



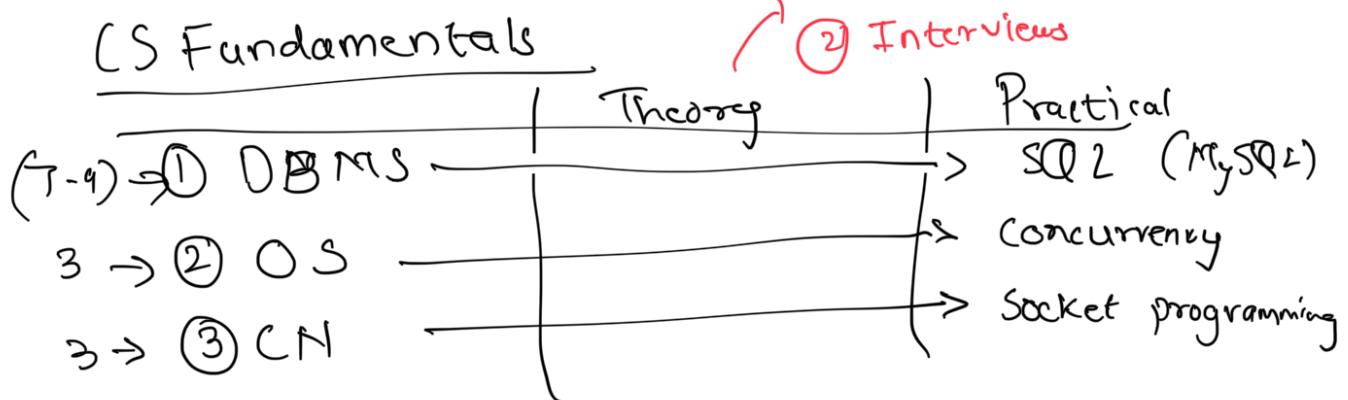
Bhavik Dand

Arto.io

Ex-Dunzo, CleverTap, Paytm

① Day to day job

② Interviews



### DBMS Curriculum

① Intro to DBMS & SQL

② Schema design

- SQL {
- ③ Joins
  - ④ Aggregate & Builtin Functions
  - \* ⑤ Indexing Transactions & ACID properties

Advanced {

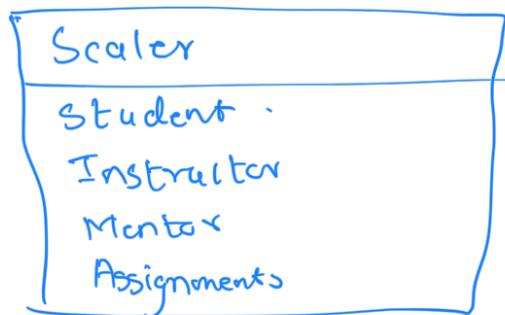
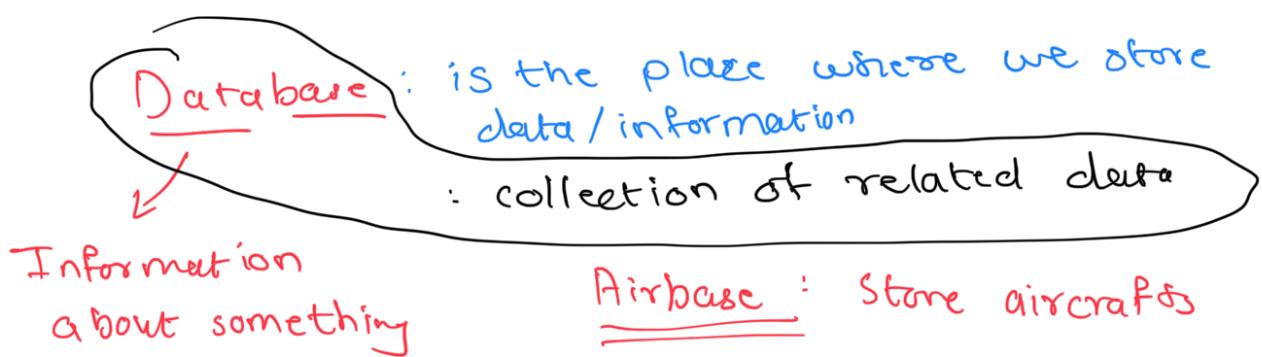
- ⑥ Subqueries & Views

SQL {

- ⑦ Procedure / Triggers & Events

Intro to DBMS & SQL

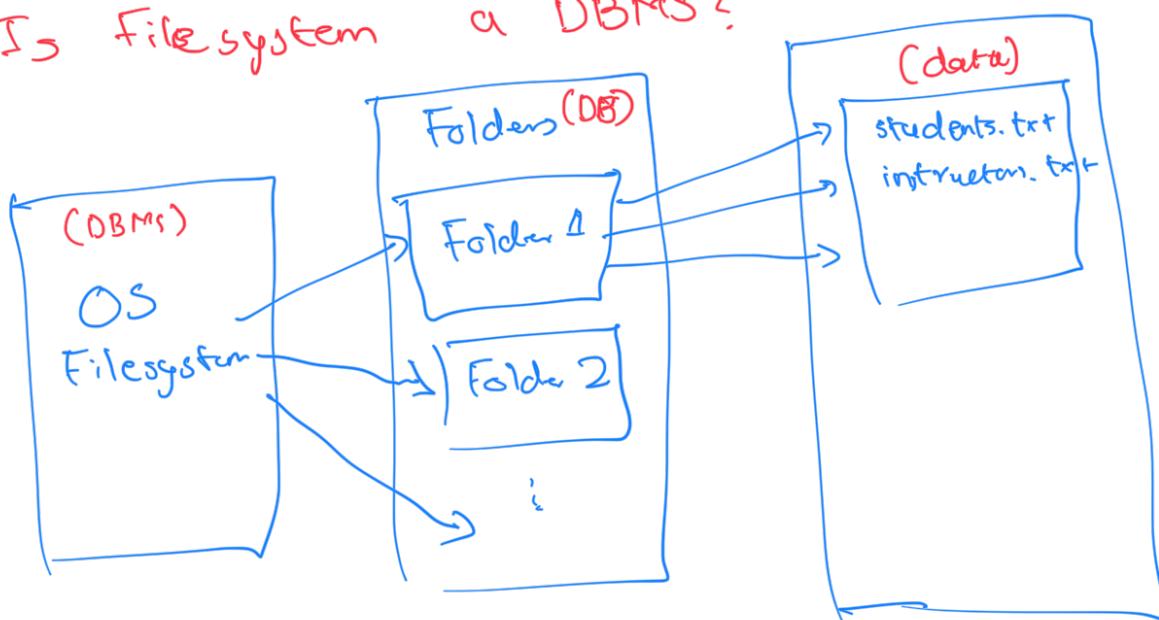
# DBMS: Database Management Systems



DBMS: is a software system which helps us to manage a database

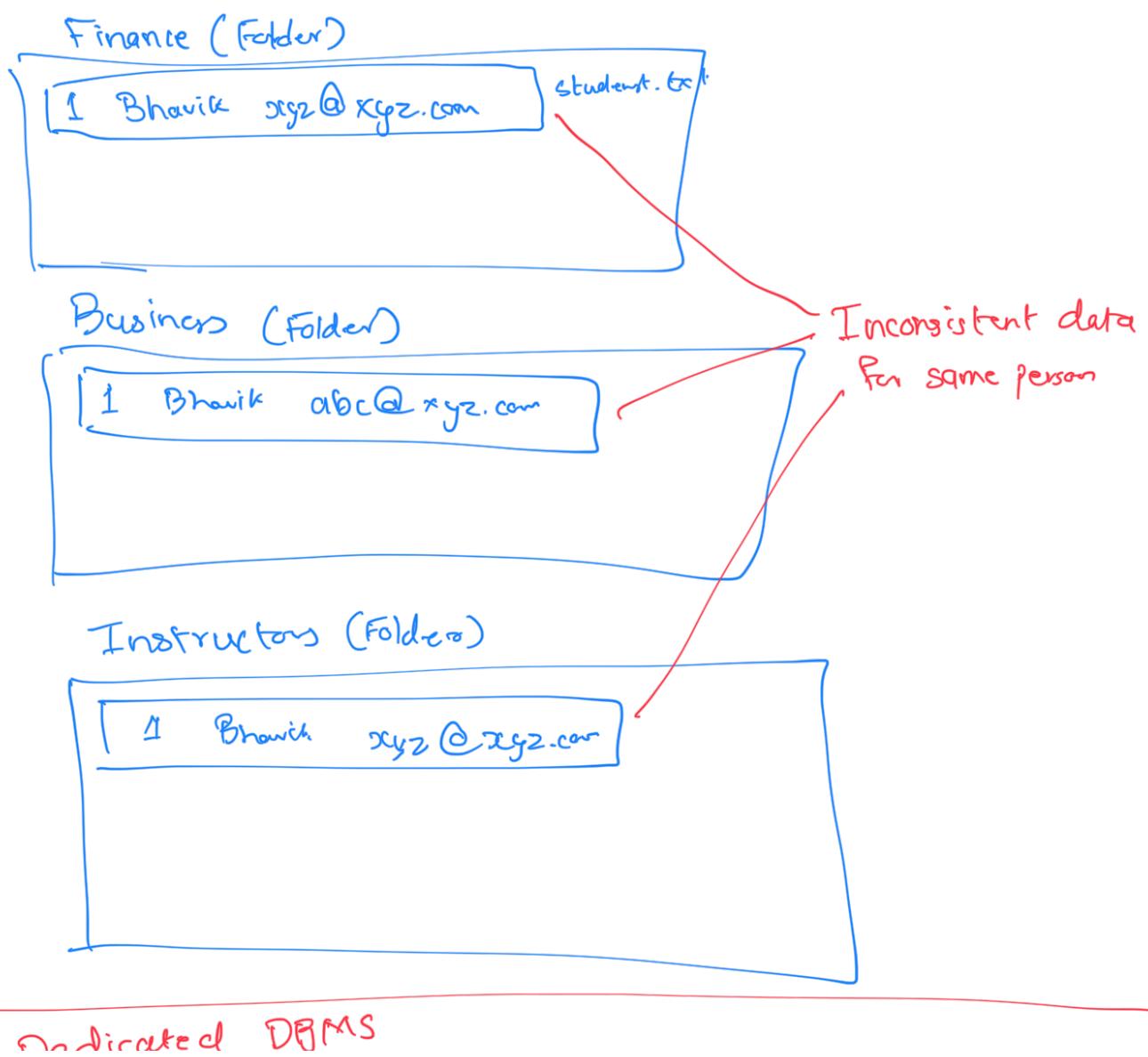
- ↓
- ① creating db
  - ② getting data from db
  - ③ updating db
  - ④ deleting db

Is Filesystem a DBMS?



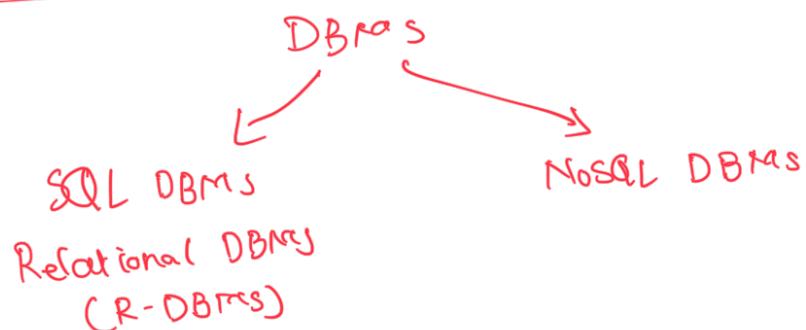
## Issues:

- ① Not fast
- ② Extra space
- ③ Unable to relate information
- ④ Querying is difficult (Aggregate operations X)
- ⑤ Security is a concern
- ⑥ Data recovery is not good enough
- ⑦ Redundancy of data
- ⑧ Concurrent access is not possible



QUESTION :-

- Pros:
- ① Backups
  - ② Querying is simple
  - ③ Concurrency ✓
  - ④ Efficient at storage & querying



## NoSQL DBMS (Not SQL) / (Not structured)

- ① Came into existence because of boom in scale of appln's
- ② Various usecases are supported by various type

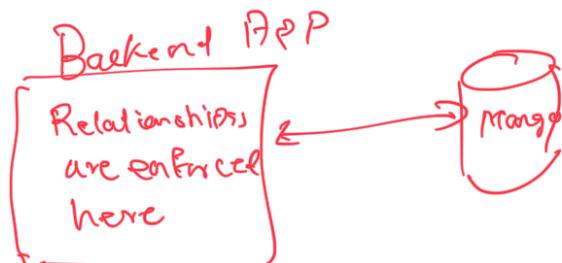
### Types

- ① Graph based DBs:  
Used to model relationships/network like data  
eg: Neo4j
- ② Columnar DB:  
Analytics workload  
eg: Cassandra/ Bigtable
- ③ Document DB:  
Stores plain unstructured objects like JSON  
eg: MongoDB

#### ④ Key Value Pair DB:

Similar to hashmap

Eg: Redis, DynamoDB



```
{ id: 1  
  name: Bhavik  
  friendsList: [2, 22, 23, 29, ..]
```

},

```
{ id: 22  
  name: "Bhanu"
```

3

---

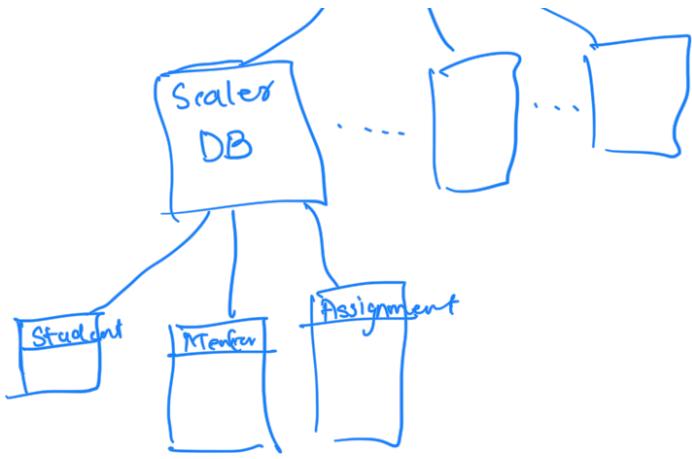
#### Relational DB

as a DB which is a collection of related relations

Relations: Table (rows & columns)

Scalar DB in MySQL





Entity: something which has existences  
can be living & a non living thing

Entities in scalar:

- ① Student
- ② Mentors
- ③ Instructors
- ④ Class
- ⑤ Mock interview
- ⑥ Assignments

Each entity becomes its own table in RDBMS

Relation means a table.

Break till 9:05 AM

Entity  $\equiv$  Relation  $\equiv$  Table  $\not\equiv$  Relationship

Terms:

① Relation: Table

② Attributes: Properties of a relation / table  
columns

③ Tuples : Row of a table

④ Degree : # of attributes of a relation  
eg: 5

⑤ Cardinality : # of tuples of a relation  
eg: 1

Student				
F-name	I-name	Email	Phone	City
Bhavik	Dand	bhavik@scaler.com	1234	Bangalore

Cell

Properties of a relation

① Every tuple must be unique

F-name	City
Bhavik	Bangalore

X

② Every cell in a relation must be atomic

(JSONs, lists, objects  
are not allowed in a cell)

single value

F-name	phone-num
Bhavik	123, 456

Breaks atomic

Exception: Recent releases of RDBMs have allowed support for lists & JSONs

③ Order of columns can differ

F-name	L-name	City	Email

select \* from students;

select F-name, Email from students;

④ Order of rows can also differ

select \* from students where name = "Bhavik"

200



456

1098

456  
200  
1098

Which math concept is R-DBMS inspired from?

R-DBMS → Set Theory

1. ...  $\cap$   $\cup$  ... 2. ...  $\cap$   $\cup$  ...

Set  $\{H, D\}$  三 set  $\{D, H\}$

$$\text{List}[A, B] \neq \text{List}[B, A]$$

Relation as a set of tuples  
(rows)  
(Table)

$$\text{Set } \{A, A, B, C\} \equiv \text{Set } \{A, B, C\}$$

How do we identify a row uniquely?

st_id	f_name	l_name	email	phone_num
1				
2				
3				
.				
.				
.				

Key: Set of attributes which uniquely identify  
a row

Composite Keys	Super Key?	Candidate Key?	Primary
f-name	X	X	X
f-name, l-name	X	X	X
f-name, l-name email	✓	X	X
f-name, l-name, address	✓	X	X

email, phone	✓	✓	✓	✗
email	✓		✓	✗
phone-num	✓		✓	✗
st-id	✓		✓	✓

Types of Keys:

- ① Super Key: set of attributes which uniquely identify a row
- ② Candidate key: set of attrs of minimum size which can uniquely identify a row
- ③ Primary Key: is 1 of the candidate keys as chosen by the DB owner / creator / architect

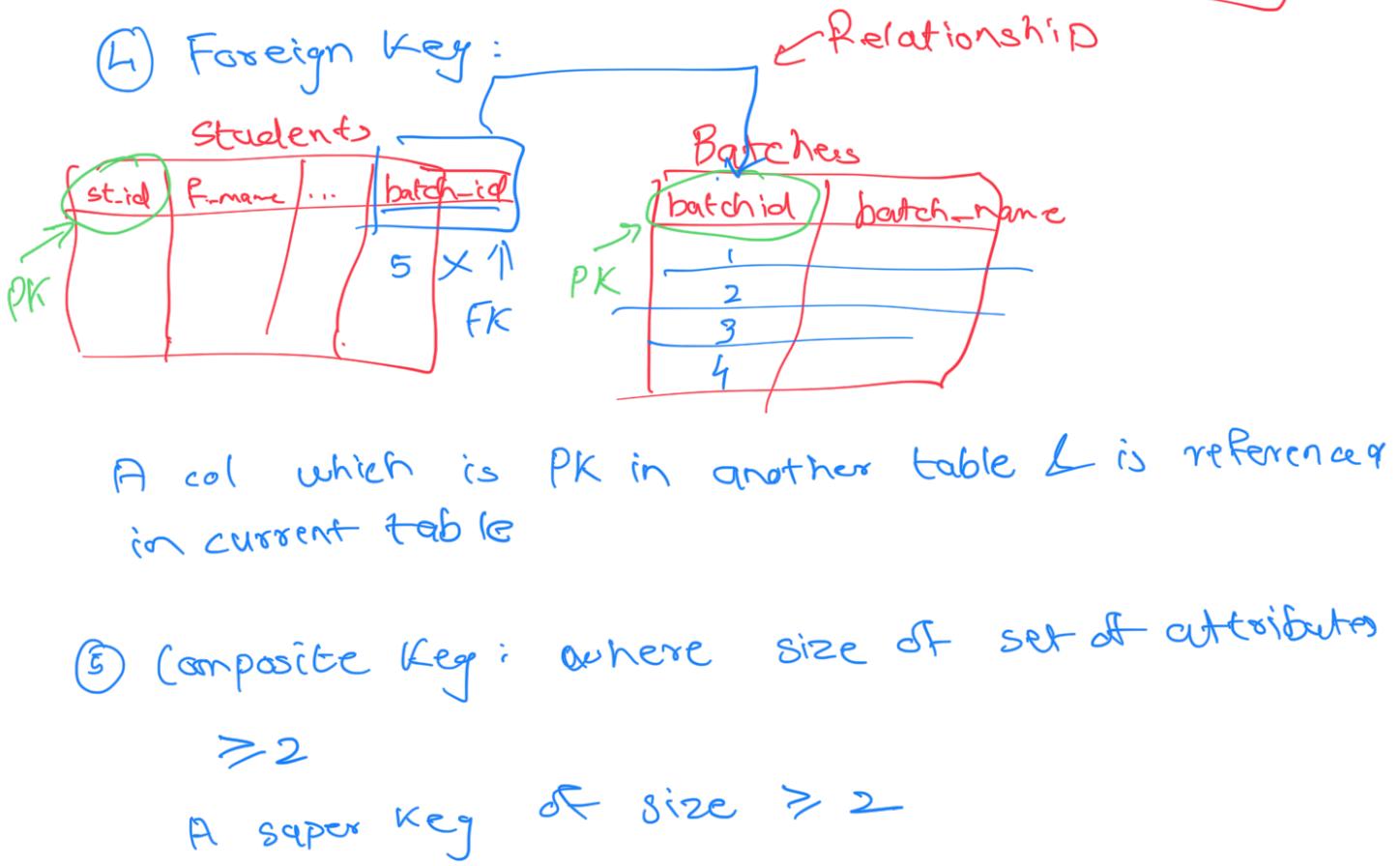
student-mentor

st-email	mentor-email

Issue with email as PK:

- ① Email can change
- ② More memory is required for storage

Do not use unique entity attributes as PK as it can change in future



Break till 10:00 AM