Schema Design - I

Agenda

Till now we had a db given & we wrote queries on it.

Today we see what goes behind creating a database.

How to frame a database. - tables, PK, FK etc.

I
1. What is schema design
2. How to approach schema design. eg tables, PK etc.
3. Conventions → naming etc
4. Representing cardinalities → 1:1, 1:m, etc.

II
5. Deciding Primary Keys → when PK has multiple cols
6. Sparse table
7. Relations with attributes

What is schema Design → Solving how should something be solved CONSIDERING THE CONSTRAINTS

↓

The structure of a database
↳ what all tables
↳ what all columns in every table
↳ Primary key
↳ foreign "
↳ Index

In other words,
pictorial rep of how the dB is structured.

What is design → basic blueprint everyone/ you will follow → LLD / HLD / DB design.

↓

eg house design

What is LLD like, what is HLD like
[constraint matter]

Design - Pictorial representation considering all the constraints

→ Before tables etc are created, they create a design document
→ schemas → class diagram → arch. diag

How to approach Schema design

We will understand this by solving a case study.

Scaler.

All entities about which I need to store data?

Requirements:

(1) Scaler will have multiple batches, About each batch we have to store their name, start month, current instructor. if only name we don't need table but if other values then a table

(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, student rating)

(11) For every batch we have to store if its an Academy batch or DSML batch.

STEPS.

What is the most crutial/central part of a dB.

↳ Tables.

(1) Create the tables.

How to identify tables.

Person, place, animal, thing

(1) Find all 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 ahead

Convention about names

(1) Name of the table should be plural.

Because it is storing multiple things

'mentor_sessions' (called snake case)

MentorSessions
mentorSessions

(2) Name of the colms is a singular - snake case)

★ Representation of a schema doesn't matter.

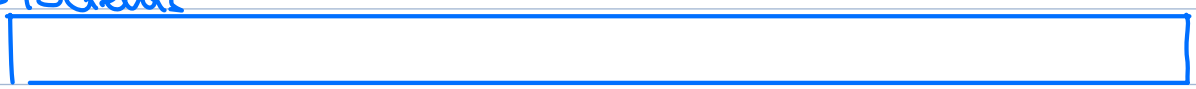
what matter is whether you have all tables need to satisfy req.

Schema.

batches

instructors

students



classes

mentors

mentor_sessions

② Step 2 → in the tables that you just created, add the id(primary key), all attributes about that entity.

→ Dont care about FK.

→ Dont care about relation to other entities.

Expectations with PK.

① It should rarely change

(why ① update index
② re-sort data on disc)

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

(why - ① sort
② lower size)

⇒ Have a seperate integer / big integer (10^18) column called 'id' as a primary key.

Convention wrt naming key column:

① call it 'id'

② call it {tablename}_id / {entity}_id eg batch_id

batches

batch_id	name	start_month

instructors

instructor_id	name	email	avg_rating

students

student_id	name	email	phone_no	grad_year	univ_name

classes

class_id	name	scheduled_time

mentors

mentor_id	name	current_company

mentor_sessions

mentor_session_id	time	duration	stud_rating	mentor

What is missed ? Relations

STEP 3 → Represent relations

How to represent relations.

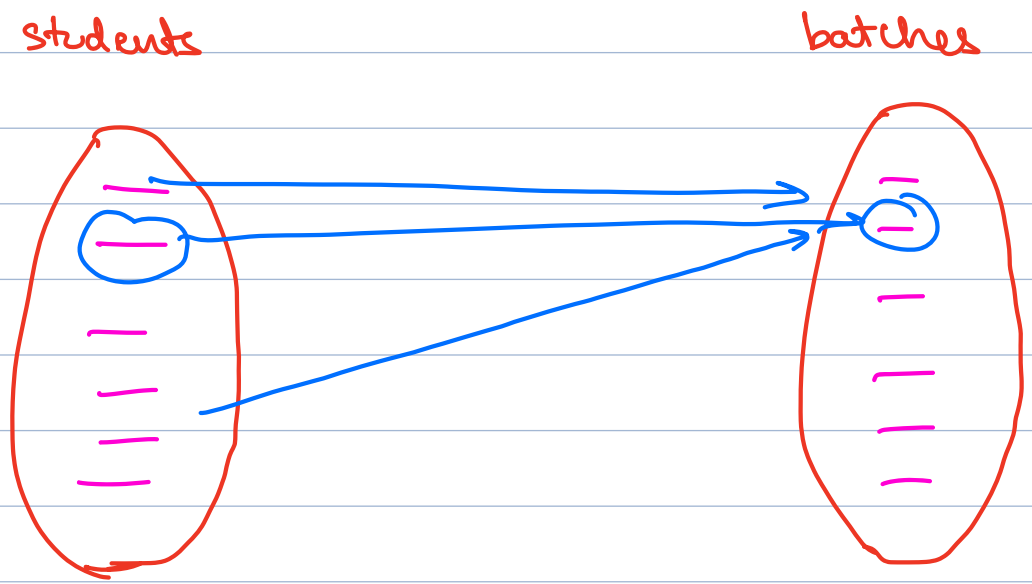
Cardinality → How many of one is related to how many of other.

eg. student ⟹ batch.

Cardinality : How many students are related to how many batches and vice versa.

Possibilities

- 1 : 1 → one stud has one batch & one batch has one stud
- 1 : m → one student belongs to m batches & 1 batch has 1 stud
- m : 1 → one stud to one batch & one batch to multi stud.
- m : m → one stud to many batch & one batch to many stud.



(1) : an entity can be associated to at max 1 instance
⇒ [0,1]

(m) : an entity can be associated (even if sometimes) to more than one instance
[0, 1, 2 ..... ∞]

How to represent cardinalities in relations.

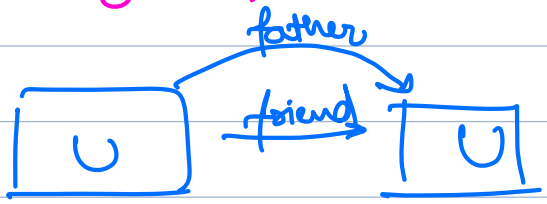
(1) How to find cardinality in a rel"

What is cardinality of user to user in facebook.

what is card" b/w

depends on their relation

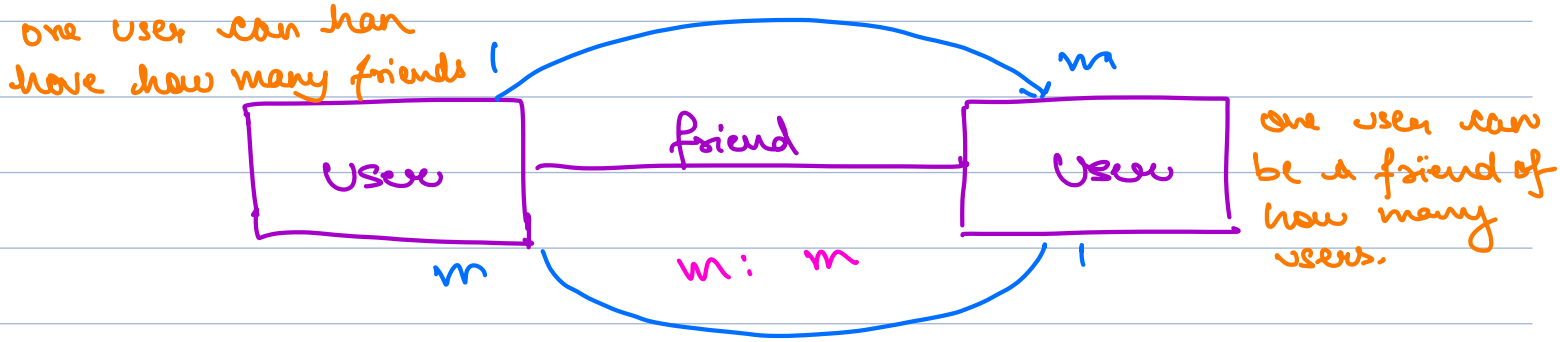
(1) b/w 2 entities there can be multiple type of rel". Each type of rel" can have a diff cardinality.



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

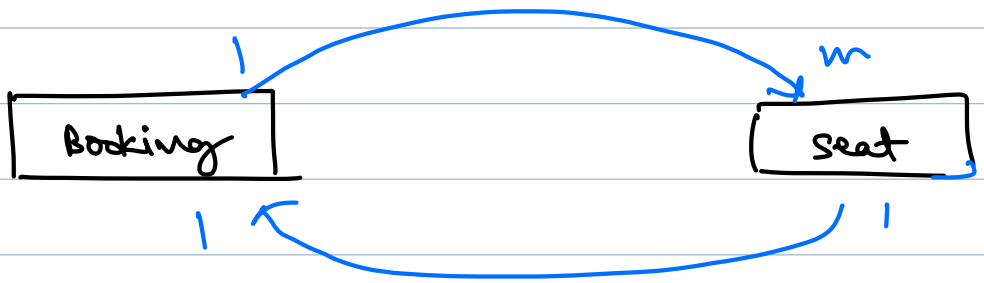
find cardinality of.

SOLN : 2 step approach.

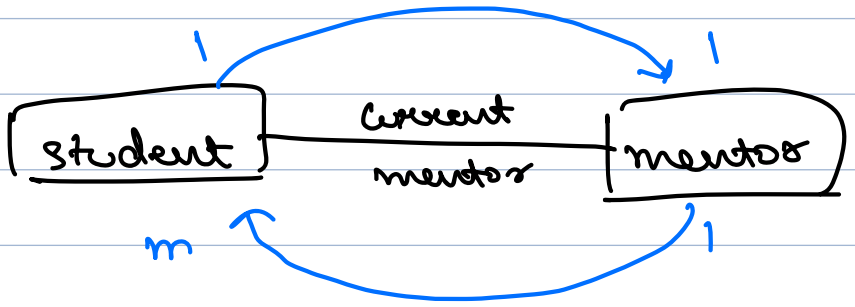


① go left to right putting 1 on left side
② go right to left putting 1 on right side.
③ if there is m on any side, put m on that side in final ans.

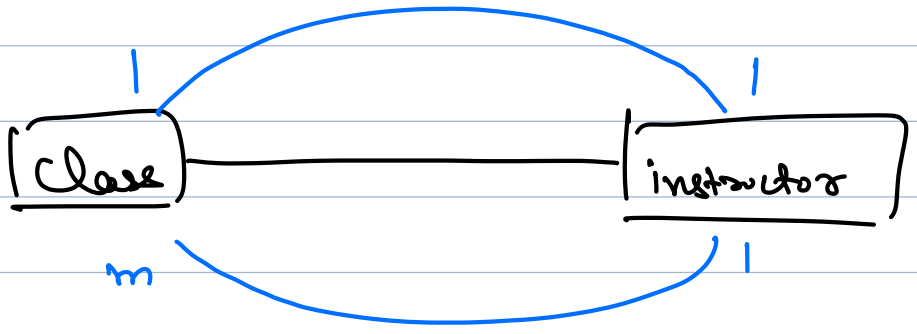
In BMS.



1 : m



m : 1

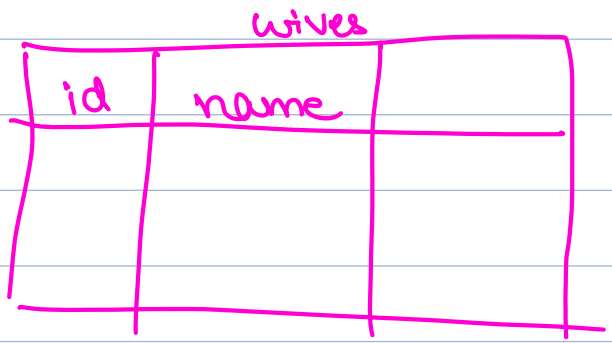


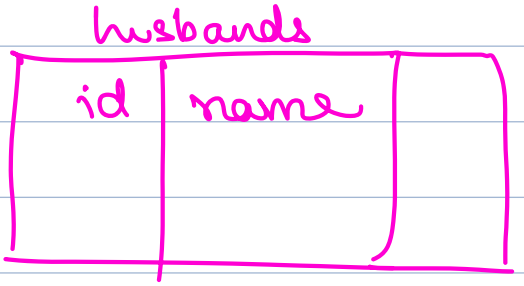
m : 1

[break]

How to represent diff cardinalities

1 : 1





① husband_id on wives table

OR

② wives_id on husband table.

Soln Id of any one side to another side.

how to decide → performance

→ I can also put on both sides but the problem would be redundancy.

↓

Same thing stored at two different places.

↓

Can cause update anomaly

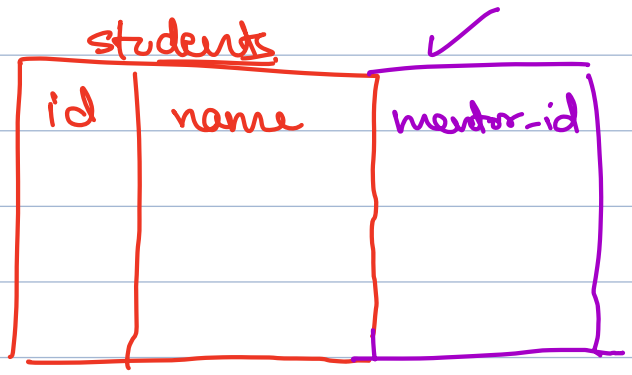
↳ diff data at diff places inconsistency

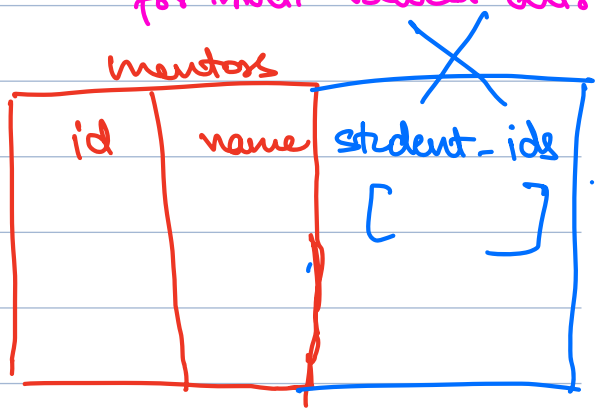
1 : m AND m : 1

{ A : B → 1 : m
  B : A → m : 1 }

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

SQL is not optimised for multi-valued attr.

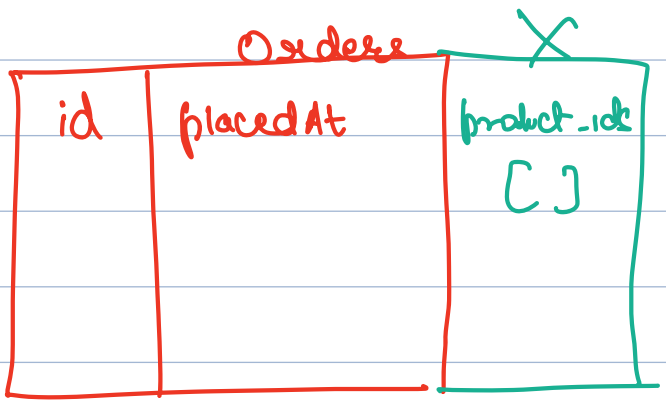


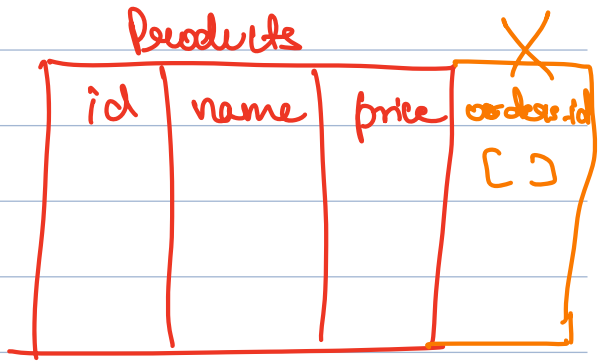


One mentor can have many students

Soln id of 1 side on m side.

m:m





If husband not good with wife's parents & wife not good with husband's parents They live in a new house.

order_products → Mapping Table

or

lookup table

order_id	product_id
1	1
1	2
2	2
2	3
3	6

1:1 → id of one side on other side

1:m or m:1 → id of one side on m side

m:m → mapping table

Nuances when representing tables.

1 : m or m : 1

students

id (8B)	name (4B)	mentor_id (8B)
1	Ujjwal	NULL
2	Amit	2
3	Pramod	2
4	Alok	NULL
5	Shiv	3

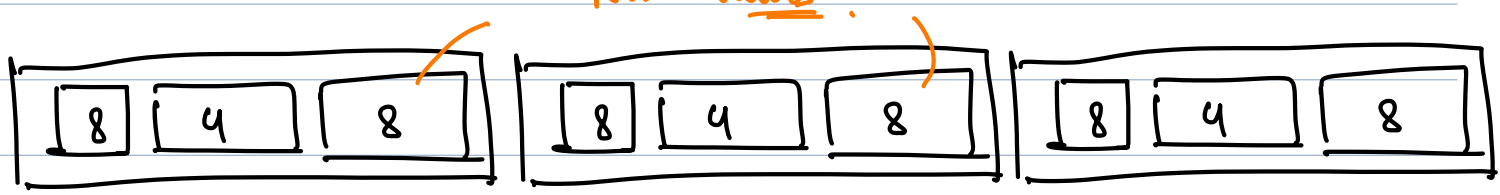
mentors

id	name

2M - 30K
= 1.97M nulls.

20,00,000 ⇒ 30,000
rows

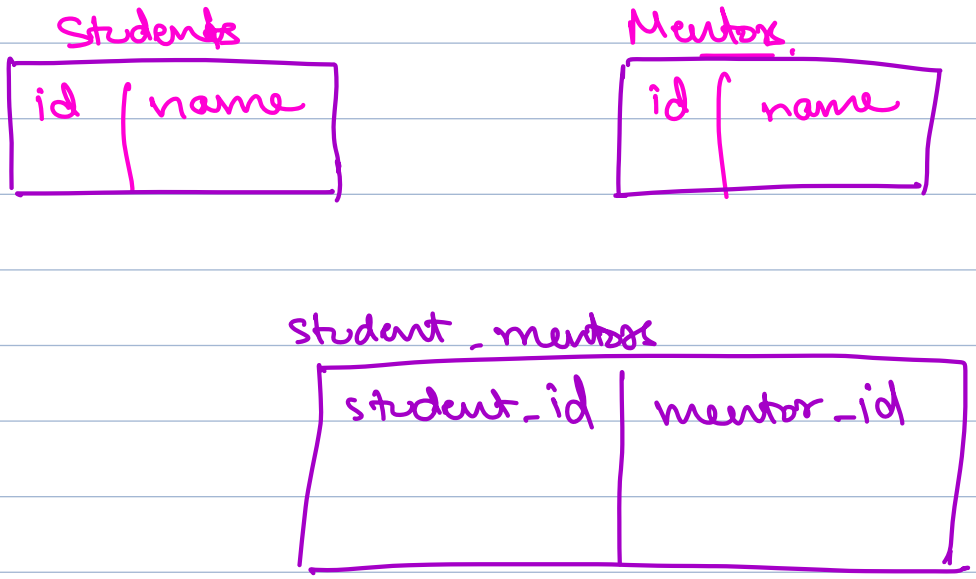
for MySQL each row / column uses same space.
even though the value is null it will consume 8 bytes.



⇒ waste of storage

Sparse relation → where a lot of entities are not part of the relation.

Soln → create a new table



Pros → saved storage

cons → Need joins (affect performance)

1:1

this is also sparse

Soln → New table

marriages

husband_id	wife_id

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

⇒ there can be use cases where we have to store information about a rel" as well.

husbands

id	name	wife_id	marriage date	date of meeting

wives

id	name

Purpose of table defeated.

Diluting purpose of husband's table

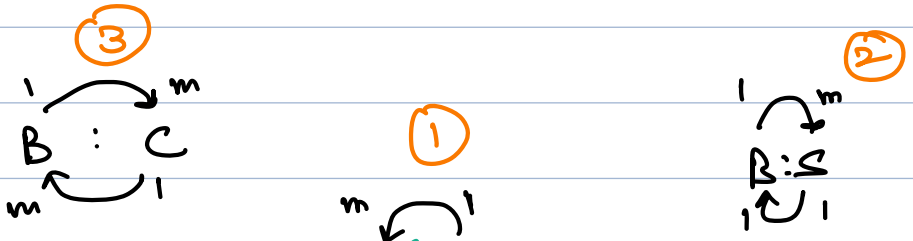
SRP → Single Responsibility principle

↓

Responsibility of everything must be defined.

Marriages

hus_id	wife_id	marriage date	meeting date



Scaler.

Requirements

① Scaler will have multiple batches, About each batch we have to store their name, start month, current instructor. if only name we dont need table but if other values then a table

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, student-rating)

⑪ For every batch we have to store if its an Academy batch or DSML batch.

batches

batch-id	name	start-month	current_inst-id

instructors

instructor-id	name	email	avg-rating

students

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

classes

class-id	name	scheduled-time

mentors

mentor-id	name	current-company

mentor_sessions

mentor_session_id	time	duration	stud_rating	mentor

batch_classes

batch_id	class_id

H/w Represent rel" after Pt 4