

Today's Agenda :-

- 1) Revision to Cardinality
- 2) Sparse Relations
- 3) Scaler's Schema Design
- 4) Represent relations / foreign keys
- 5) Netflix Schema Design

St Men sessions

Student_Mentor ✓

<u>St_id</u>	<u>Mentor_id</u>

Pro : Saved space ✓

Con : Need joins

Girls

1

h_id

Girl Boy 1:1
| : |
| : | SR ??

Boys

2

Marriages :

wife_id	h_id	date	budget

also

The mapping table_n gives the opportunity to
store data about the
relation itself.

Cardinality	Normal	Sparse	Rela ⁿ has attributes
<div>1 : 1</div> <div>1 : M OR M : 1</div> <div>✓ m : m</div>	Any 1 id on other side id of 1 side on m side Mapping Table	Mapping Table	Mapping Table.

$$\left. \begin{array}{c} 1 \\ \underline{B} \\ m \end{array} \quad \begin{array}{c} 1 \\ \underline{C:I} \\ 1 \end{array} \right\} \underline{m:1}$$

1) Scaler will have multiple batches. About each

batch, store their name, start month and
current instructor.

$$\begin{array}{c} 1 \\ \underline{B} \\ 1 \end{array} \quad \begin{array}{c} m \\ \underline{I} \\ 1 \end{array} : \underline{1:m}$$

2) Each batch of Scaler will have multiple Students.

3) Each batch has multiple classes.

$$\begin{array}{c} 1 \\ \underline{B} \\ m \end{array} \quad \begin{array}{c} m \\ \underline{C} \\ 1 \end{array} : \underline{m:m}$$

4) For each class, store the name of the class,

date & time of class, instructor of class. $\underline{C} \quad \checkmark$
 \underline{I}

5) For every student, store their name, grad year,
university name, email, phone number. $\frac{1}{m} : \frac{1}{1}$
 $\underline{S} : \underline{S}$

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 joining date.

$$\begin{array}{c} 1 \\ \underline{S} \\ m \end{array} : \begin{array}{c} 1 \\ \underline{M} \\ 1 \end{array} \quad \underline{m:1}$$

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. $\begin{matrix} MS & 2 \\ MS & n \end{matrix} \} 1:1$

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

batches

batch_id	Name	Start month	cur-ins-id	
----------	------	-------------	------------	--

instructors

instructor_id	Name	phone	email	avg-rating
---------------	------	-------	-------	------------

Students

Student-id	Name	email	phone	grad-year	univ_name
------------	------	-------	-------	-----------	-----------

batch-id	buddy-id	mentor-id
----------	----------	-----------

Classes

class-id	Name	Schedule	
----------	------	----------	--

1 Schema Design 2 9PM

Mentor

mentor-id	Name	current-company
-----------	------	-----------------

mentor_sessions

mentor-session-id	time	duration	st-rating	m-rating
-------------------	------	----------	-----------	----------

st-id	mentor-id
-------	-----------

batches_classes $B : C$
(m : m)

batch_id	class_id	instructor_id
----------	----------	---------------

1

3

→ attribute | info

about
class

happening in
a batch

Students_Batches

st_id	b_id	joining_date
-------	------	--------------

Batches Type

id	Name
----	------

1) Create tables

2) Attributes

3) Rep relations

Enums :- Enums are used to represent constants.

a) Use &strings

Batches

id	Name	type
.		"ACADEMY"
		"DSML"
		"DS"

Select * from ...
where b_type = "Aca

→ &string

↓ Redundant

5

"ACADEMY"

"UMANG" X

X

Cons :- i) Spell Mistakes

ii) Redundant data, more space.

iii) &string comparison

Pro :- Readability

b) Integer

Batches

id	Name	b_type	id
		1	✓
		2	✓
		3	✓



+k
Cascade
Not Null

3) Enums | Mapping Table

A handwritten diagram illustrating a 'Batch Type' table. The title 'Batch Type' is circled in yellow. Below it is a table with two columns: 'id' and 'Name'. The first row contains the values '1' and '"Academy"', and the second row contains '2' and '"DSML"'. The entire diagram is enclosed in a large yellow circle.

id	Name
1	"Academy"
2	"DSML"

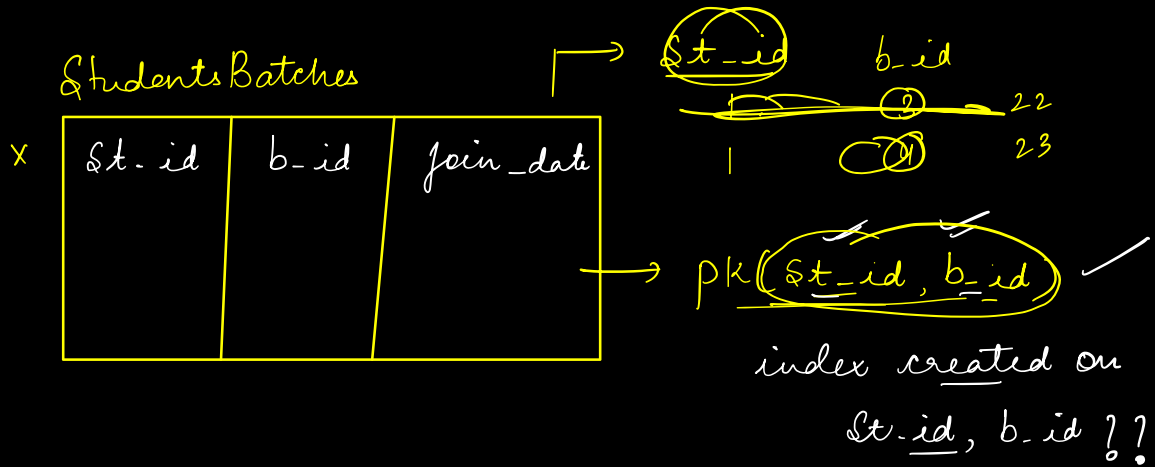
Batches

[illegible]

100 rows

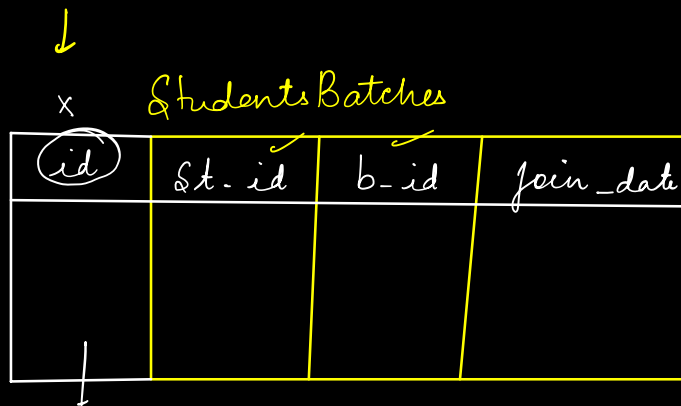
defining all
the constants at 1
place -

How to decide pk of Mapping table?



Add a sep key

Will the queries be faster on these col ?



pk(id) ✓

↳ index on id ?

Advantages of sep key :-

1) Save space because indexing size will be small.

Advantages of not having a sep key :-

1) operations are faster

Always indexing gets governed by use cases.

Along with Schema design problems, you are
also given use cases ?

Please talk about indexes at the end.

$(\overline{A}, \overline{B})$

For whichever columns you want to make queries faster for, on those colⁿ, you should create the index.

Next part : Netflix Schema

Design

7 Min

Steps for Schema Design :-

- 1) Create the tables.
- 2) Add the pk to every table
- 3) Add all attributes
- 4) Represent relations (fk)
- 5) Indexes based on use cases.

Done.