

DBMS - I

Naman Bhalde

⇒ Module lead of CS Fundam
of LLD

⇒ Product lead @ Scaler
Topics

⇒ SWE@ Google
One fit
Clipsy

Agenda: Introduction to DBMS

- What is a Database
- Normalized NAMC

Relational DBMS

ER Diagrams

DBMS Module

- Normalization

~~Project~~

MySQL → [SQL (1-2) Basic / Advanced]

- Indexing (B-Tree)

- Query Optimisation

- Transactions

→ ACID

LCD

2 Schema Design

DBMS

Database

Management (M)

Systems (S)

What is Data?

Information about different thing

Eg- Student

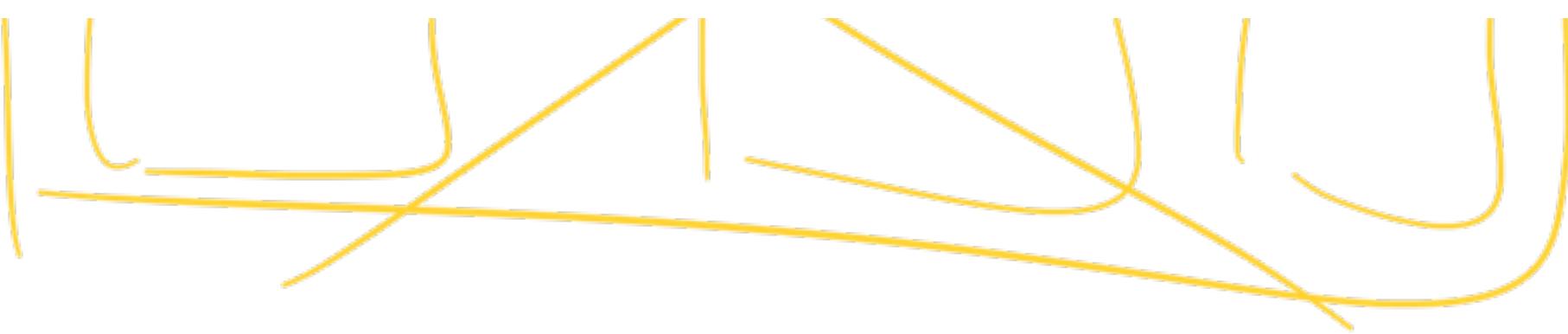
What is a Database

Collection of data

Scalar Database

Student Courses Mentor





DBMS

→ Way to store / Manage / query data
Searching / organized / delete

Types of DBMS

→ List of Scalar Students

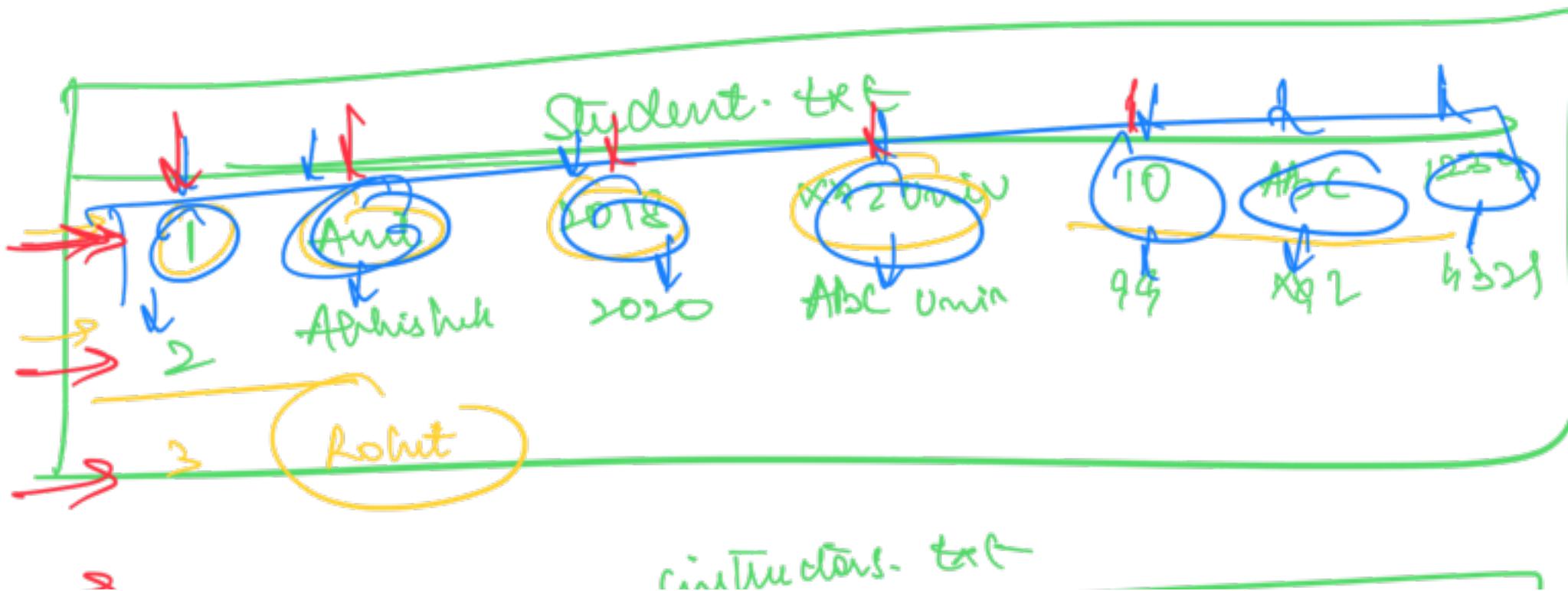
- ① Excel / Google Sheet
- ② File

① files as a Database

→ Students, mentors, assignments,
instructions

Studenti.txt mentors.txt assignments.txt

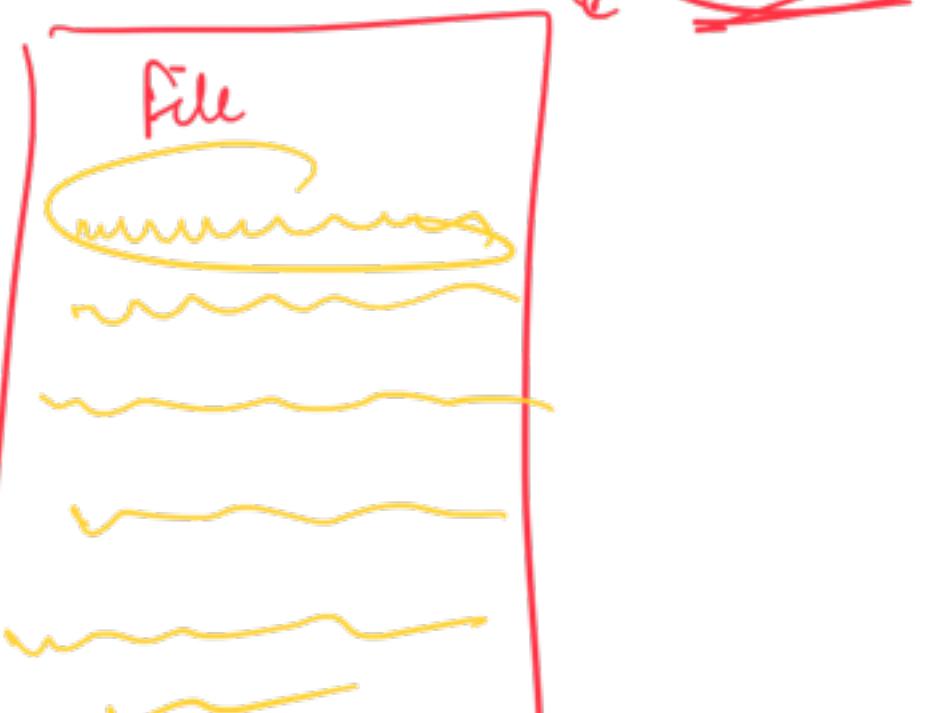
instructions.txt

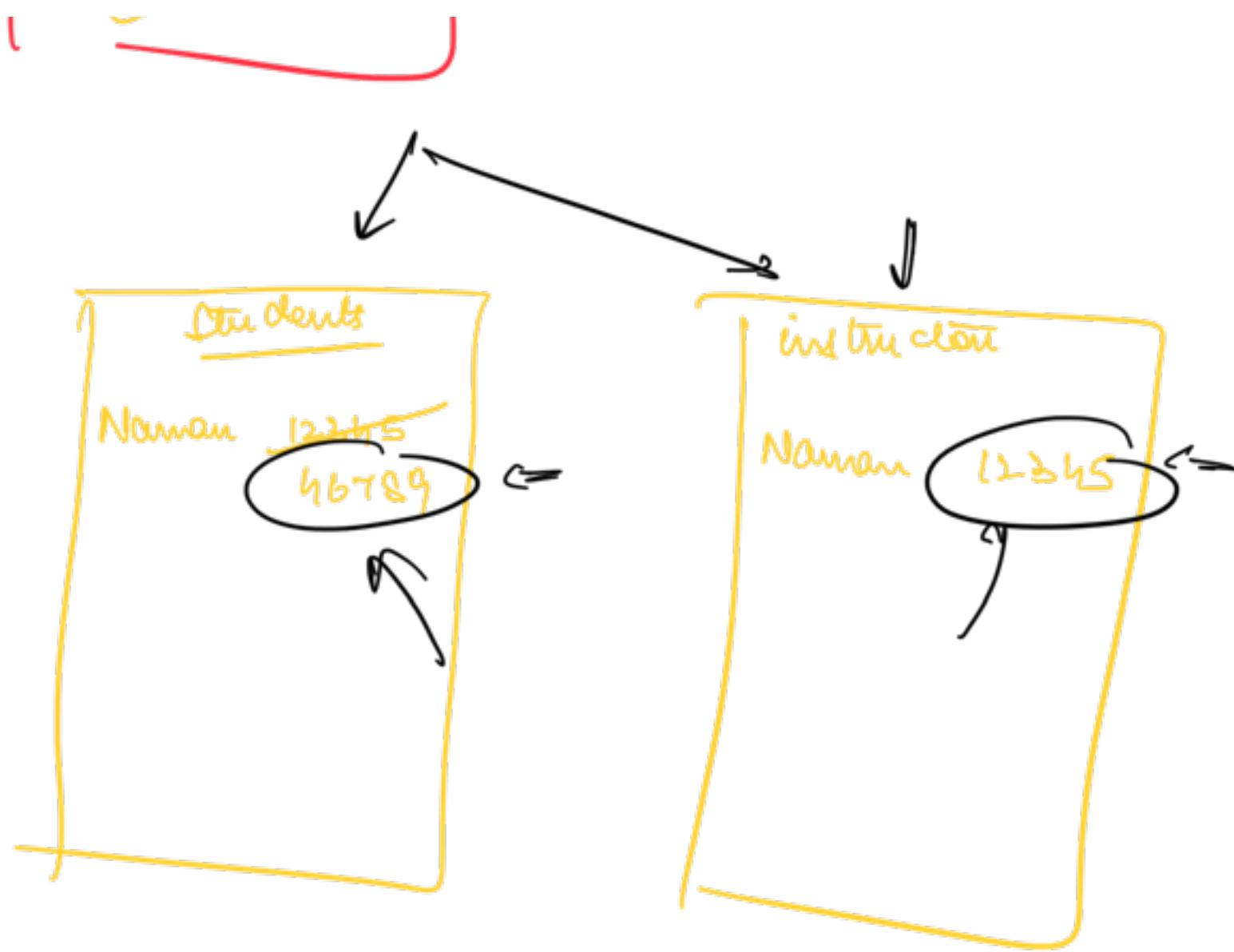




Cons

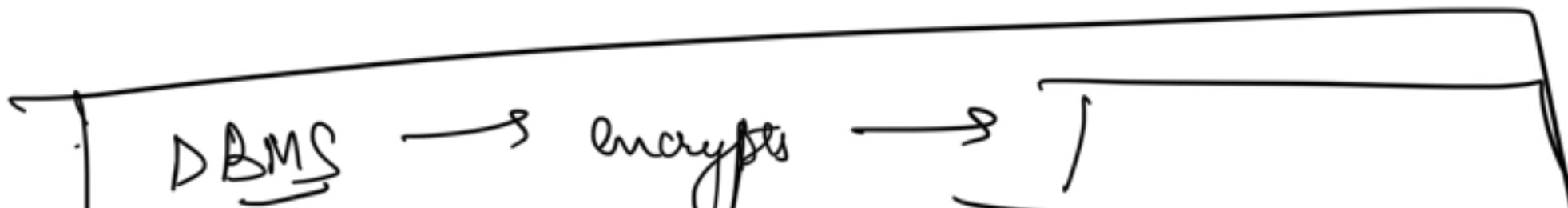
- ① Searching (Not efficient)
→ files have only sequential access
- ② Redundancy : Storing 1 info at diff places.
- ③ inconsistency :





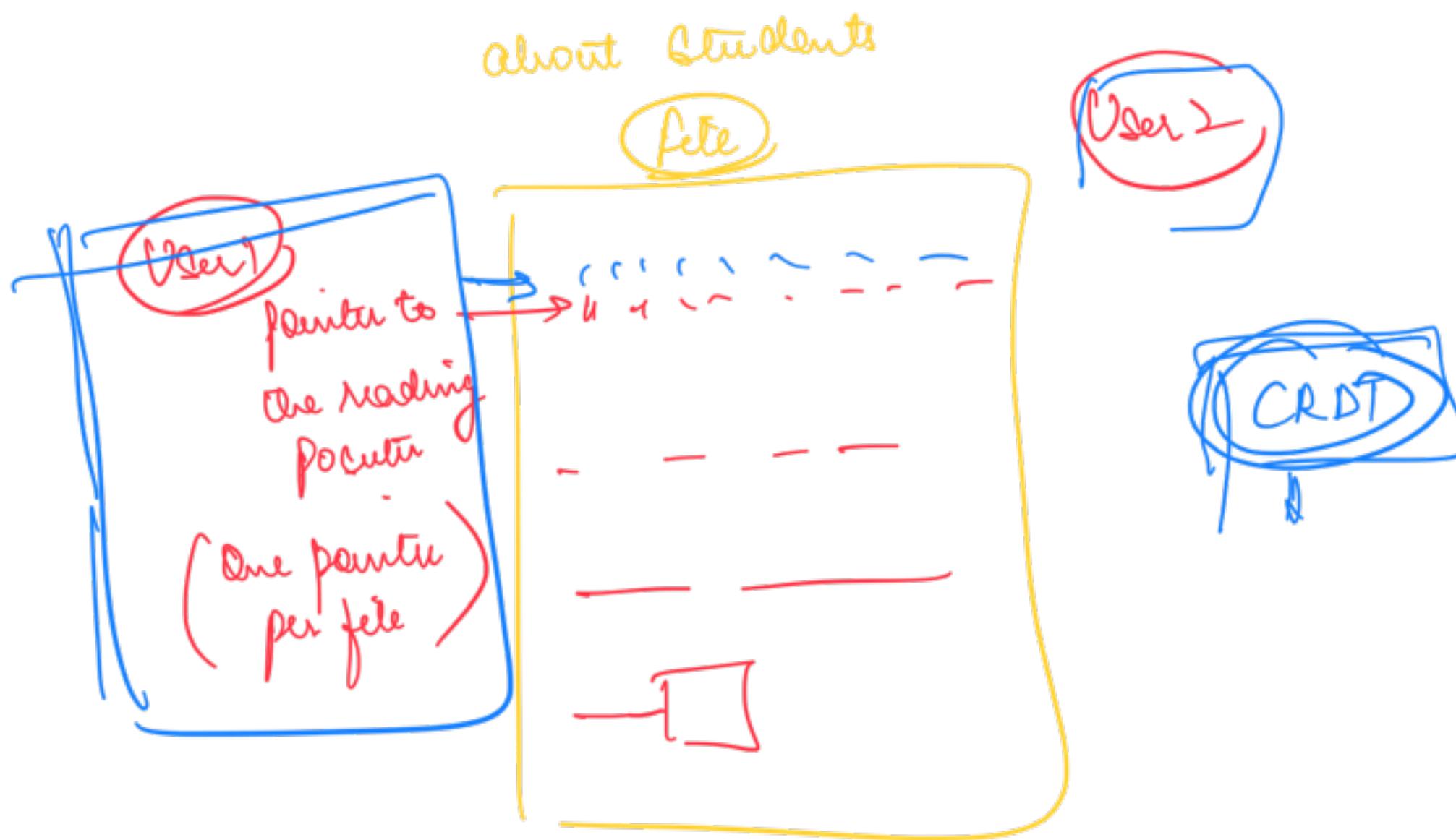
④ Security

Often files are stored as it is on
the disk



⑤ Concurrent Access

Situation Multiple people want to read info about Students

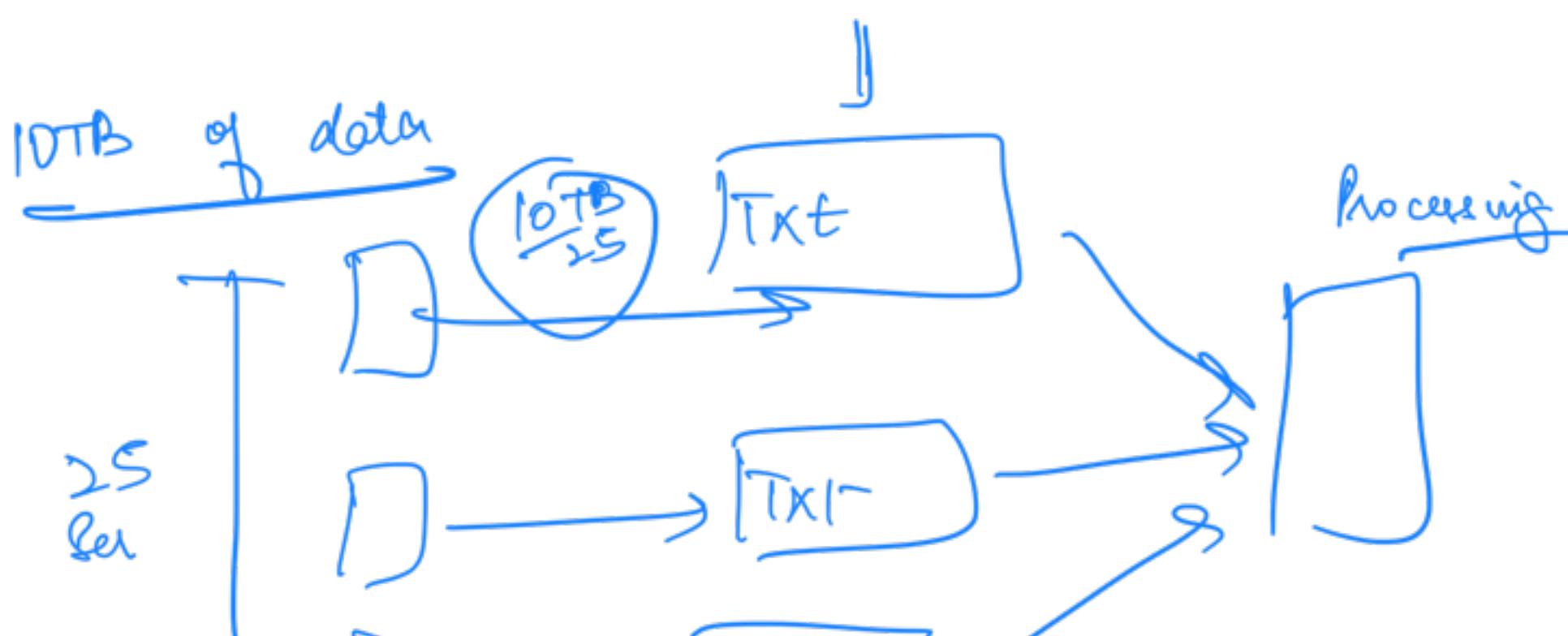


(IM QPS)

In real systems, file based DBMS are used only

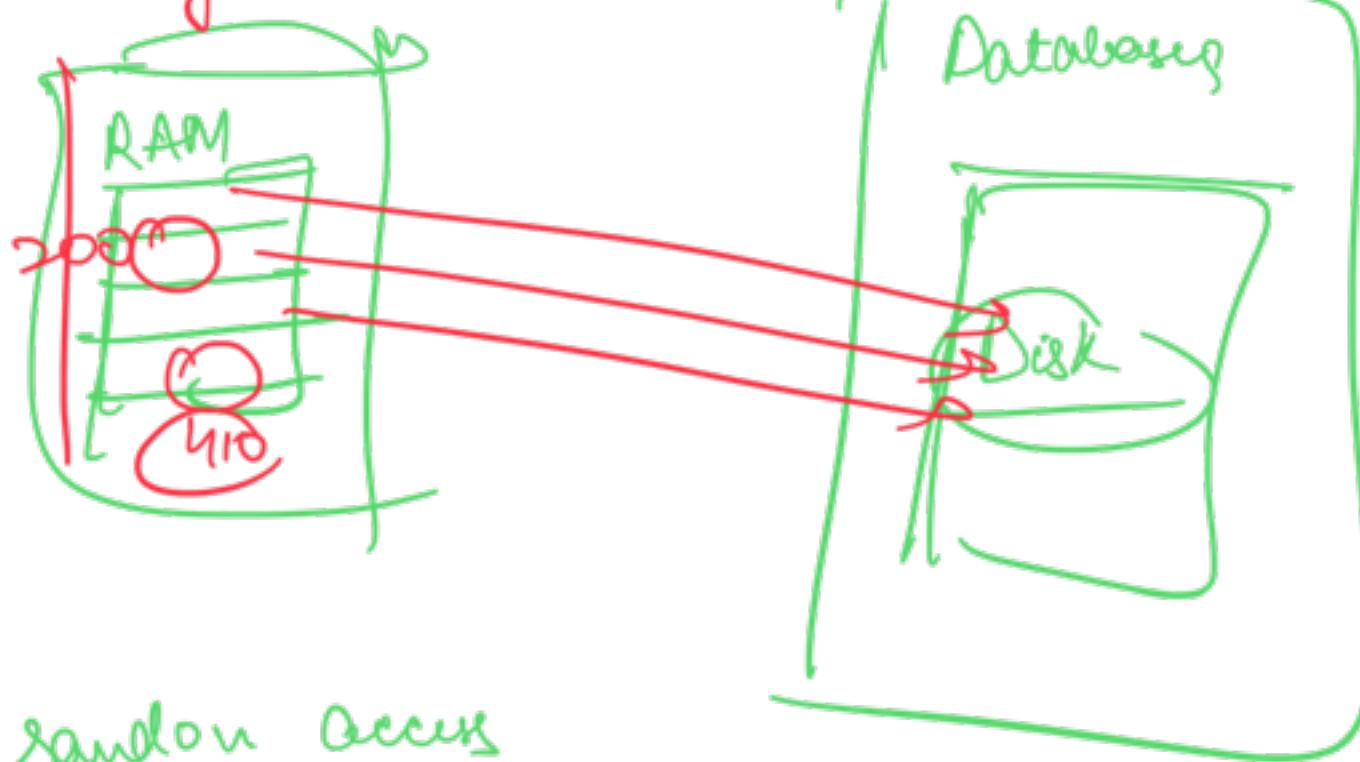
if

- ① There are no updates = (once per day)
- ② file is small (500 chars)
- ③ Very low throughput (1 RPS)



+ D → i Tx (-)

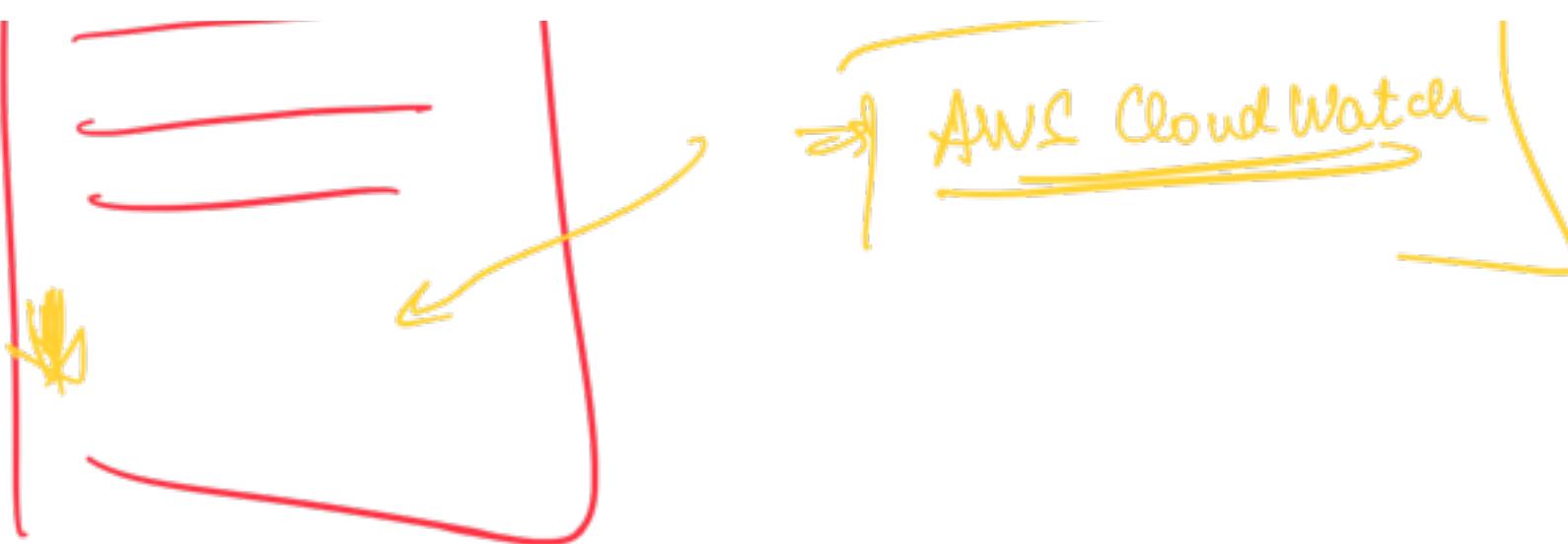
In RAM we can do
randomly access



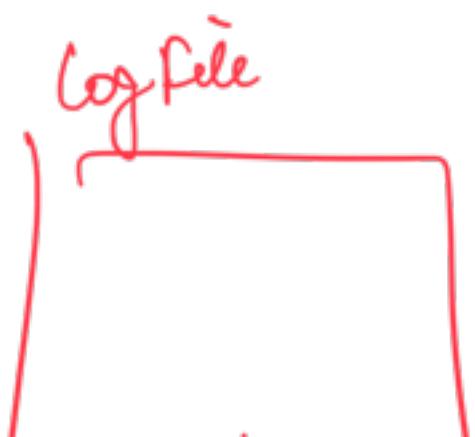
In RAM → random access

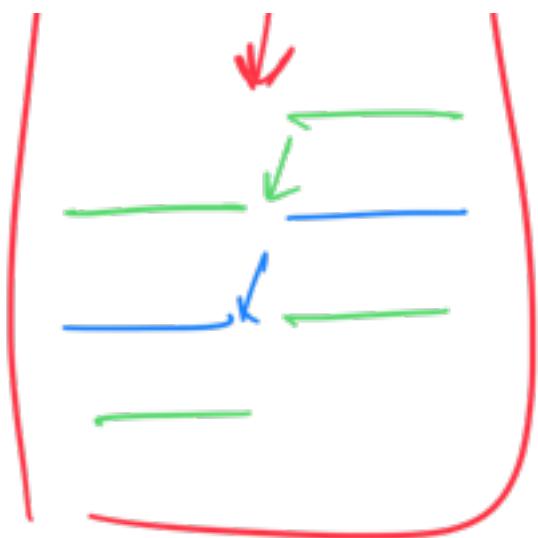
Disk → Sequential Access





① Whenever we open file,
a pointer to the start
location of file is created



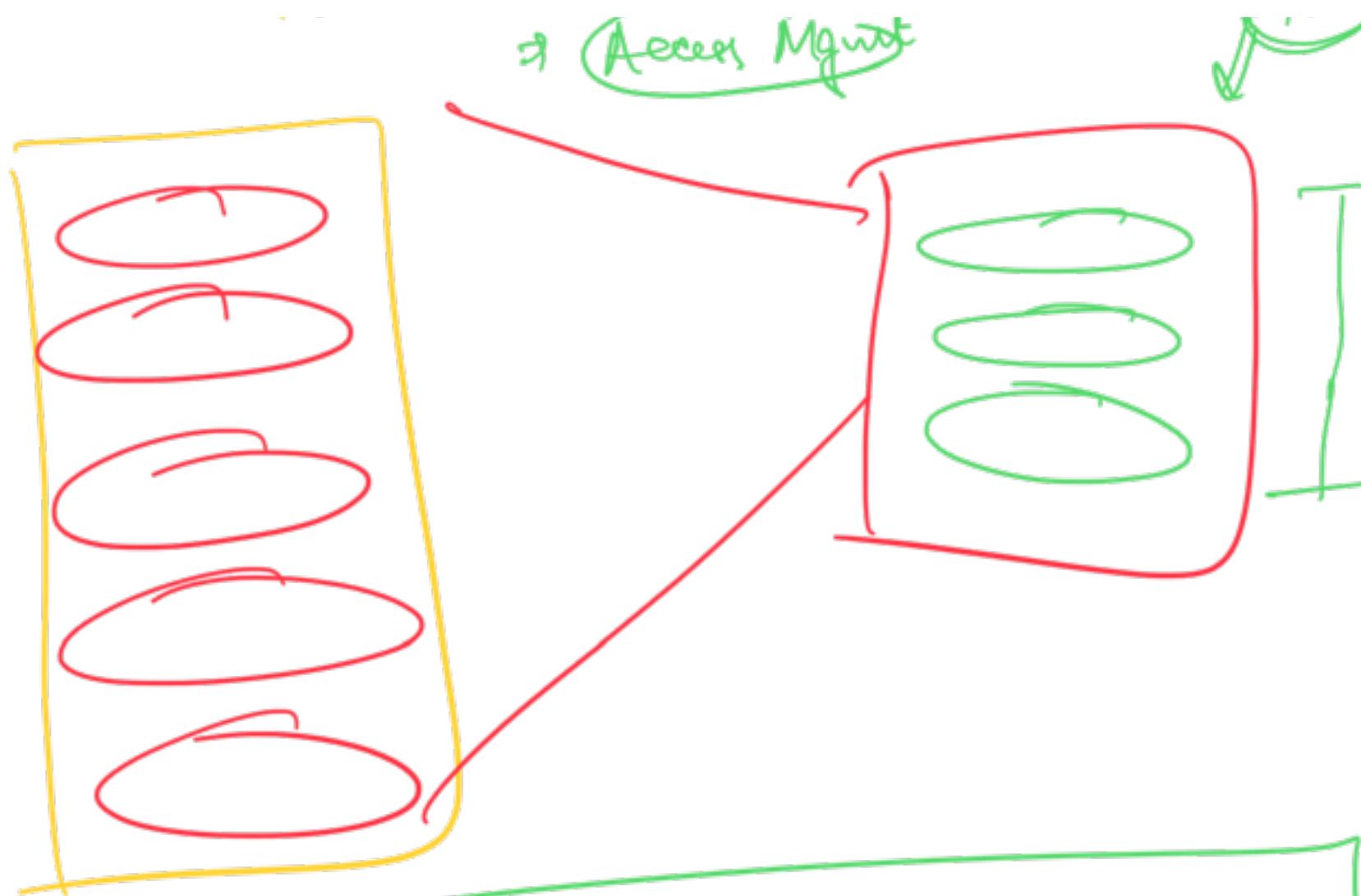


Dedicated DBMS

- Backups
- Concurrency
- Security → Encryption
- Efficient access / storage
- ~~consistency~~ (Avoid inconsistency)
- Views → Security .



TA



Types of Dedicated DBMS

① Relational Databases (SQL)

→ Data are multiple tables that are

~~same row~~ ---
related to each other

→ Store one type of data in one table

Students

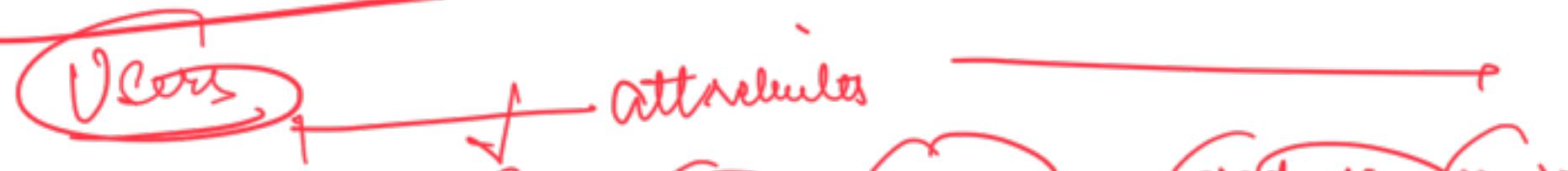
Instructors

Mentors

batches

Strength	Name	Dept	Curr. Year

Scalar Database



→ Name, Email, phNo, (password), (grad-year, sem)

Batches

Name	Cur. IIS	Start-date
------	----------	------------

Mentors

Name	Email	Cur. company	PoE
------	-------	--------------	-----

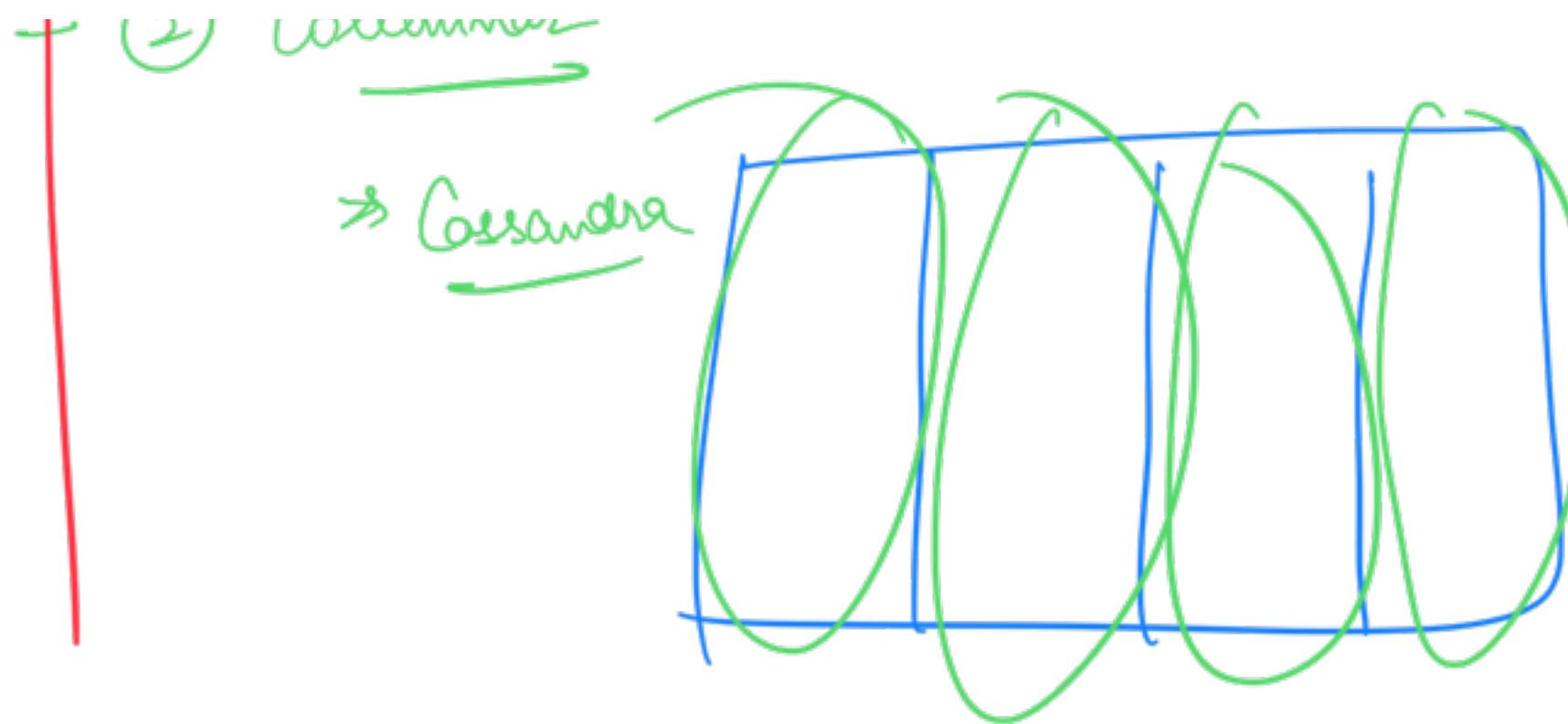


Other Databases

NoSQL

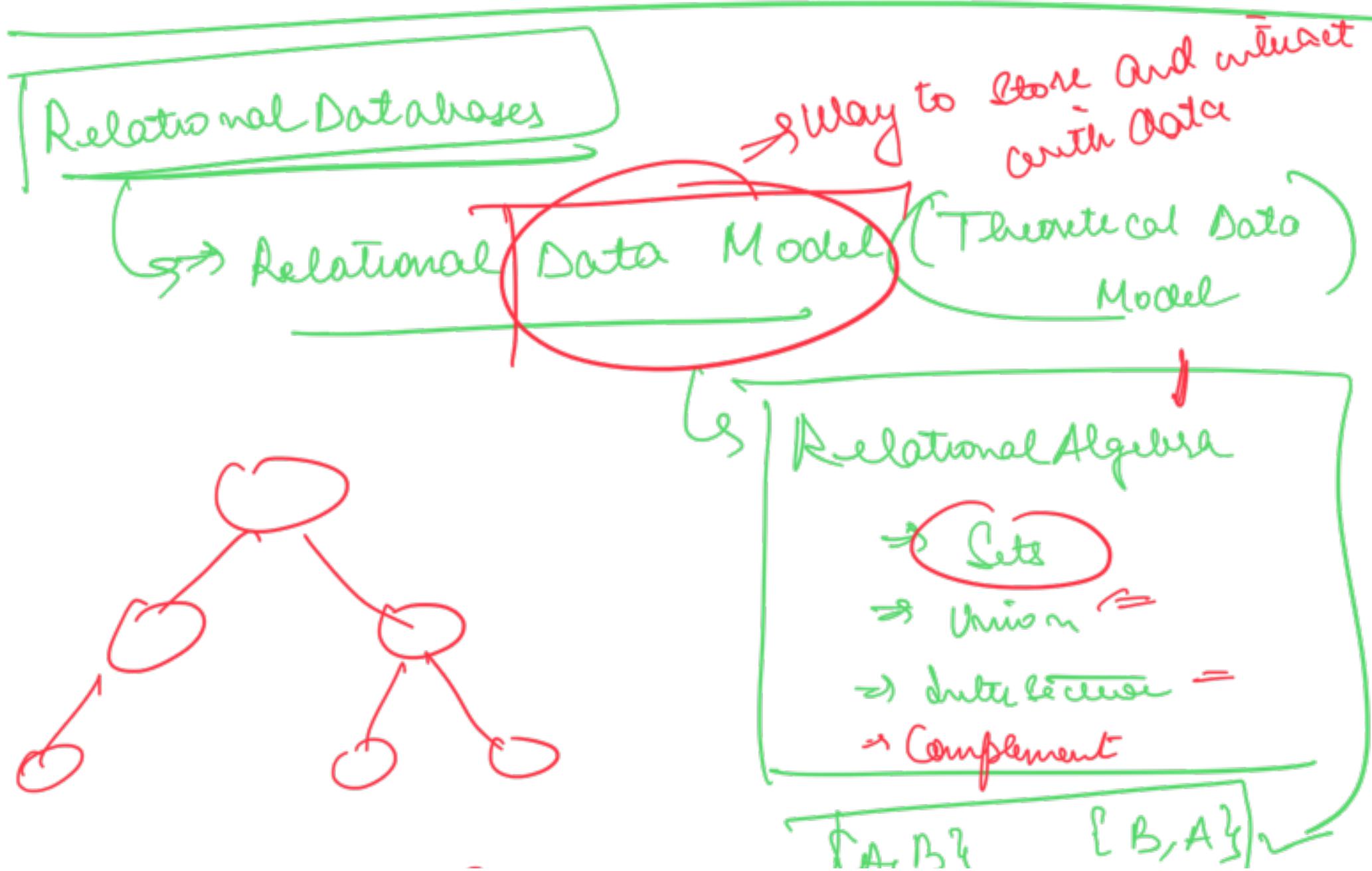
① Graph based ~~Graph~~ Neo4J

Columnar



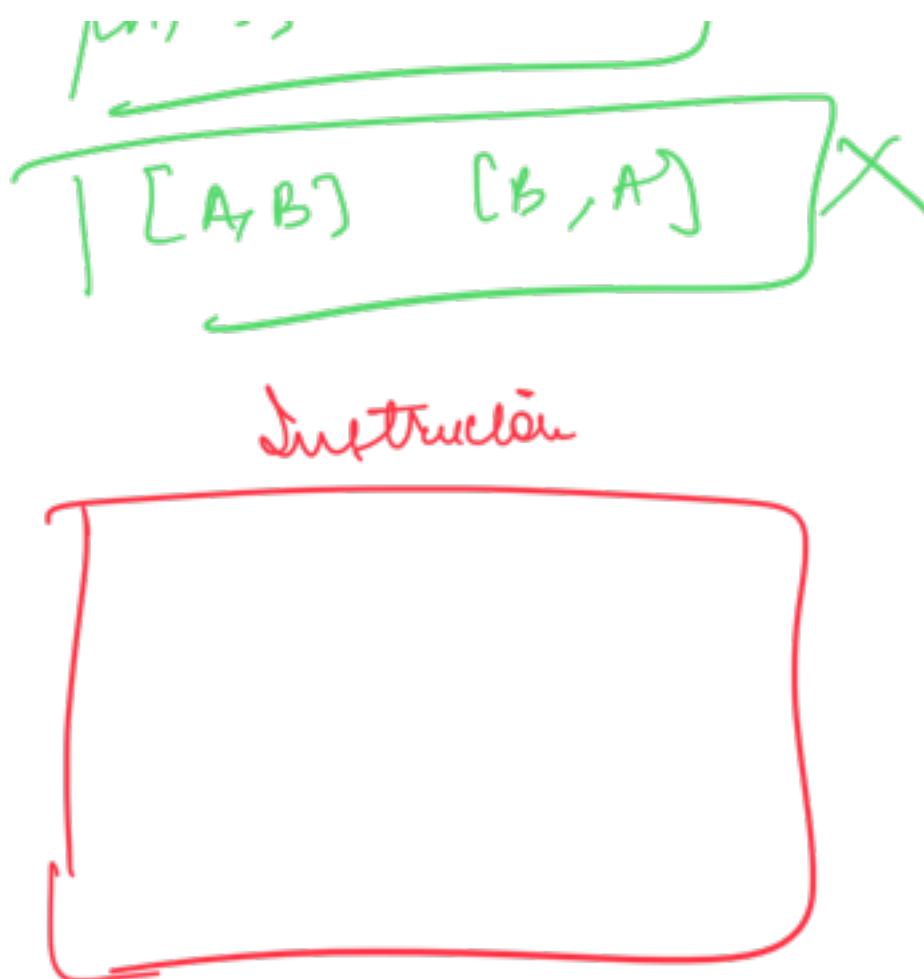
- ③ Time Based ~~Time Scale Nb is internally in SQL~~
- ④ Document ~~infex~~ ~~Time Scale Nb is internally in SQL~~
- ⑤ Key Value (Redis)





SQL

Select *
from Students
where AND



Batches



Select * from Students
where Name = Noman

Subset

SQL and Relational Algebra

Relational Data Model

- ① Most widely used model ✓
- ② Data Model behind SQL ✓
- ③ rep data as multiple tables → Parent
- ↳ rep data as relations

Set / relation → table of values

Student

Mentor

instructor

Select from



(4) Every relation has multiple real instances each rep as a row

{Student}

Select* from Student

where Name = Name

{Student} \cap {Student}

Name

join \Rightarrow Get Multiplicities

{A, B, C} \wedge {C, D, E}

Student

Name

Ashwin

Sanchi

A C BC
AD BD
 C E CD
 BF CE
 C F CFz

Cartesian product

Select * from student

where

Name = Naveen

OR age = 21

SQL \rightarrow Relation Algebra

Terms

① Attributes: properties of a relation
a. domain b. ^ truth

② tuple \Rightarrow single info of a single entity/instance

Name	StudentId	Age	E-mail	Phone
Ranishk	21	abc@xyz.com	1234	
Abhay	22	xyz@abc.com	5678	

③ degree : number of attributes of a relⁿ
 $\Rightarrow 4$ for Student above

④ Cardinality : number of elements/tuples in a relation

$\Rightarrow 2$ for Student table above

$$R = T P A \quad b, c \quad Q = \{ D, E, F \}$$

\cup | \cap , \subset , \supset
 Cardinality of X^{∞} $\Rightarrow 9$
 $AD, AF, AF, BD, BE,$
 $\rightarrow 9$

$\textcircled{5}$ Null \Rightarrow Missing value

~~Go to~~ \rightarrow Null

Students

	1	1	1	1	1
Cardinality					
= 3					

degree = 5

Properties of a Table / Relation

- = ① every row is unique (at least 1 column will be diff than other rows)
- $$\{A, A, B, B\} = \{\underline{A}, B\}$$

	X	Y	Z	
A	A	B	C	D
A	A	B	C	D
				F

(typical)
(typical)

② every row should have same data type
→ for corresponding columns

Student		
Name	batch	grad-year 2020 "Two Three")

③ Order of columns doesn't matter

Select Name, batch from Students;

Select batch, Name from Students;

④ Order of rows doesn't matter

Select ^ from Student
P1

P2

Order by Name
age

Students	
Name	
A	
B	
C	

C
B
A

C
A
B

⑤ Every column name should be distinct

unique

| | Order doesn't matter

order
doesn't
matter

Same
data
type

⑥ Every value in a column should be

atomic

atomic →

Single values → 1, true, "Hello", 1.49

lists X
arrays X
maps X

["Name": "John", "Age": 23]
Name: Name
age: 123

MySQL → JSON
PSQL - JSON

map
objects

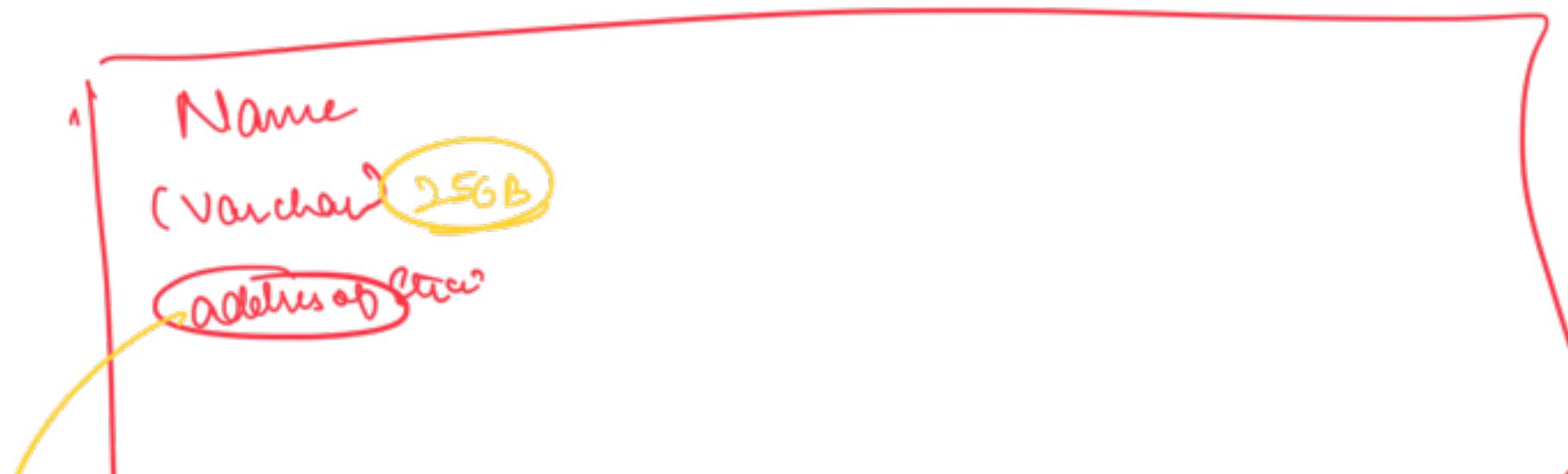
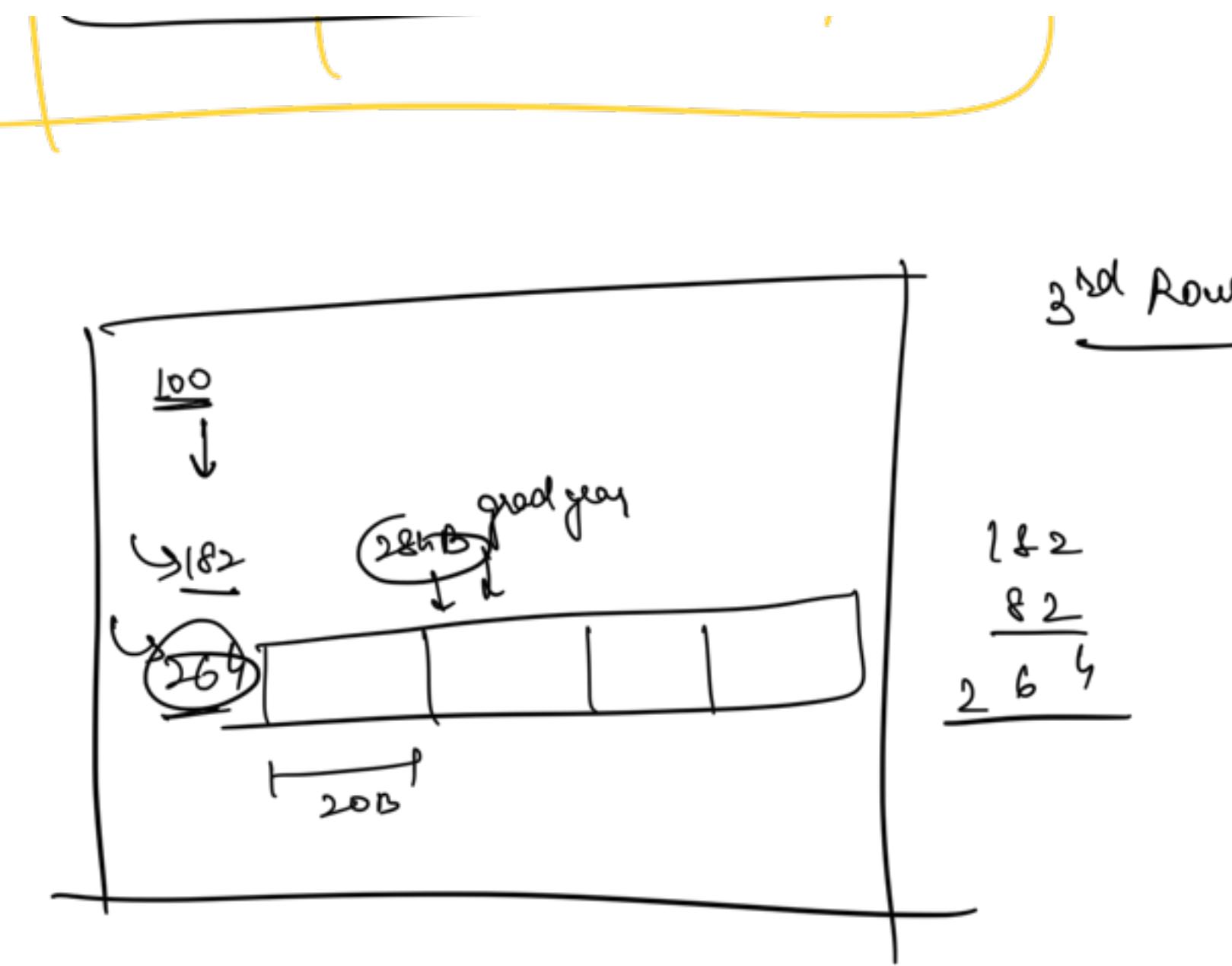
list

~~MySQL~~

Whenever I am selecting/filtering values

Because with collection data types comparisons
are difficult

Name	grad year	univ	P & P	list
String (50)	int (4B)	String (50) 50B	double 8B	= 82B



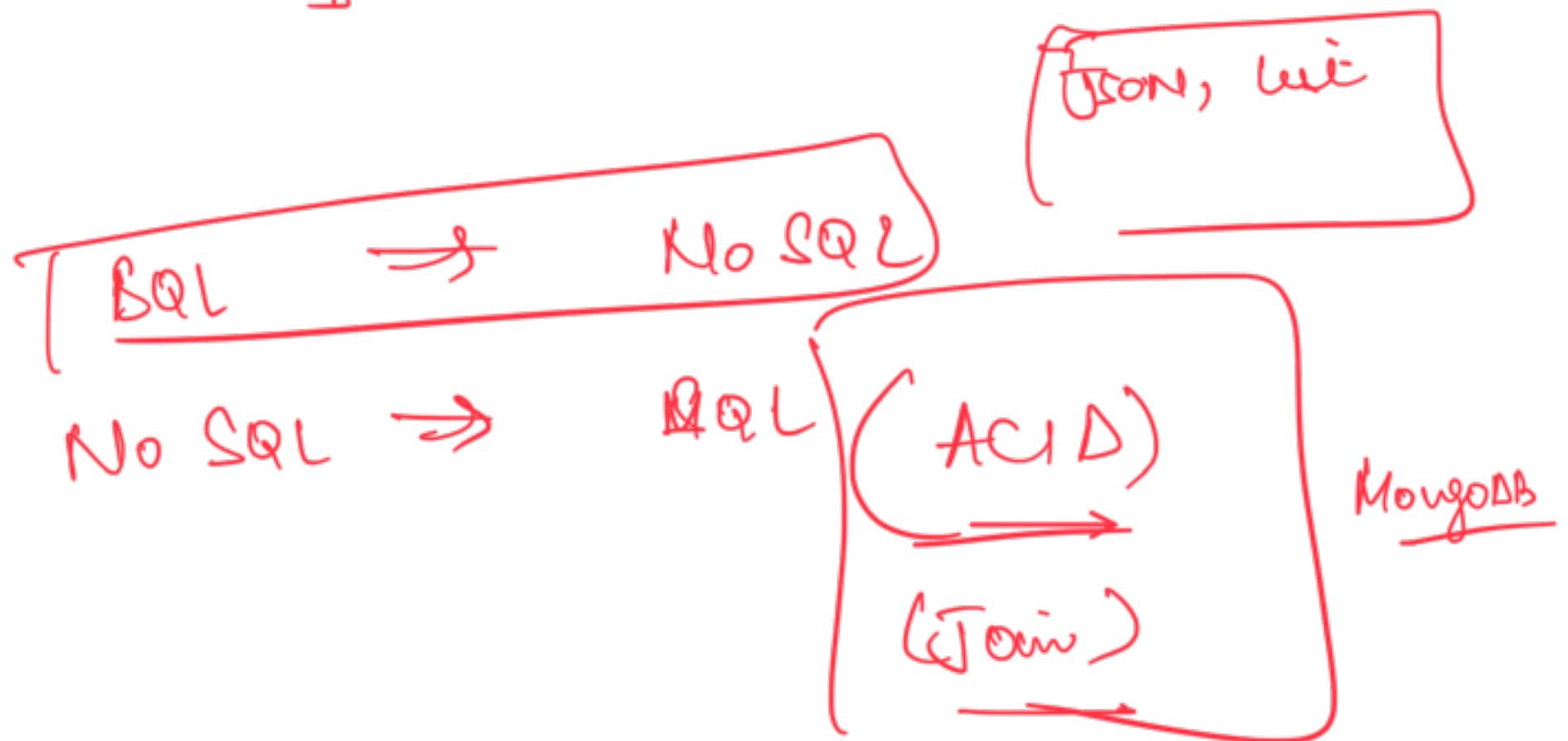


- ① difficulty to storing
- ② difficulty to compare

JSON Java Script Object Notation

```
{ data: [  
    {name: ABC  
    ...}]}
```

```
age: 12  
email: —  
1,  
{  
  Name: BCA  
  age: 14  
}  
3  
<  
>  
]  
§
```



do I have much too many writes \rightarrow NoSQL
too many joins \rightarrow SQL
too many reads \rightarrow SQL

Keys

① Identify a record / row

② Identify relations b/w 2 sets

rollNumber

Students

-id

id	Name	Email	phNo	batch	namebatch-id
1	Amit Kumar	amit.kumar@gmail.com	99941	X Augt Adv	

1	Devansh	Devansh@abc.com	12345	Sep 21 Mon
2	Ishan	Xyz@abc.com	12345	Sep 21 Tue
3	Name	abc@xyzd.com	54321	Oct 21 Mon
4	Devansh	123@abc.com	12345	Sep 21 Tue

- 1 Aug 21 Adv
- 2 Sep 21 Inter
- 3 Dec 21 Adv

Identifier

Create own Identifier so as to uniquely diff. every record

Types of Keys in a DB

{id, Name} ≥ 2
{id} ≥ 1

Students						
id	Name	Email	Ph No	PSP	Break	Batch-id

Superkey

A set of attributes that can uniquely identify a student

→ There can be no 2 students with the same value of the attribute set

(eg)

{id, name}

{ name, email }

{ id, name, email, phone Number }

{ complete attribute set }

{ id } =

{ email }

{ ph No }

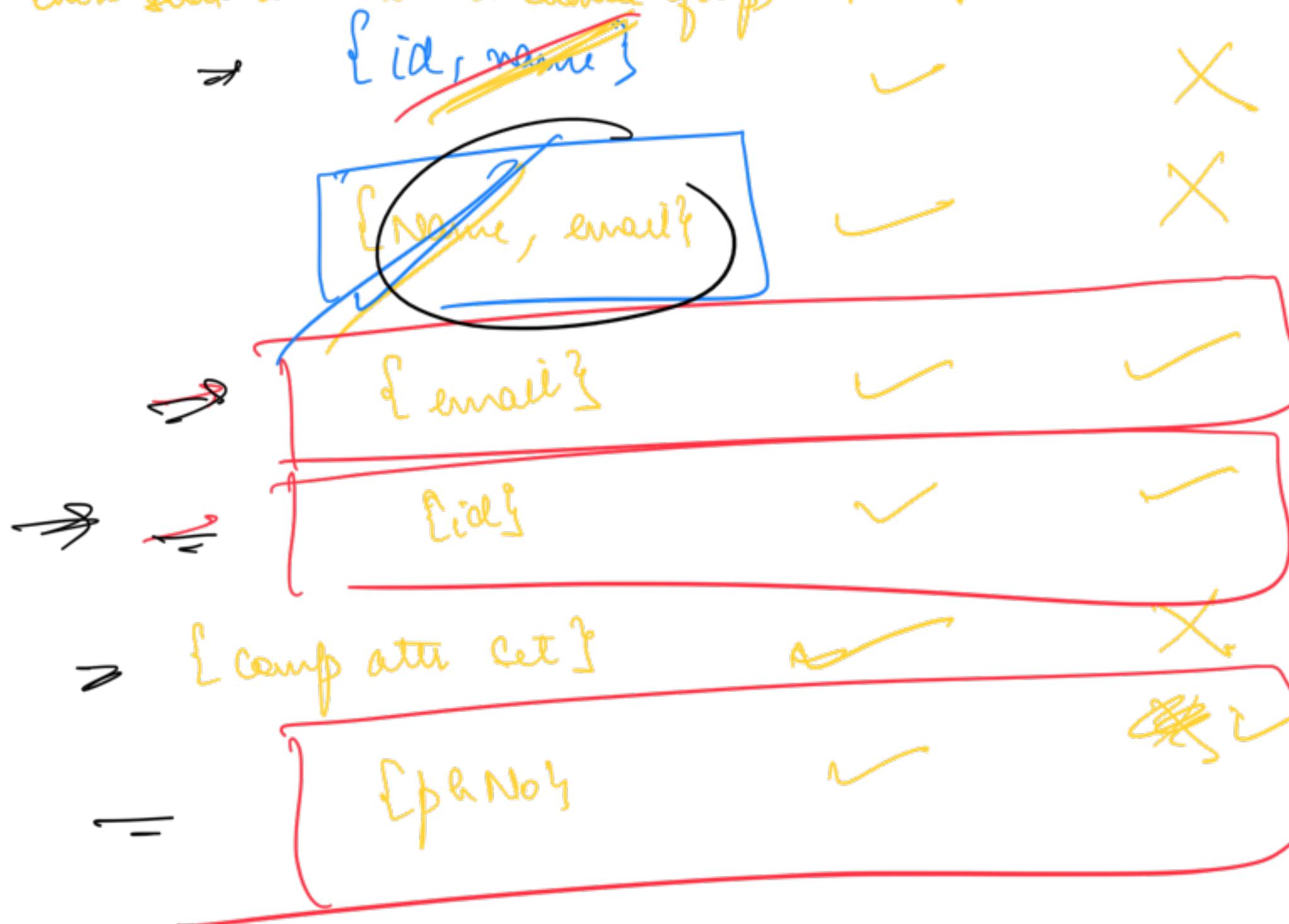
Candidate key

Any ^① subset of attribute where no 2 records
will have the same value and the size of

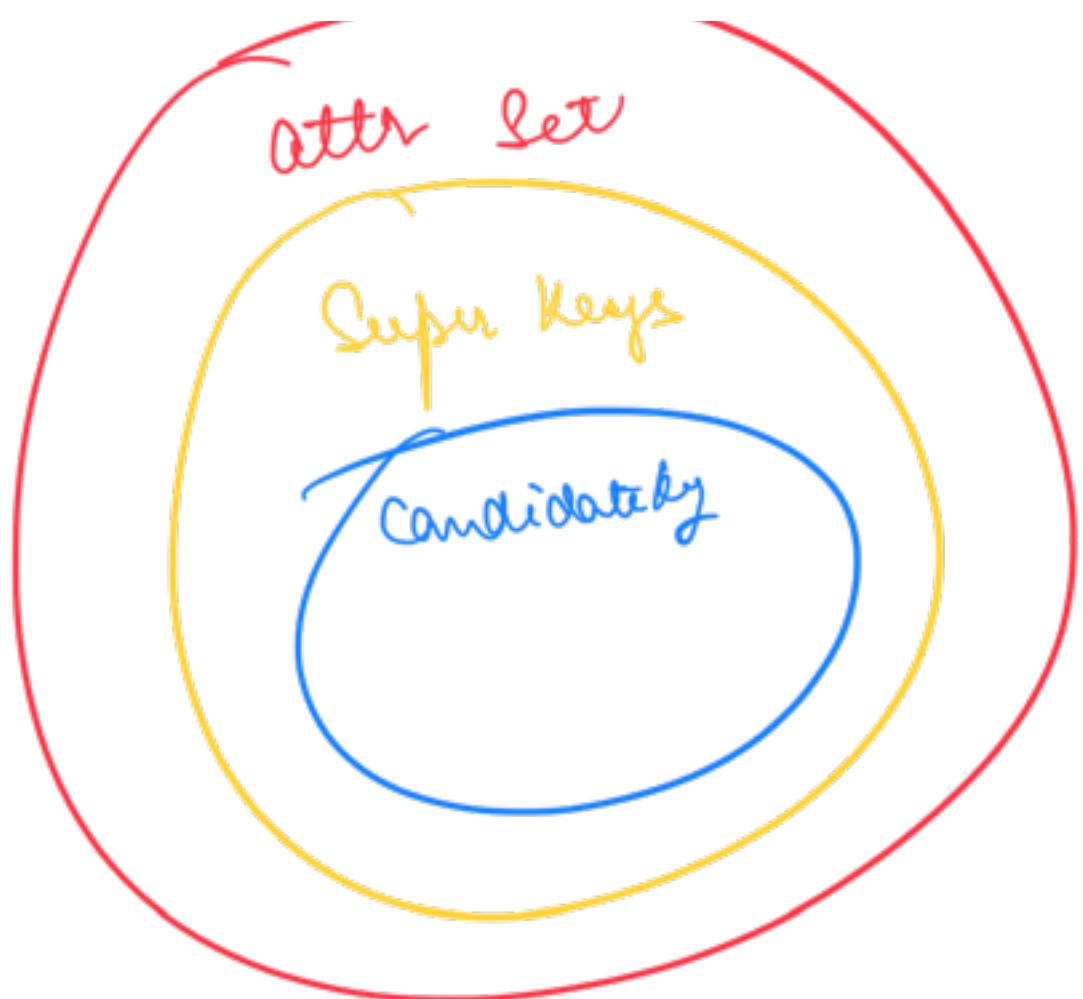
the subset is minimum : ^②

1st normal form \rightarrow no two attributes \rightarrow no ... \rightarrow no mistake.

(if you remove any of the column super key
that will be a chance of dup)



Super Key always contains a Candidate Key



Case Study : Scaler Student Mentor Session

I have a table to store info about
mentor session of Scaler Student

mentor - lesson

Student - id	Mentor - id	agenda	time	feedback

{ Student_id, Mentor_id, term }

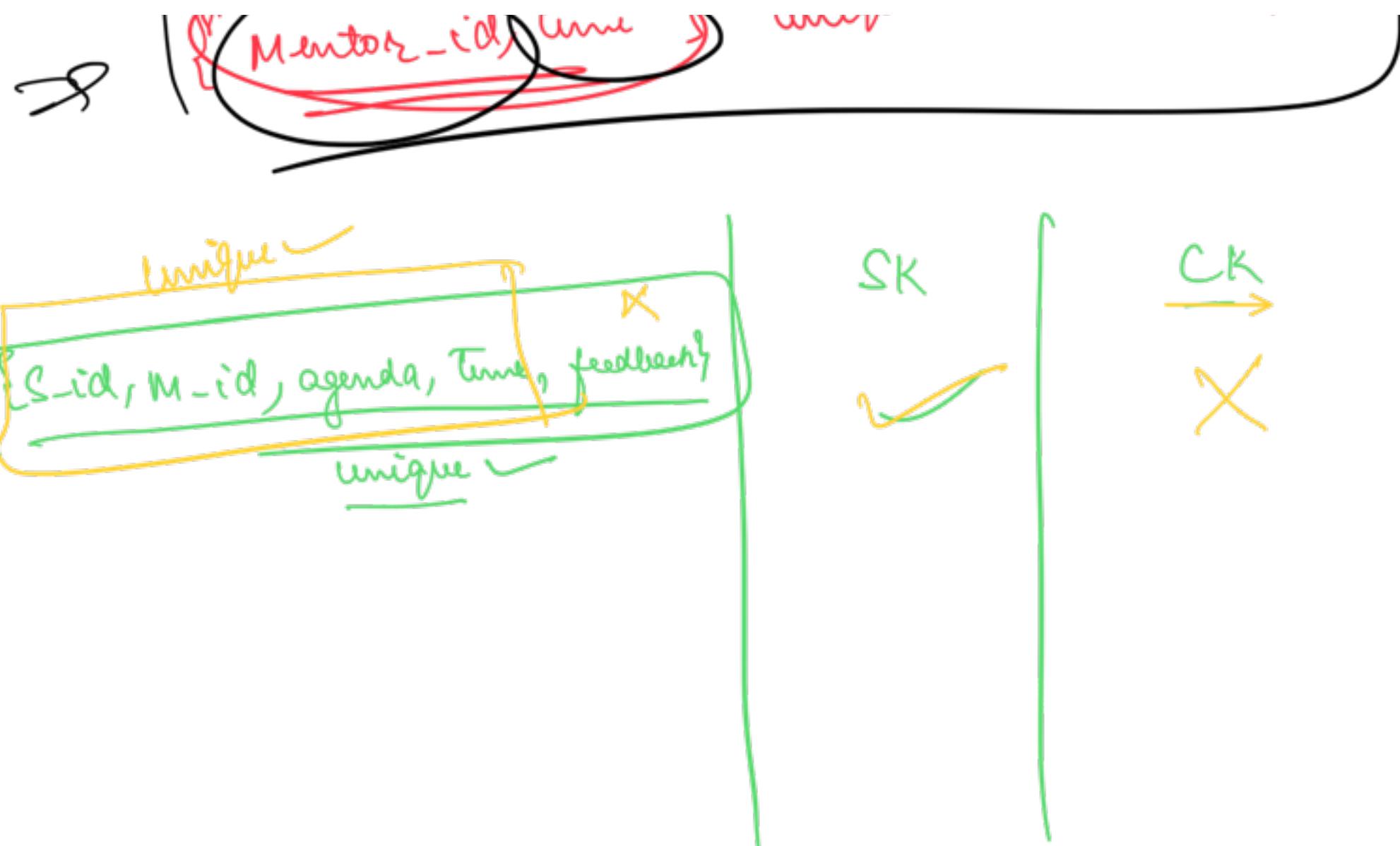
{ Mentor - id }

if I remove mentor_id
 $f = 1 - \frac{1}{n}$

{ Time } \rightarrow ~~single~~

{Mentor.id} } → critique

Unique 2-t



Super Key

Can there be any 2 rows in my DB

with the exact same value of all the
attributes in key

Yes \Rightarrow NOT A SUPER KEY

No \Rightarrow IT IS A SUPER KEY

Candidate Key

If I remove any 1 attr from the ct:

Can there be any 2 rows in my DB
with the exact same value of rem attr

\rightarrow NO \Rightarrow uniqueness ✓

\rightarrow Not a candidate key

else :

it is a Candidate Key

X

L-ID, M-ID, agenda, ...

Uniqueness \rightarrow

✓ CK

✗ CK

{ L-ID, M-ID, agenda, time }

Uniqueness \rightarrow

Primary key

\exists Any 1 of the Candidate keys

\rightarrow Only 1 primary key for a table

\rightarrow Can have more than 1 alt

Super keys

Candidate

Primary

Anyone who is
eligible for
job

keys
highest rating in
interview

Person we
hire

Job: uniquely identify a row

PK → CK

- ① We choose a short 4/S string
- ② one for whom comparisons are fast

Composite key

attribute

- Key of $\Sigma \subseteq \perp$
- where no single attr can uniquely identify.

Foreign Key

→ Handle relations between relations of w 2 tables.

- Primary key of another tables
- Basically linking 2 tables.

Key == Unique Key \rightarrow A set of attr that uniquely identify a row.

now

Composite Key

→ key that has more than 1 attr

→ where you can't uniquely identify a record with a single attribute

