Today's Agenda :-

1) What is Schema Design
2) How to approach Schema Design
3) Conventions
4) Represent Cardinalities
5) Deciding Primary Keys
6) Sparse Relations.

Schema Design :-

- Structure of the database
  - what all tables
  - what all col^n in each table
  - Primary keys
  - foreign keys
  - Indexing
- Pictorial Representation of the database considering all constraints.

How to approach Schema Design?

1) Scaler will have multiple batches. About each batch, store their name, Start month and current instructor.

1	1
Batches	Instructor
m	1

m : 1

2) Each batch of Scaler will have multiple Students.

1	m
B	S
1	1

1 : m

3) Each batch has multiple classes.

1	m
B	C
m	1

: m : m

4) For each class, store the name of the class, date & time of class, instructor of class.

1	1
C	I
m	1

: m : 1

5) For every Student, store their name, 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.

1	1
S	M
m	1

: m : 1

9) Every student also has a mentor. For every mentor, store their name and current company name.

10) We have to store info about all mentor sessions

store the time, duration, student, mentor,

stud_rating, mentor_rating.

mentor_session student

mentor

11) For every batch we have to store info if it is Academy batch or DSML batch.

Steps :-

1) Create the tables

How to identify the tables ?

a) Just find all nouns that are present in the requirements

b) for each noun, ask if you want to store data about entity in your database

c) If yes :- Create a table

if no :- don't create.

Snake case

Conventions :-

1) Name of table should be plural because it is storing multiple things.

'mentor_Sessions'

MentorSessions ✓

mentorSessions ✓

2) Name of a column is singular.

batches

batch_id	Name	Start_month	instructor_id

instructors

instructor_id	Name	phone	email	avg_rating

Students

Student_id	name	email	phone	gradYear	Univ_Name	batch_id	mentor_id

classes

class_id	name	Schedule	instructor_id

mentors

mentor_id	name	current_company

mentor_Sessions

mentor_session_id	time	duration	St_rating	men_rating	St_id	mentor_id

batches classes

batch_id	class_id

Step 2:-

In all the tables you just created :-

1) Add the id (primary key)

2) Add all other attributes about the entity.

* don't worry about fk
* don't worry about relations with other enti

y

Expectations with pk :-

1) It should rarely change.

Why ? → update the index

2) It should ideally be a data type easy to work with & of small size.

Why ? → sort

↳ Having a separate integer col as pk

Add column 'id' as a primary key

i) Call it 'id'

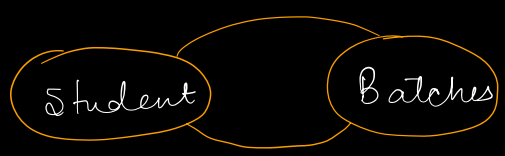
ii) call it '{table_name}_id'

Convention

How to represent relations ?

Cardinality :- Whenever 2 entities are related to each other :-

Ask a ques :- how many of one are related to how many other.



How many students are related to how many batches & vice versa ?

Possibilities :- 1 : 1

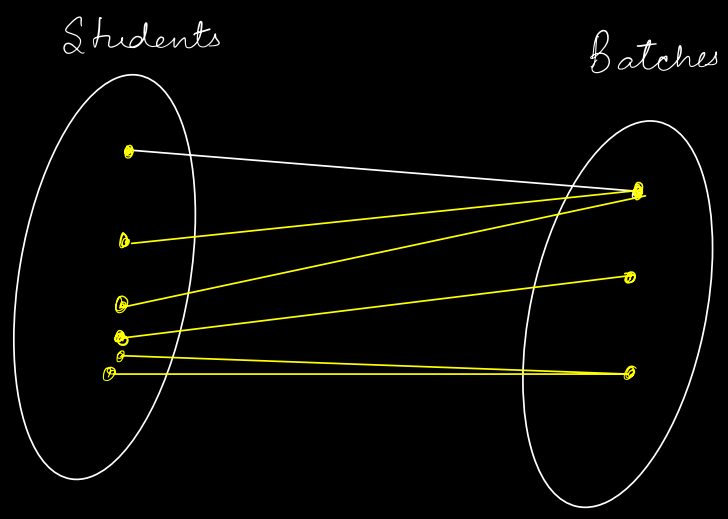
1 : m

m : 1

m : m

1 : an entity can be associated to atmax 1 instance [0, 1]

m : an entity can be associated to more than 1 instance [0, 1, 2, 3, . . .]
mostly



L — R

R — L

1 | 1

m | 1

m | 1

$\boxed{m : 1}$ ✓

1 | 1

Boy | Girl

1 | 1

Husband - wife

1 : 1

1 | m

Boy | Girl

m | 1

Siblings

m : m

How to represent Cardinalities :-

Between 2 entities there can be multiple types of relation. Each type of relation can have different cardinality.

a) find 2 entities b/w which you have to find cardinality

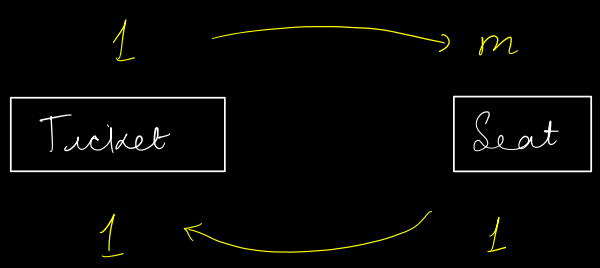
b) find what relation b/w them on basis of what you need to find cardinality

2 - Step approach (trick)

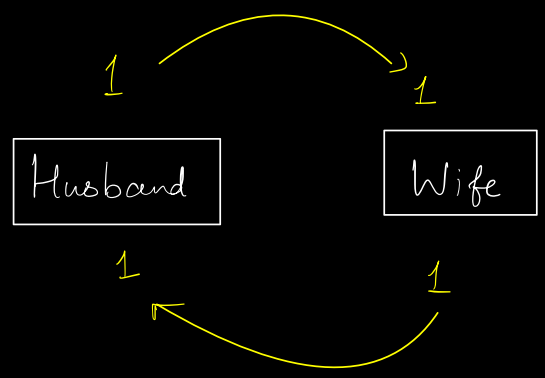
a) go left to right putting 1 on left side

b) go right to left putting 1 on right side

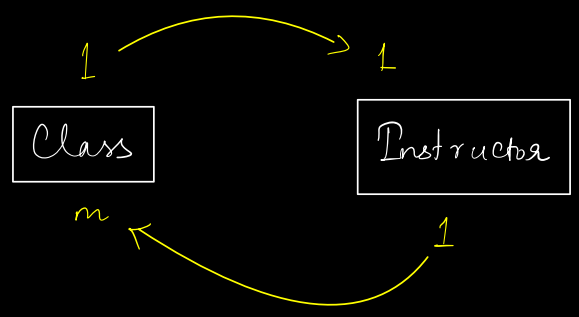
c) if there is m on any side, put m on that side in final answer.



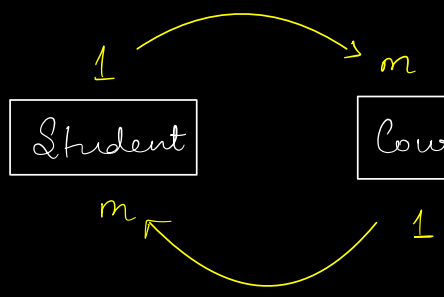
1 : m



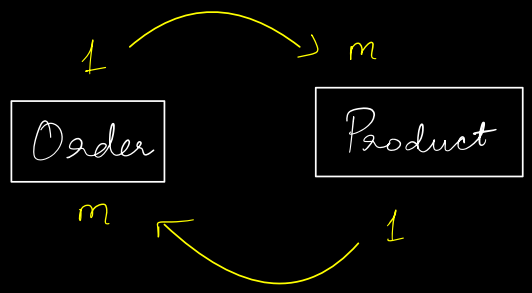
1 : 1



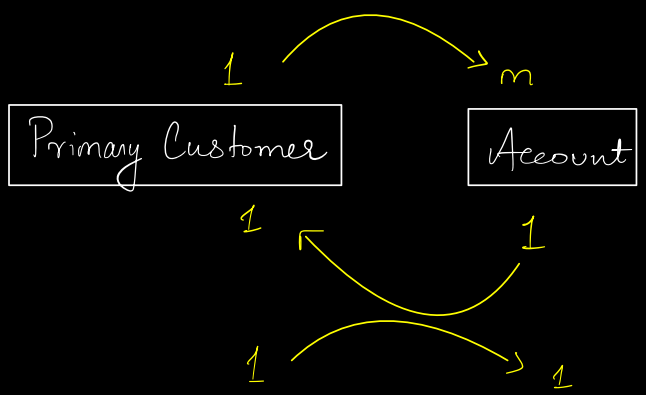
m : 1


1
m
Student
Course
m
1

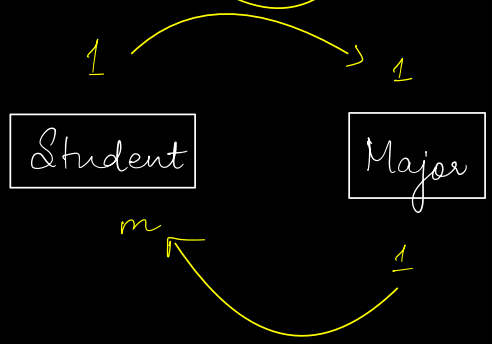
m : m


1
m
Order
Product
m
1

m : m


1
m
Primary Customer
Account
1
1

1 : m


1
1
Student
Major
m
1

: m : 1

How to represent relations in tables ?

break of 6 min

① $\boxed{1 : 1}$ → id of any 1 side on the other side

Wives

id	Name	husband_id
3		2
5		2

OR

Husbands

id	Name	wife_id
2		~~3~~ 5

Add a husband_id in wives table

OR

Add a wife_id in husband table

I can put on both side :-

* Space is wasted
* same data at 2 diff places
* update anamoly

② 1 : m or m : 1

↳ id of 1 side on m side ✓

1 Students m

1 Mentors 1 : m : 1

Students

id	Name	mentor_id
2		1
3		1
4		1

Mentors

id	Name	~~list<Students>~~
1		{2, 3, 4, }

3) $\boxed{m : m}$

Order Products : $\boxed{m : m}$ ✓

Orders

id	Name	~~list<Prod>~~

Products

id	Name	~~list<Ord>~~

<u>Mapping table | look up table</u>

create table order_products
order_id
prod_id }

Order_products

order_id	prod_id
1	2
2	3
1	3
2	4
.	.

→ {order_id, prod_id}

1 1
User Mentor $\boxed{m : 1}$
m 1

User

id	Name	..	men_id

Mentor

id	Name

2 M Users at scalers : Lot of NULL values.

↳ 30 K active learners

↳ 2M - 30K

19.7 M user doesn't have a mentor.

Sparse Relations :- where lot of instances are not part of relation

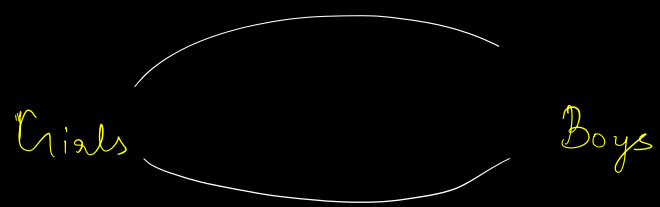
↳ Better to a new table

Users_Mentors → 30K rows

user_id	mentor_id
✓	✓
✓	⋮
⋮	

Pro : Saved space

Con : Need joins



hus_id

OR

wife_id

New table

New table for the relation

hus_id	wife_id	marriage_date	anniversary
✓	✓	✓	✓

Representing Cardinalities in tables :-

Cardinality	Normal	Sparse Relation	Rel^n have attributes
1 : 1	Any 1 id on Other side	Mapping table	Mapping table
1 : m m : 1	id of 1 side on m side		
m : m	Mapping table		

H/W Please try to revise entire schema design & cardinalities concept we learned today

Next class :- 1) Complete the schema design for Scaler } 15

2) Case Study for Netflix. }

SQL contest

27th July 2024

7 : AM

Window 7 days

Stud

id
1

Mentors

id
2

Students_mentors

st_id	men_id
1	2