Schema Design - 2

★ If you haven't attended previous class, leave this class.

Agenda

1. Completing schema design of scaler
2. How to decide primary key of a mapping table
3. How to rep foreign keys
4. How to rep indexes

5. schema design of netflix

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.

C : I m:1

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. → self relation

⑦ A student may move from one batch to another.

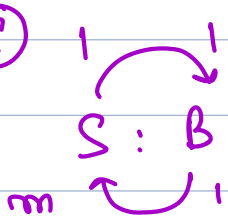
⑧ For each batch a student goes to we have to store the start date of that batch. [Relation with attrs] S—B

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

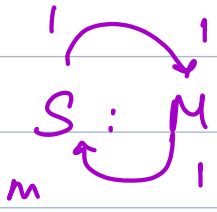
⑩ We have to store information about all mentors sessions (time, duration, student, mentor, stud-rating) [If I had forgotten then also we need this as attrs of relation]

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

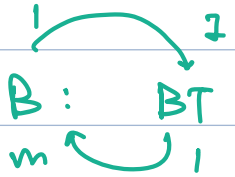
⑥



M:1 → ID of buddy in student



M:1



M:1

batches

batch_id	name	start_month	instructor_id	batch_type_id

instructor

instructor_id	name	email	avg_rating

we can avoid this but beaf imparted when you want current batch.

students

student_id	name	grad_year	univ_name	email	phone_no	batch_id
					buddy_id	mentor_id

classes

class_id	name	scheduled_time	instructor_id

mentors

index

mentor_id	name	current_company

mentor_sessions

mentor_session_id	time	duration	stud_satisfaction	stud_id	mentor_id

batch_classes

batch_id	class_id

→ index on batch_id.

student_batches

student_id	batch_id	wave_date

batch - type

id	name

Batch_Type : enum → have one of the given set of values

→ used to represent all types of a particular things

→ one of a fixed set of values

BatchType
Academy
DSML

```
enum Gender {
    MALE,   0
    FEMALE  1
}
```

How to represent enums.

a) Using strings

batches

id	name	———	type
			"Academy"
			"DSML"
			"Academy"
			"Academy"

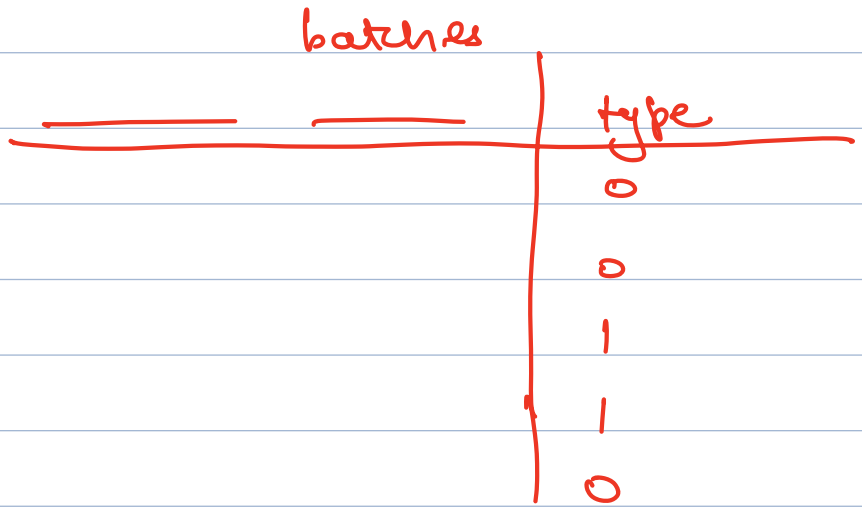
What can be a problem

CONS → space → each value takes a lot of space

→ slow string comparisons.

PROs → Readability
→ no joins is required
→

(b) Using integers



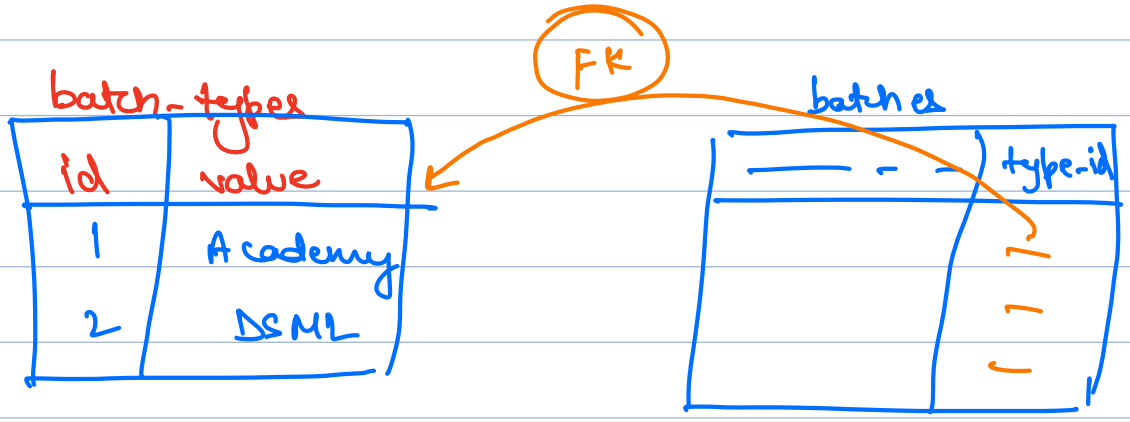
PROs → less space
→ faster to search

CONs → not readable
→ cant add values in BETWEEN
→ cant delete values in BETWEEN
→ what a particular value denotes is not present in the db.

(c) Lookup Table

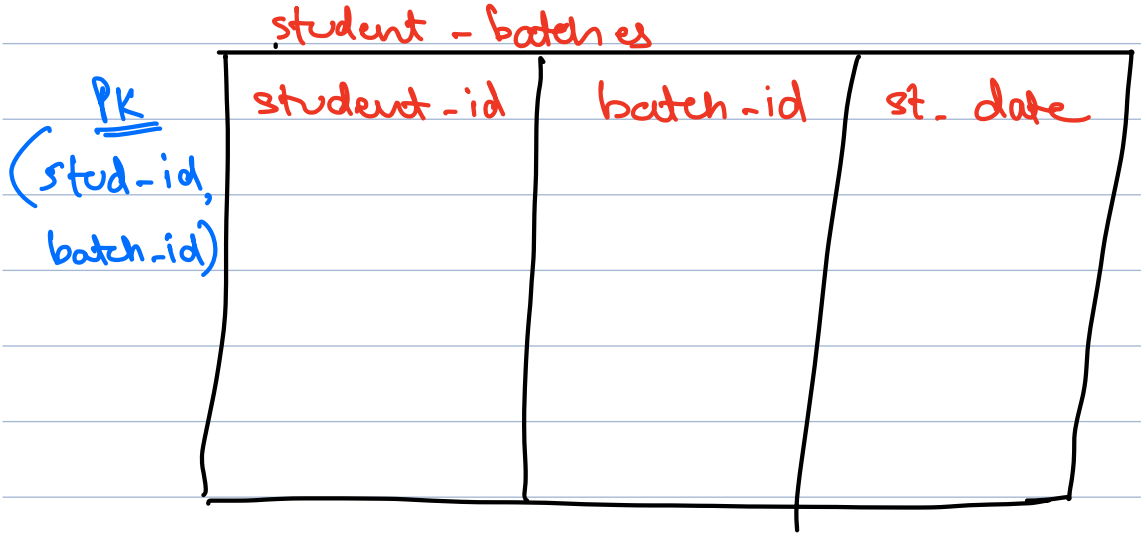
↳ Separate table to store all the types.

↳ each type is stored as a separate row.

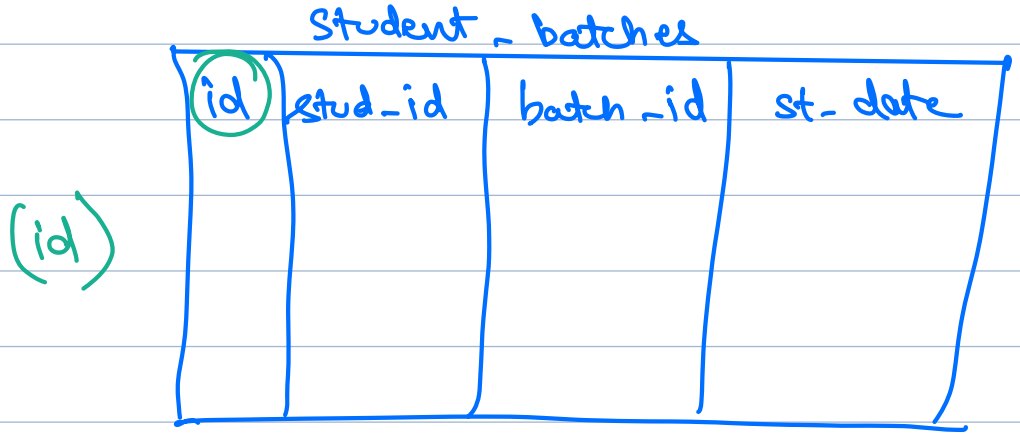


All cons resolved.

How to decide primary key of a mapping table



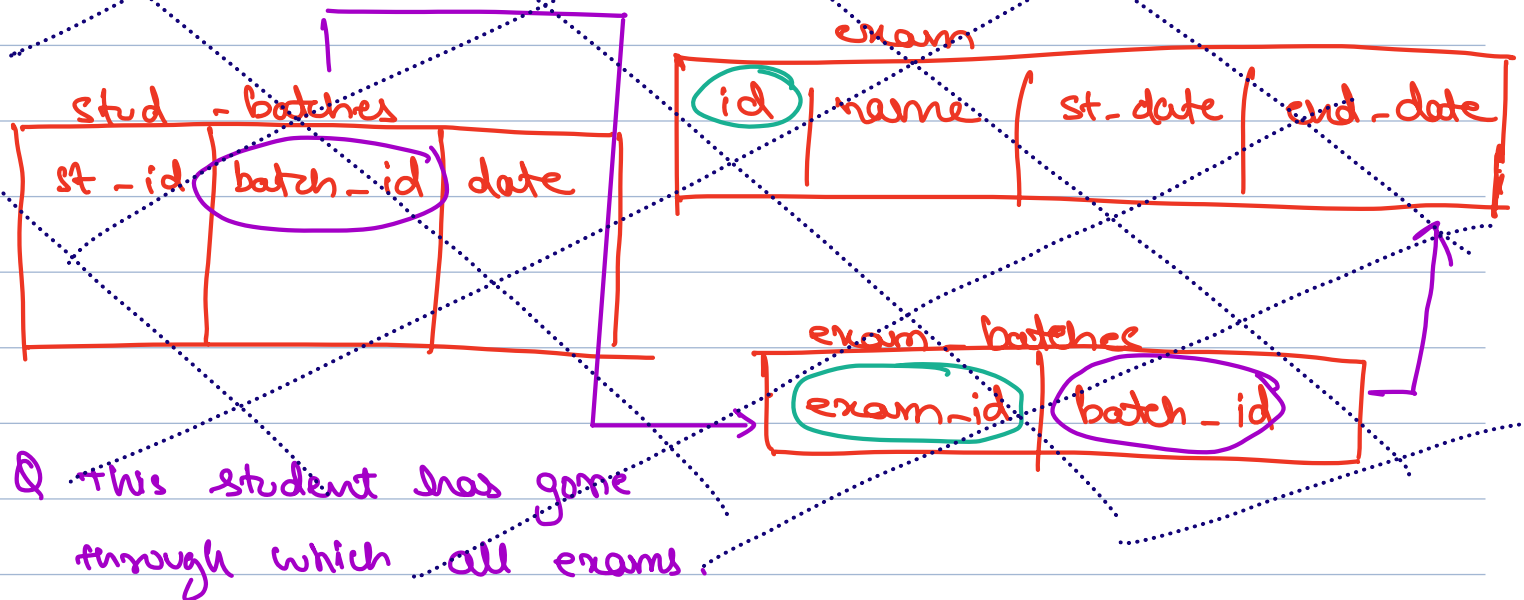
VS.



By default we have index on PK, so whose index will be bigger.

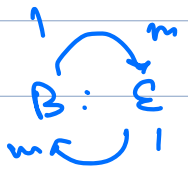
→ Size of index will be bigger in ①

① Scalex has exams
② for each batch a student joins they will have to give exams of that batch
③ Each exam is associated to a batch



Q this student has gone through which all exams.

① One student can belong to multiple batches
② Every batch has exams
③ Same exam may happen on diff batches on diff dates.
④ If a student moves a batch, they may have to give some exams again.



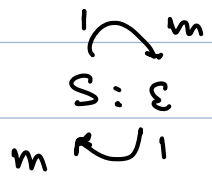
student - batches

st-id	batch-id	st_date

batch_exams

id	batch-id	exam-id	date

Q which student has given exam for which batch.



student_exams (pk)

student_id	exam_id	marks
1	1	
1	1	

Do you think this table is enough?

One student can't give the same exam twice.

You will also have to add batch_id to the PK.

student_batch_exams

stud_id	batch_id	exam_id	marks
—	—	→	—

↓

batch - exam - id

⇒ Sometimes a mapping may also have a relation with another entity.

↓

in those cases not having a PK can cause problems [space issue]

Adv of sep key

→ If a rel" is begin mapped to another entity [rel" → saves cost

Adv of no separate key

→ (batch_id, exam_id)

If no sep PK, queries on (batch_id) will become faster

↓

Because the table will be sorted by batch_id.

⇒ Bcoz mapping tables are representing rel" for the most time, it is very common to have joins on them.

Most of the times not having a separate PK makes more sense, it will make those joins faster.

How to represent FK and indexes

→ Rep doesn't matter explanation does

→ Along with schema design question, use cases are also mentioned.

eg. → find all the exams of student
find all batches of student been to.

use cases govern what indexes will be there.

→ we need the fn to find all class of a batch.

→ learners often search mentor by a name

→ FK is mentioned alongside creating schema
→ tables
→ attributes
mention FK ⇒ → ref rel^{n}

→ x coln of y table will have a FK referring to z coln of a table.

★ Talk about indexes at the end.

↓

with {reason} → to make which use case faster.

#

Netflix schema design.

→ take time, review the doc given.

Entities.

- Users
- Profile
- Profile_type
- Videos
- Actors
- watch_status_type

we need to store info about cast

↓

a mapping b/w video & actor

user
- id
- email
- password

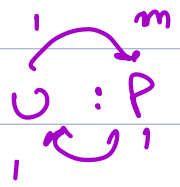
profile
- id
- name
- user_id
- profile_type_id

profile_type
- id
- value

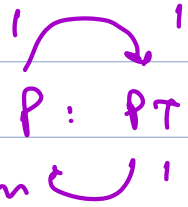
videos
- id
- title
- desp
Rel" → cast X

actors
→ name
Rel" → list<videos> X

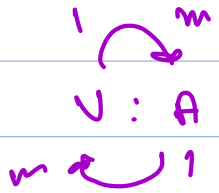
watch_status_type
- id
- value



1:m



m:1



m:m

video_actors
- video_id
- actor_id

video_profile
- video_id
- profile_id
- status
- last_watch_timestamp

Q what should be PK of video_profile?
① (video_id, profile_id)
② (profile_id, video_id) ✓
③ (id)