Class starting at T+10 mins.
↳ Class time.

Features of a dedicated DBMS.

1) Data Backups
2) Concurrency
3) Security → Encryption
4) Efficient storage and Retrieval of data

Non-Relational DBMS. (NoSQL DBMS)

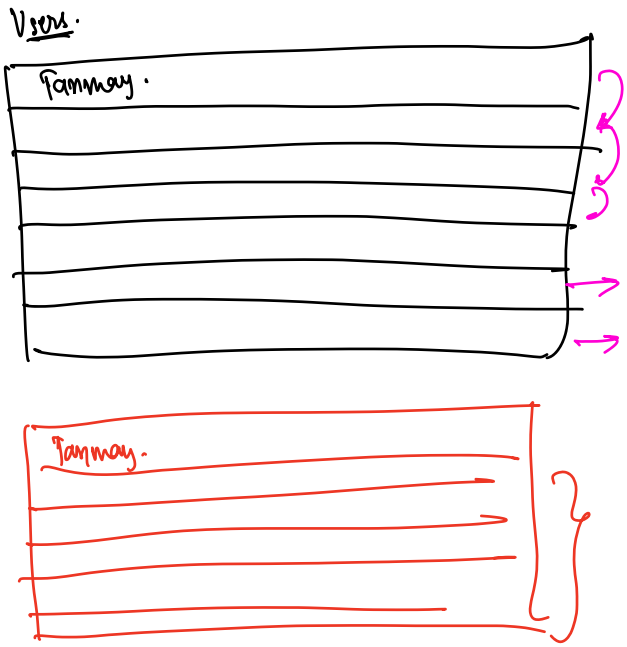
→ Don't follow relational model of storing data.
↳ Data is not stored as rows.

Students Table.

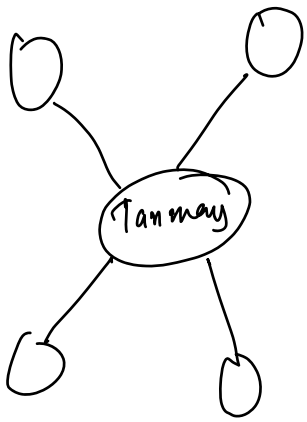
1
2
3
4

} Rows in this table.

facebook friends.



Tanmay: [4, 45, 1001, 10005, 1006]



Types of NOSQL DBMS.

1) Graph Based (Neo4J)
2) Columnar (Cassandra.)
3) Document Type (MongoDB, elastic search, Amazon Document DB, etc.).
4) Key-value pairs. (Redis)

Relational Databases. → Most widely used model.

→ Relational Model is the model behind MySQL.

↓

Relational Model represents data as collection of relations

Database:- Collection of data.

Relation:- Nothing but a table of values.

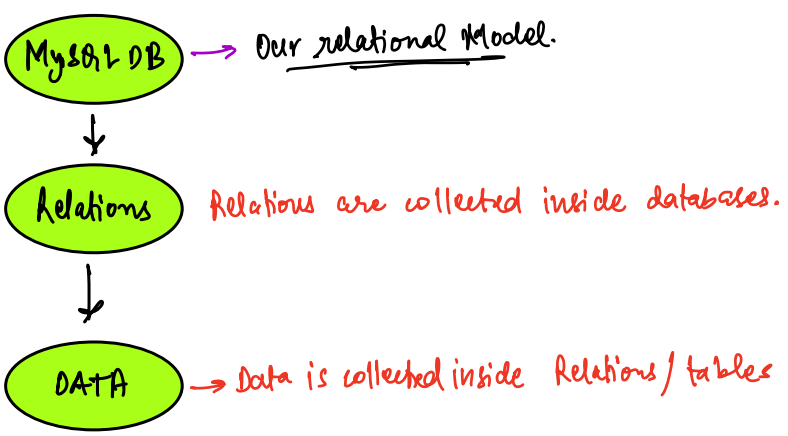
Students table:-

id	first_name	Last_name
1	Alok	Singh.
2	Subba	Reddy.
	~	

Entity

Instructors table

~	
~	
~	
~	



Terms :-

1) Relation :- Table.
2) Attributes of Relation :- Columns of a table.
3) Tuple :- Row in a table.
   (set of values)
   (1, Tanmay, Upadhayay)
4) Degree of a table / Relation :- No. of attributes of a relation
   No. of columns of a table
5) Cardinality of a table / Relation :- No. of tuples in a relation
   No of rows in a table.

Properties of relations / tables.

1) Every row should be unique (one column should have different value)

Students table :-

id	first_name	Last_name	PSP
1	Sumit	Kumar	90
2	Sumit	Kumar.	90

2) Values in cells should be atomic
   ↓ only single values.
   No list / JSON / collection.

First_Name	Last_Name	Phone Numbers
Alok	Singh	[8115..., 8505...] → storing it as a string is a hack.

✗

Ideally,

Students table:-

id	first_name	last_name	PSP
1	Sumit	Kumar	90
2	Sushil	Sridhar	90

Phone Numbers:-

id	stu-id.	Phone No..	
1	1	8195910235	
2	1	8195910337	
3	2	8505 - -	
4	2	8505 - -	

stu-id is a foreign key

3) Order of columns should not matter.

Select fname, psp, lastname from Students;

4) Order of rows should not matter.

e.g. Select * from students where name = 'Alok'

Row 1
Row 2
Row 3
1st

Row 2
Row 1
Row 3
2nd

SET THEORY

Relation = Set = Table.

Set {A, B} = Set {B, A}

List [A, B] ≠ List [B, A]

Every tuple is a value inside a set

Table = Set { ( ), ( ), ( ), ( ), ( ) ... }

↳ The order of these tuples / rows in this set/ table does not matter

Cardinality of a set = # of values in a set

Cardinality of a table = # rows in a table.

Tell me,

$\{A, A, B, C\} = \{C, B, A\}$?

↓

✗ Not possible.

Identify tuples in a Relation / row in a table.

Students Table.

	First_name	Last_name	Email	Phone
	Sumit	Kumar	~	~
	Sushil	Sridhar	~	~

Key:- Attributes / set of attributes that can uniquely identify a row in a relation.

Attributes	Key / Superkey	Candidate key
f_name	✗	✗
l_name	✗	✗
(f_name, l_name)	✗	✗
(f_name, l_name, e_mail)	✓	✗
(f_name, l_name, e_mail, phone)	✓	✗
email	✓	✓
phone_no.	✓	✓
(email, phone)	✓	✗

Types of Keys.

1) Super key :- A set of attributes that can uniquely identify a row.

2) Candidate key :- A key of min. size that can uniquely identify a row.
   ↓
   Candidate key is a super key of min. size.

3) Primary key :- One of the candidate key that is chosen by the Database Architect as the key of the table.

Few problems with choosing email / phone no. as PK.

1) It can change
2) It will be updated in mapping / referring tables where storing long string isn't very ideal.

Ideally, a PK should never change

Create another column called student id.
   ↓
   Super key, candidate key, PK.

4) Composite key :- Key with ≥ 2 attributes.

5) Foreign key :-

(9:22) Resuming.