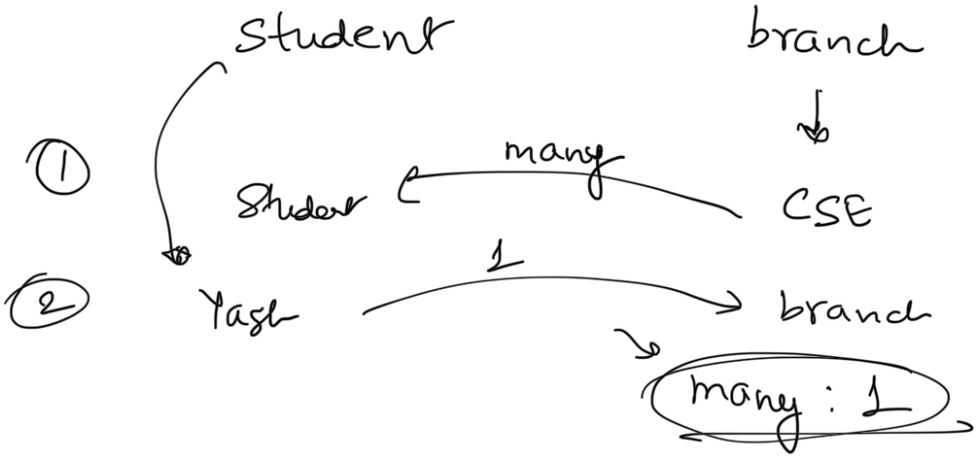
SCHEMA DESIGN - 2

AGENDA

- SPARSE RELATION
- SCALER SCHEMA DESIGN
  - CHOOSING PRIMARY KEY
  - Foreign Key + Index.
- [OPTIONAL] SCHEMA DESIGN OF NETFLIX

SPARSE RELATIONS

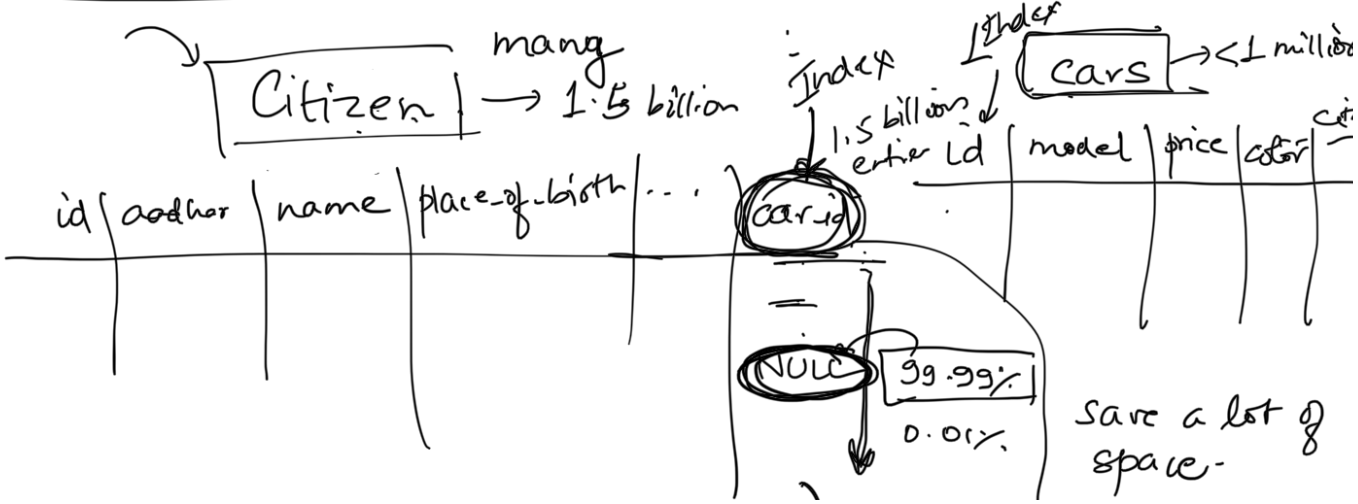


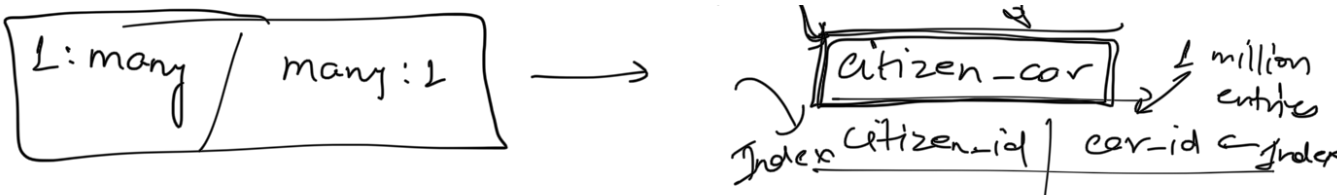
Student.

id	name	age	...	branch_id

branch

id	name	HOD	...



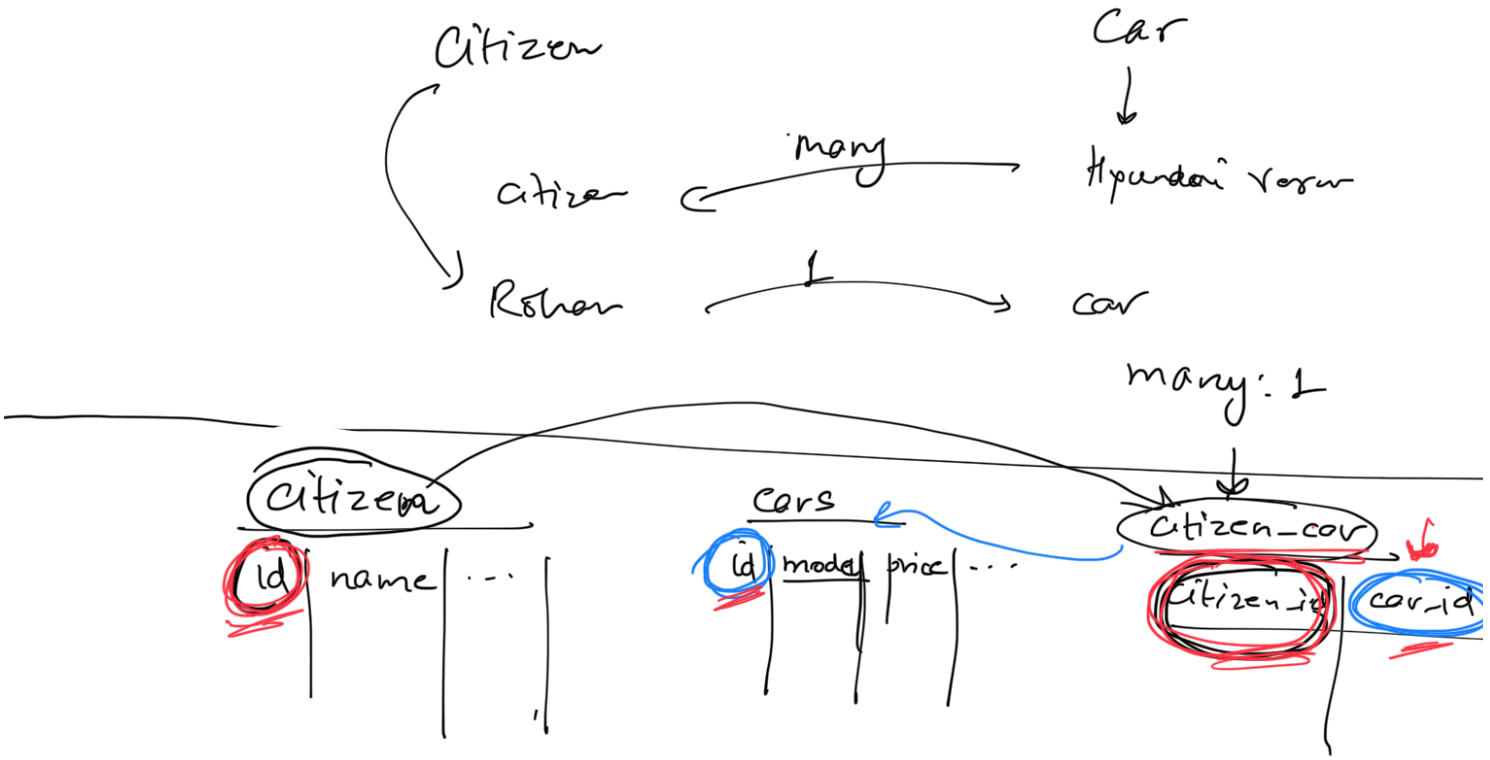


① (1:1, 1:many, many:1) → Should not create separate table. Add a column

→ Exception : Sparse relationship

man —marriage→ wife

② many : many → create a sep table for relationship



name, car_model

```
SELECT . . .
    citizen c
        JOIN citizen_car cc
            ON c.id = cc.citizen_id
        JOIN cars cr
```

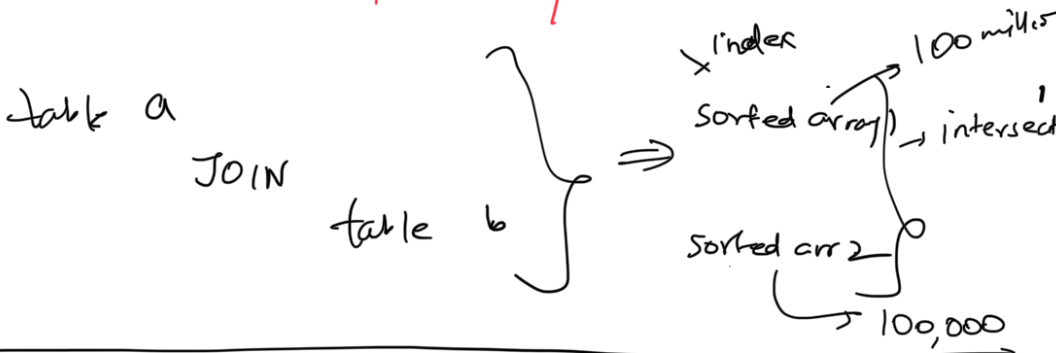
c is null

ON cr.id = cc.car_id

0
multiple

1 X
1 Y

a = NULL X



Schema design

- Nouns
- Relationship → cardinality
- attributes

- Multiple batches
- Batch → name, start month, current_instructor, Course
- Batch → a list of students.
- Batch → a list of classes.
- Classes → name, date & time, instructor.
- students → name, graduation year, university, email, phone_number, attendance.
  - move batches
- Student → mentor
  - name
  - current employer
  - email, ph_no.
- mentor - Sessions → time, duration, agenda, Studentrating

mentor_rating.

– instructor

students | instructors | mentors | classes

batches | courses

CREATE TABLE student (
id INT AUTO_INCRE pri—,
email varchar(50) UNIQUE,

student

id	name	email	ph_no	grad_year	univ	current_batch_id

instructor

id	name	email	employer

mentors

id	name	email	employer	grad_year

classes

id	name	topic	datetime	duration	instructor_id

class_type

batches

id	name	start_date	type

courses

id	name	. . .

batch_classes

batchid, classid

batch_id	class_id	. . .
1	1	
1	2	
1	3	

class

Student_classes

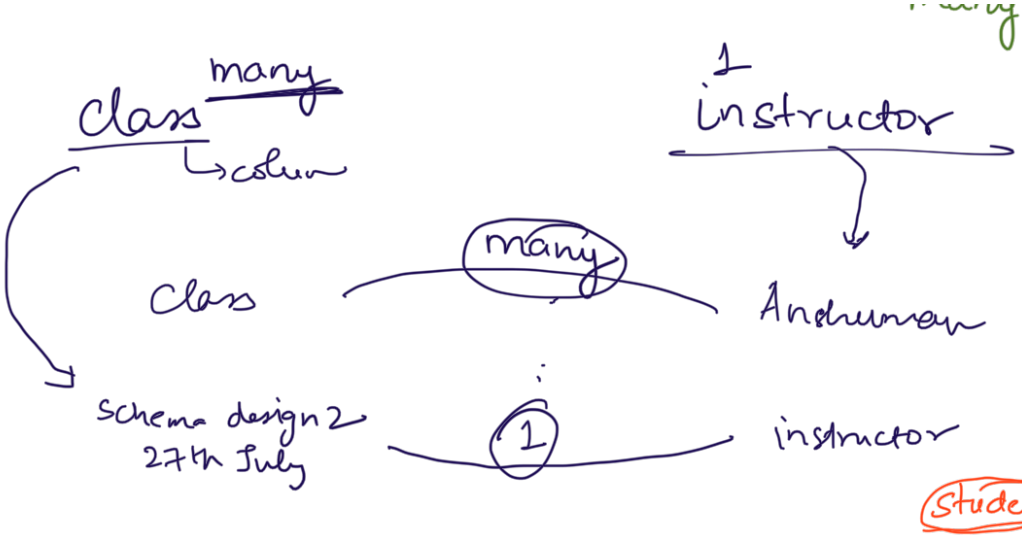
student_id	class_id	perc_attended	is_bookmarked	rating
1	2			
1	3			
2	4			

many

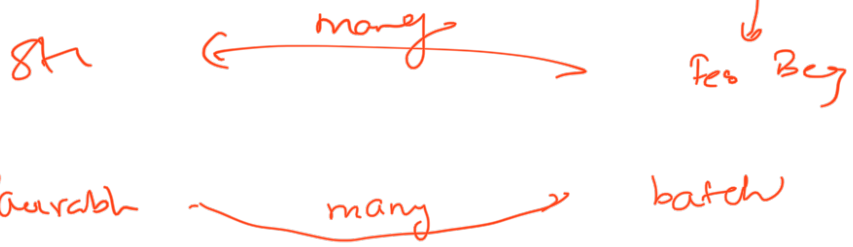
Schema design 2

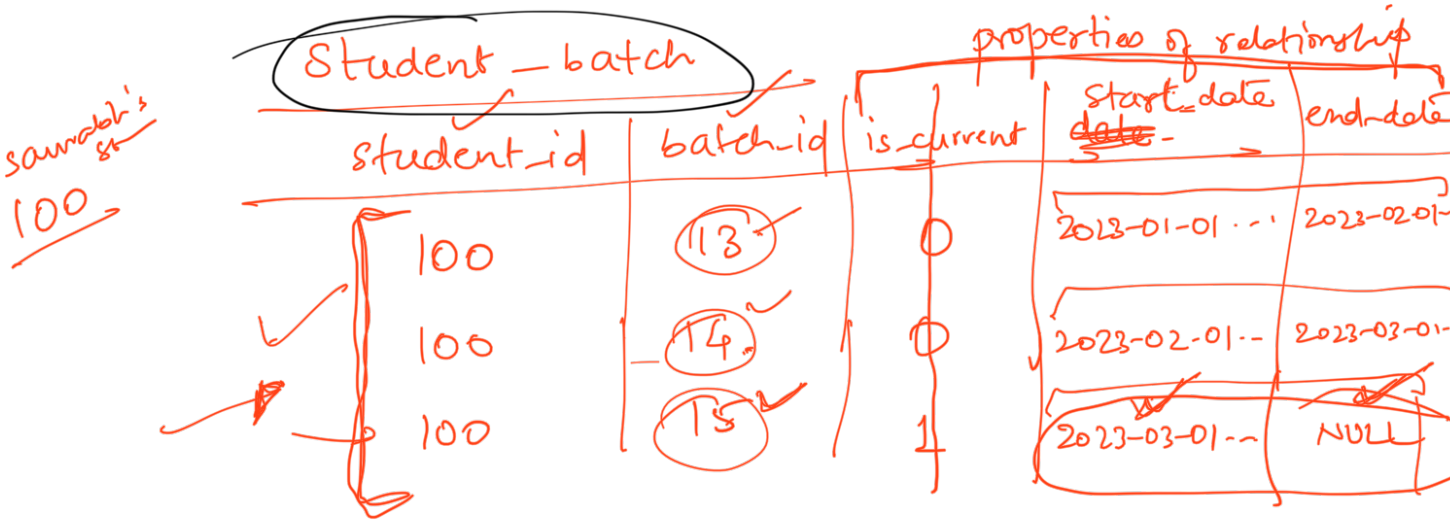
Nov22 Beg —many→ Class.

many : ma...



Student batches





Current_batch

Mentor Students mentor_id

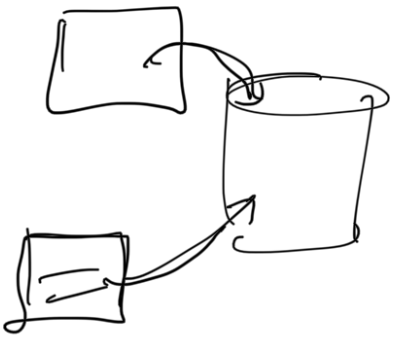
Mentor_session

10:40 pm

Classes

3 kinds of values live

class_type → recorded
enum
problem_solving
live
recorded
problem_solving
reorded

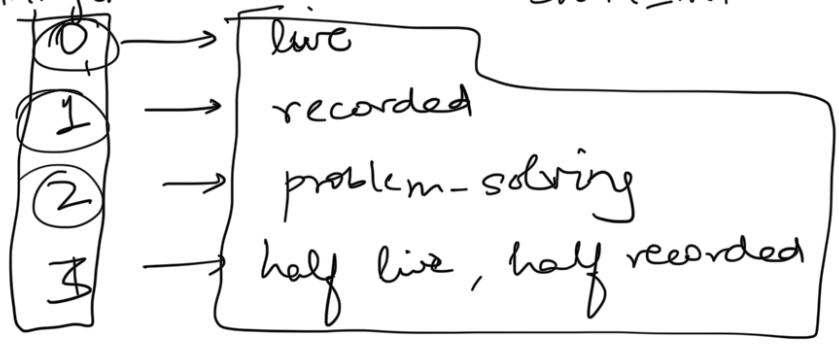


1. Human errors
2. Storage space needed

Option 1: Use strings

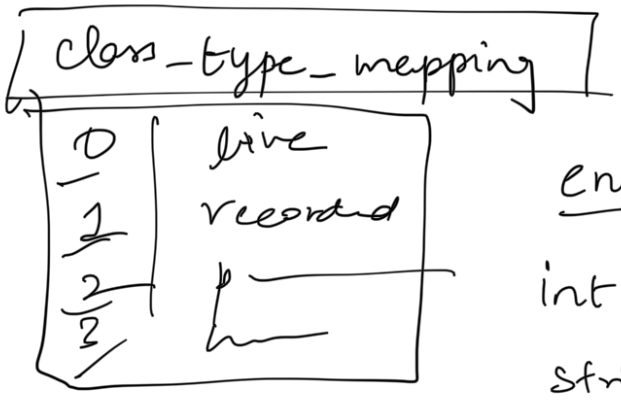
Option 2:

integers → short_int



Con: ~~there~~ mapping not being available

Option 3:



enum

int a

string

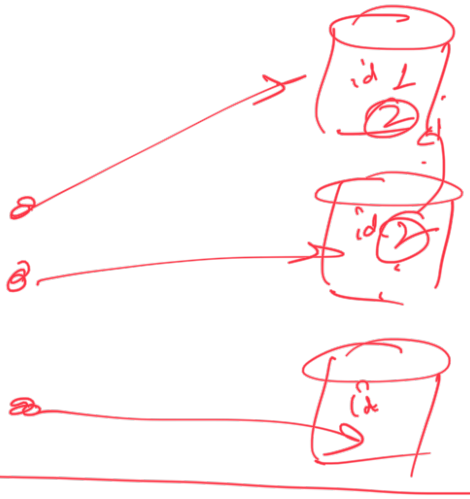
uuid

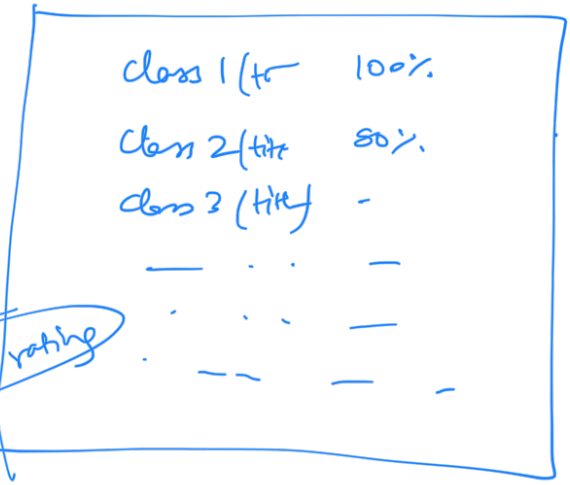
- timestamp
- machine identifier

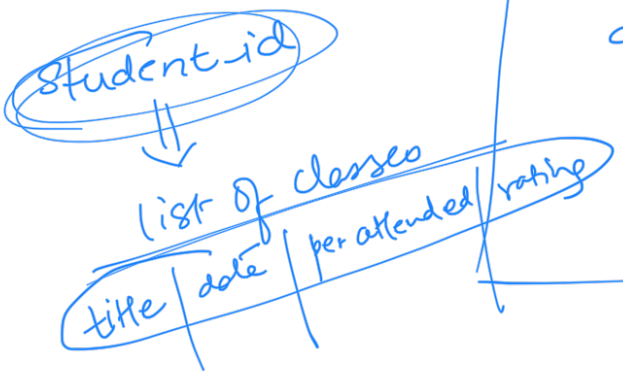
process + thread id

client

uuid





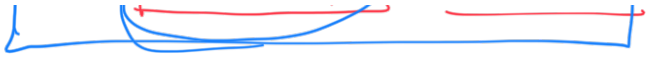



```
SELECT
    c.title, c.date, sc.per_attended, . .
FROM  student_classes sc
JOIN  classes c
ON    c.id = sc.class_id
AND   sc.student_id = <student_id>
```

Already has index

classes
id

student_classes
student_id → class_id



Steps

(1) Filter student_classes table on student_id = xyz

O(N)

For every matching row, classes.id = class_id

SELECT

name, university

FROM students

WHERE email = XYZ

email

SELECT

film.duration, ...

FROM film f

JOIN film_actor fa

ON f.film_id = fa.film_id

pri

index existing

O(log N)

X

f.film_id

fa.film_id

loop

Intersection

O(N)

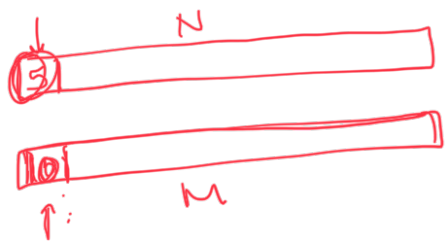
N log(N)

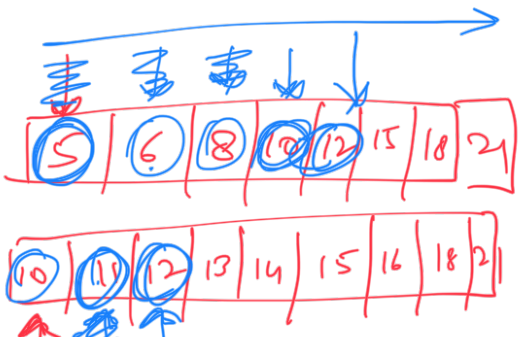
film

filmid ... ---

10

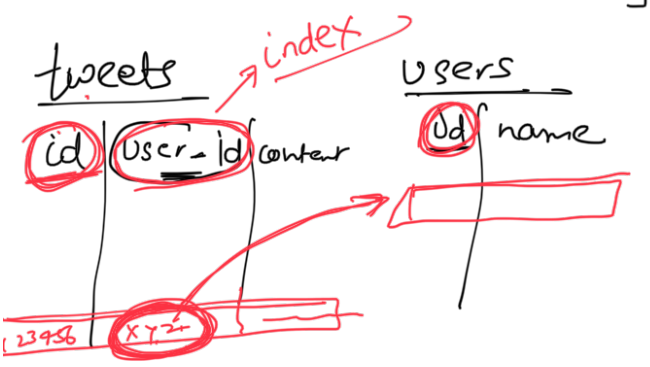
12





O(N+M)

```
SELECT
    t.content, u.name
FROM    users u
JOIN    tweets t
    [ ON    t.user_id = u.id
      AND   t.id > 123456
```




```
SELECT *
FROM    tweets t
LEFT JOIN   users u
    ON  t.user_id = u.id
    AND u.name = "Vijay Sharma"
    AND u.city = "Lucknow"
WHERE
```

tweets
index on user_id

users
(name, city)
name ✓
city ✓

10 million tweets
userid

users

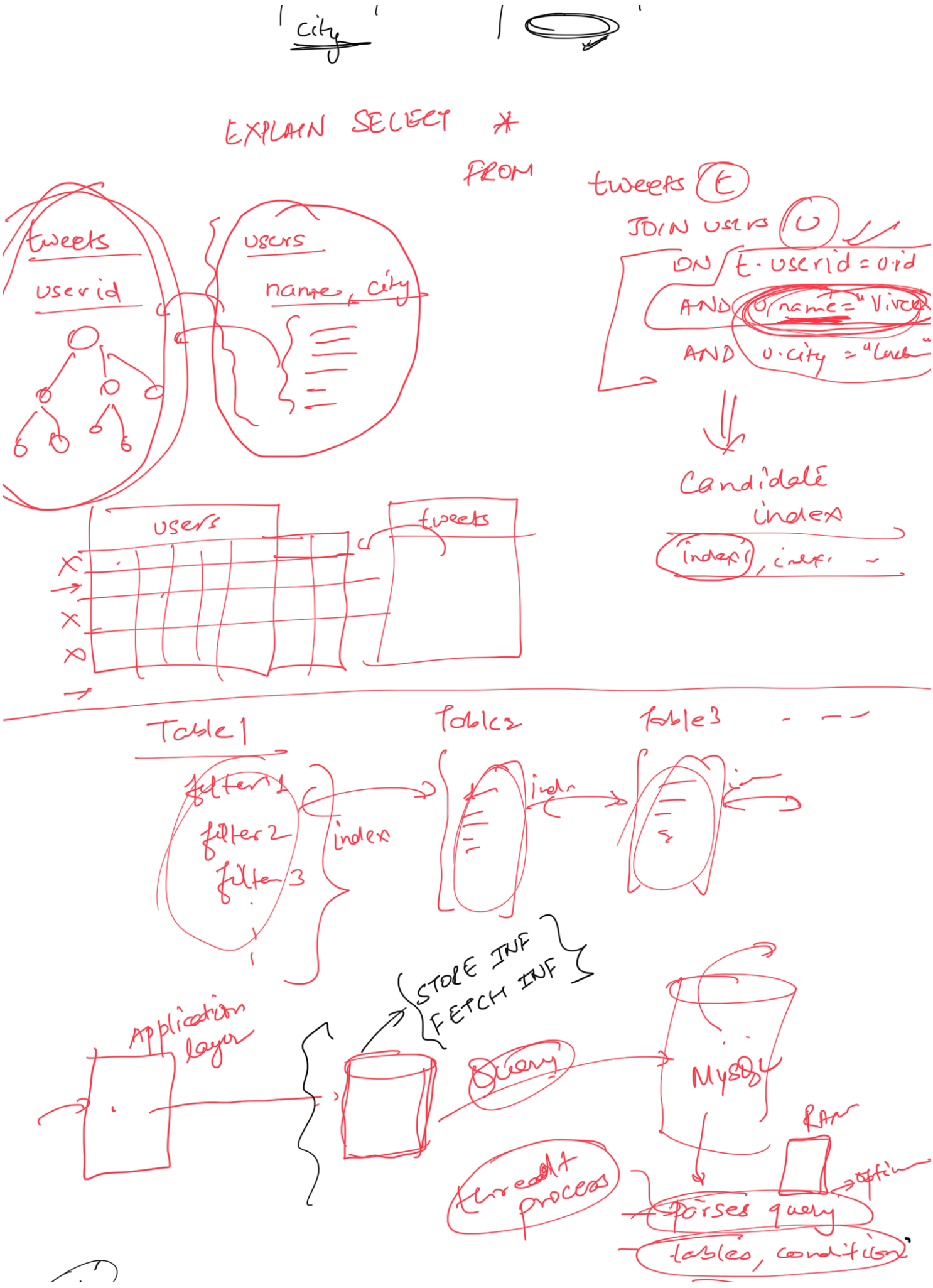
10 million

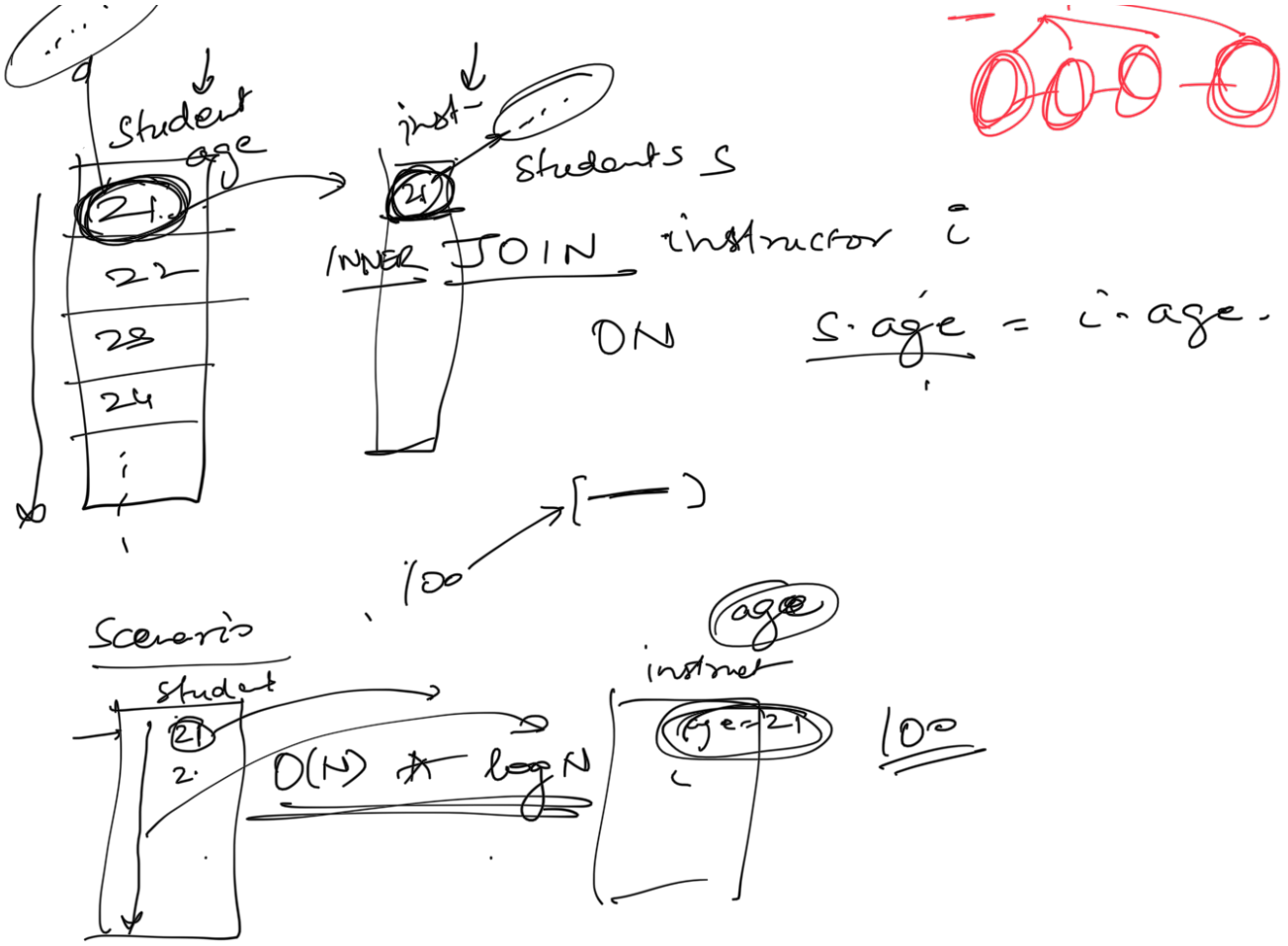
O(N)

SHOW INDEX FROM —

10 million

index name	cols	...	cardinality	...
	name			


city
EXPLAIN SELECT *
FROM
tweets
userid
users
name, city
tweets (t)
JOIN users (u)
ON t.userid = u.id
AND u.name = "Vivek"
AND u.city = "Leeds"
Candidate
index
index1, index2
users
tweets
Table1
Table2
Table3
filter1
filter2
filter3
index
index
STORE INF
FETCH INF
Application layer
Query
MySQL
RAM
thread
process
parses query
tables, condition


Student
age
21
22
23
24
i
inst
Students S
INNER JOIN instructor i
ON S.age = i.age.
100
Scenario
Student
21
2.
O(N) * logN
age
instruct
age=21
100