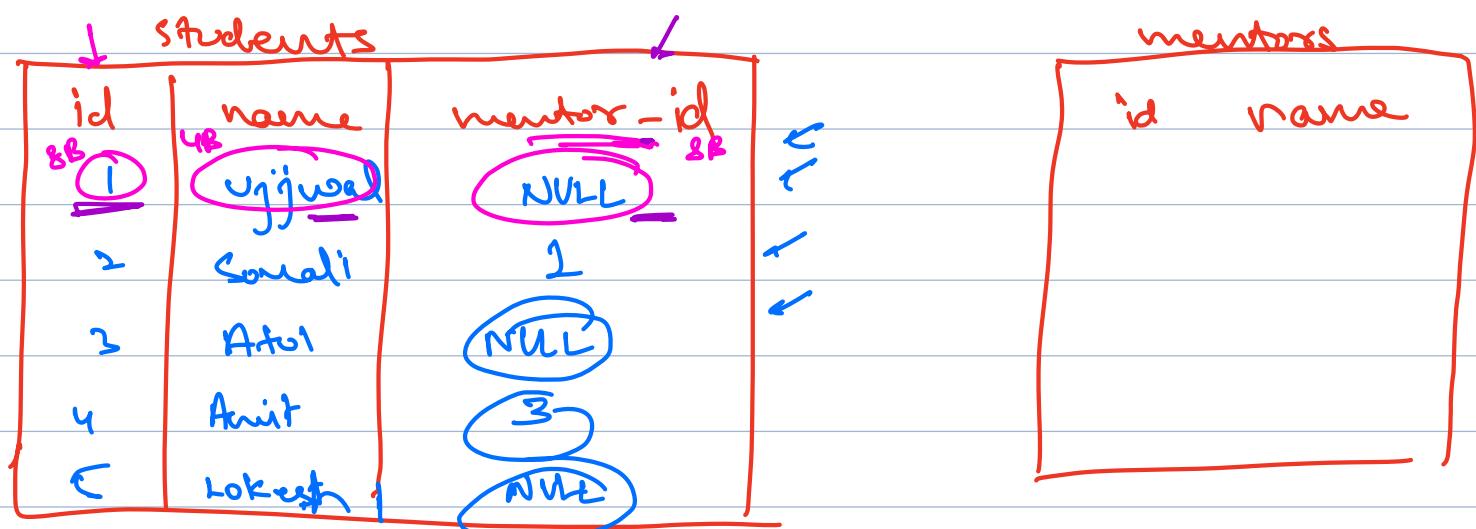
$1:1 \rightarrow$ id of one side on other side

$1:m$ or $m:1 \rightarrow$ id of 1 side on m side

$m:n \rightarrow$ mapping table

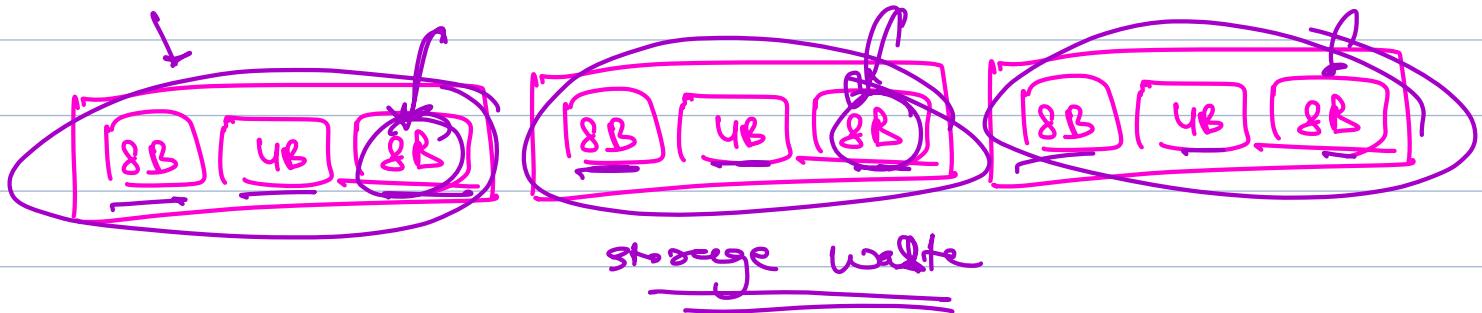
Nuances when representing tables

$1:m$ or $m:1$



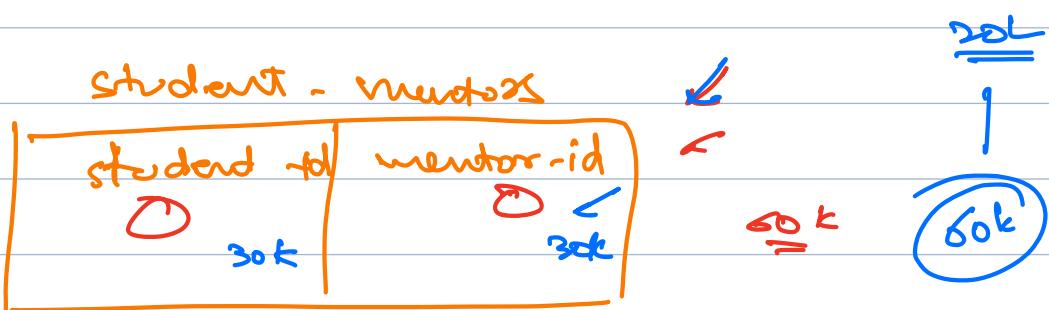
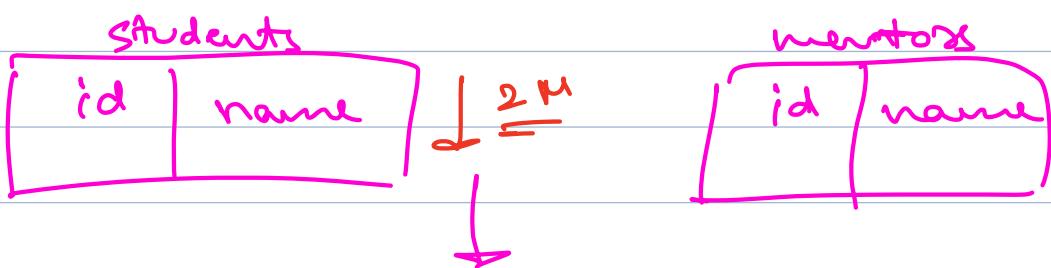
↓
20L
= 30k
→ 1.97M rows.

for MySQL each row uses same space even though the value is NULL it will consume & bytes



sparse relation → when a lot of entities are not part of the relation

Soln. create a new table.



Pro → No storage wasted

Cons → Need joins (impact performance)

1:1

girls		
id	name	husband-id
		NULL ,
		NULL ;

boys		
id	name	wife-id
		,
		;

' This is also
sparse relation

Soln → New table

marriages	
husband-id	wife-id

Cardinality	Normal	Sparse	Rel "has" attributes
1:1	id of 1 side on other side		
1:m m:1	id of 1 side on m side	Mapping Table	Mapping Table
m:m			

⇒ There can be use cases where we have to store information about relation as well.

husband		wife	
id	name	wife-id	id

↓

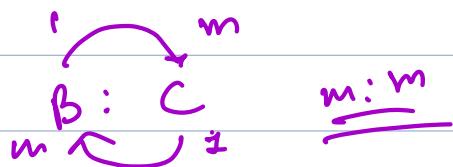
Purpose of the husband table
is defeated

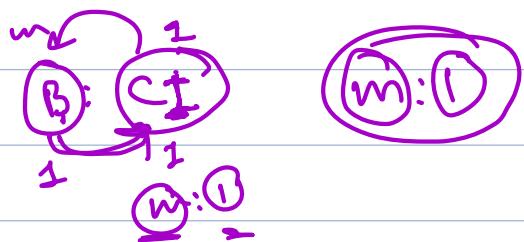
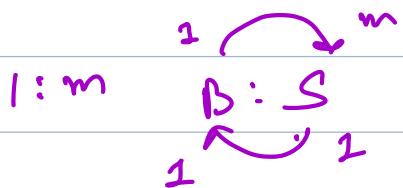


Mapping

Marriages

husband-id	wife-id	marriage date	meeting date





Requirements

- ① Scaler will have multiple batches About each batch we have to store their name, start month, current instructor.
- ② Each batch of Scaler will have multiple students
- ③ Each batch has multiple classes.
- ④ For each class we have to store the name of the class, date and time of the class, instructor of the class.
- ⑤ For every student we store their names, grad year, university name, email, phone number.
- ⑥ Every student has a buddy who is also a student.
- ⑦ A student may move from one batch to another.
- ⑧ For each batch a student goes to we have to store the start date of that batch.
- ⑨ Every student also has a mentor. For every mentor we store their name and current company
- ⑩ We have to store information about all mentor sessions (time, duration, student, mentor, study session)
- ⑪ For every batch we have to store if its an Academy batch or DEML batch.

batches

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

instructors

instructor-id	name	email	avg-rating
---------------	------	-------	------------

students

student-id	name	grad-year	univ-name	email	phone-no	batch-id
------------	------	-----------	-----------	-------	----------	----------

classes

class-id	name	scheduled-time
----------	------	----------------

mentors

mentor-id	name	current-company
-----------	------	-----------------

mentor-sessions

mentor-session-id	time	duration	stud-sat-ing
-------------------	------	----------	--------------

batch-classes

batch-id	class-id
----------	----------

H/w \Rightarrow represent rel " after " at 4