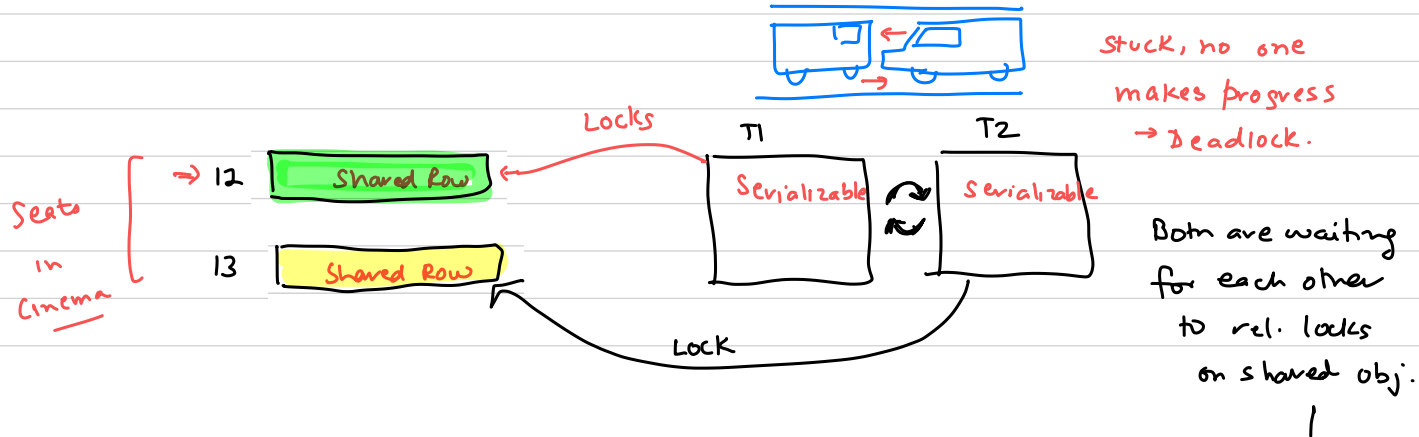# Agenda

→ Deadlocks

_____

→ Schema Design
   – Process / how to approach SP
   – 2 Case study  (Scaler, Netflix)

## Deadlocks



Stuck, no one
makes progress
→ Deadlock.

Locks          T1            T2

→ 12   Shared Row  ←          Serializable    Serializable

Seats
in
Cinema

13   Shared Row

LOCK

Both are waiting
for each other
to rel. locks
on shared obj.

**T1** → user1    ↓ Deadlock

**T2** → user2

{ wait

```
SELECT FROM seats
WHERE id = 12
FOR UPDATE;
```

```
SELECT FROM seats
WHERE id = 13
FOR UPDATE;
```

Rollback

wait

```
SELECT FROM seats
WHERE id = 13
FOR UPDATE;
```

```
SELECT FROM seats
WHERE id = 12
FOR UPDATE;
```

T1 finished

Take Locks in a fixed order
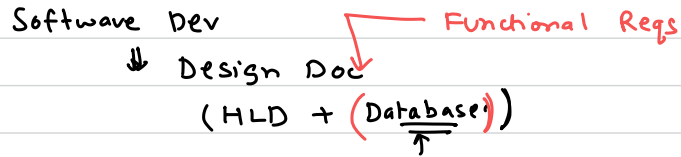
T1 → 12 , 13  finishes

T2 → 12 , 13  finishes  (take locks in asc order on resources)

• Deadlocks can't be avoided always.

↳ Roll back one of transactions when deadlock is detected.

## Schema Design — Designing the structure of DB.

{
→ Tables
→ columns
→ PK,
→ FK
→ Indexes, Views etc.
→ Relationship b/w Tables.

Software Dev
⇓ Design Doc
( HLD + ( Database ) )
↑

Functional Reqs

ER Diagram

## 4 step Process

## Case Study : Scaler

The requirements are as follows:

Scaler will have multiple batches.
For each batch, we need to store the name, start month and current instructor.
Each batch of Scaler will have multiple students.
Each batch has multiple classes.
For each class, store the name, date and time, instructor of the class.
For every student, we store their name, graduation year, University name, email, phone number.
Every student has a buddy, who is also a student.
A student may move from one batch to another.
For each batch a student moves to, the date of starting is stored.
Every student has a mentor.
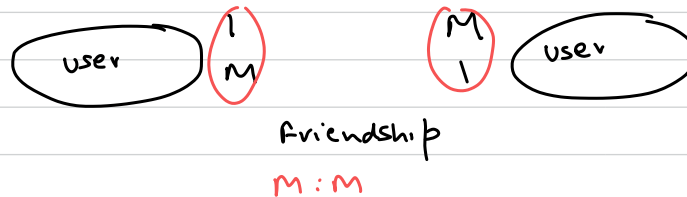For every mentor, we store their name and current company name.
Store information about all mentor sessions (time, duration, student, mentor, student rating, mentor rating).
For every batch, store if it is an Academy-batch or a DSML-batch.

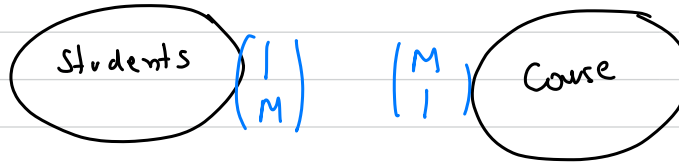Cardinality :     Relation: students enroll to batch.

1:1

M:1

1:M

M:M

Students  —  1 M max  —  1 1 max  —  Batches

$M : 1$
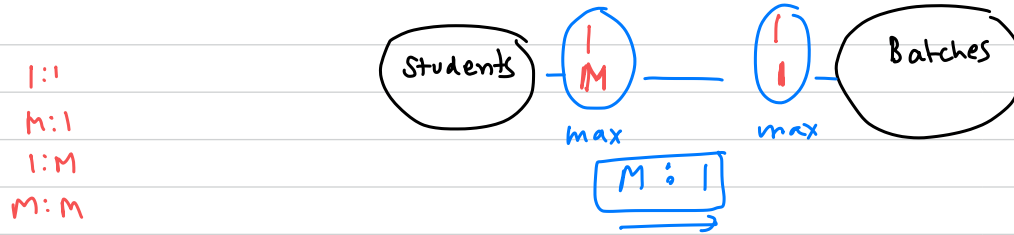
TRICKS:    Think for 1 student =  __1__ batch
           Think for 1 batch =  __M__ students

user  1 M  M 1  user

friendship

M : M

( orders )    1          M    ( Products )
              M          1

M:M

= #57

Cam
fruit
:

( Students ) $\begin{pmatrix} 1 \\ M \end{pmatrix}$  $\begin{pmatrix} M \\ 1 \end{pmatrix}$  ( Course )

M: M

( Customer ) $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $\begin{pmatrix} M \\ 1 \end{pmatrix}$ ( Bank AC )

1: M

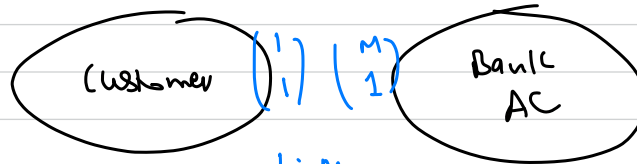Student $\left( \begin{matrix} 1 \\ M \end{matrix} \right.$ $\left. \begin{matrix} 1 \\ 1 \end{matrix} \right)$ Major
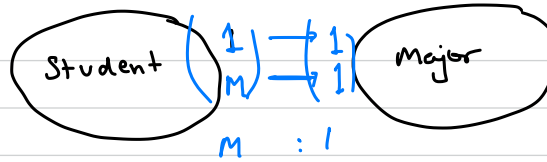
M : 1

**In an educational institution, a student opts for a major subject. This subject might be the choice of several students, but a student cannot major in more than one subject. How would you describe the cardinality between** **Student** **and** **Major** **?**

25 users have participated

| A | One-to-One | 16% |
|---|---|---|
| B | One-to-Many | 32% |
| C | Many-to-One | 48% |
| D | Many-to-Many | 4% |

## 1:1 Relationship

Husb and    wife Id       wife

| 1 | A |   |   |
|---|---|---|---|
| 2 | B |   | 4 |
| 3 | C |   |   |
| 4 | D |   |   |
| 5 | E |   | 1 |

wife:

| 1 | X |
| 2 | Z |
| 3 | W |
| 4 | O |
| 5 | P |
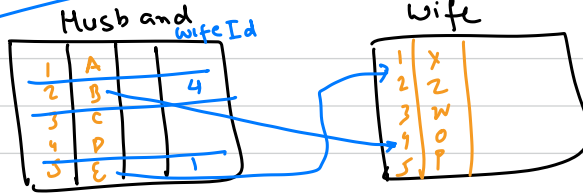
→ id of wife can go in husband table
   OR vice-versa.

→ id of anyone side on other side.

1:M

or

M:1

Students — M — 1 — Batches
         max      max

M : 1

id of '1' side
will go
on 'M'
Side.

| Student | batchid |
|---------|---------|
| Kavita | 2 |
| Rahul | 1 |
| . | . |
| . | . |

M:M

④

actors $\left(\begin{smallmatrix} 1 \\ M \end{smallmatrix}\right)$ -- $\left(\begin{smallmatrix} M \\ 1 \end{smallmatrix}\right)$ Film

M . M

create
a
separate

Mapping Table

actor ⌠

| | |
|---|---|
| 1 Shavku | |
| 2 Kivti | |

film ⌠ actors

| | |
|---|---|
| 1 XYZ | (1,2,8) |
| 2 ABC | (1,~~2~~) |
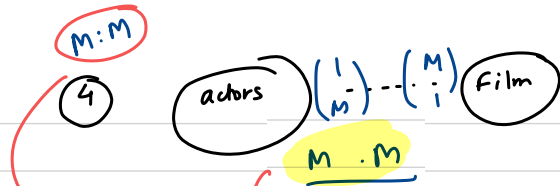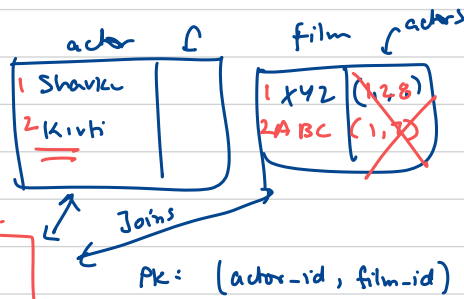
Mapping
tables
don't have
a sep
id for PK

~~id?~~
~~1~~
~~2~~
~~3~~
~~4~~
~~5~~

| actor_id | film. |
|---|---|
| 1 | 1 |
| | 2 |
| 2 | 1 |

| 7 | = |
|---|---|

Joins

PK: (actor_id, film_id)