

deepak.kasera@scaler.com.
7015608331

* SQL-DBMS

1. Intro to SQL & DBMS (Relational)

MySQL

2. Schema Design.

3. Data Types

4. Normalization

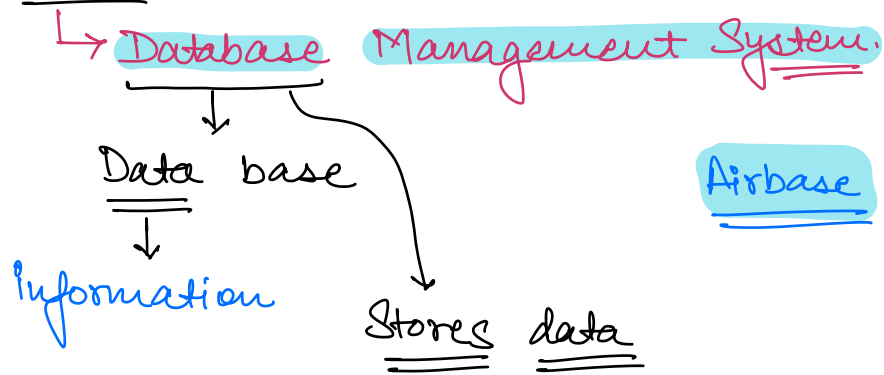
5. Indexing

6. Transactions (ACID)

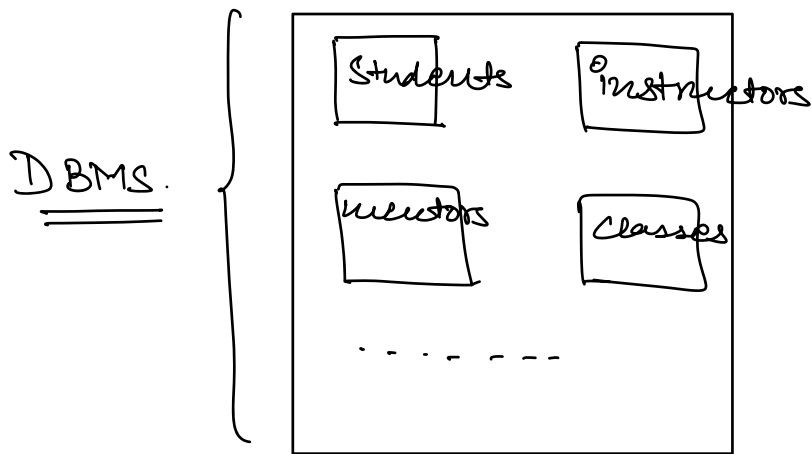
7. Aggregate Functions

8. Views

* DBMS



DBMS : Place to store organised/interrelated data.



Ex Students Leaderboard

- ↳ Organised data
- less query time.

Q. Why do we need DBMS ?

* File System.

Students.txt

name	email	ph-no	batch	...
Akshay	abc@xyz	1234	May22	Beg Monday

Instructors.txt

--

batches.txt

May22 Beg Monday	—	—	—	—	—
-----------------------------	---	---	---	---	---

Issues with File System:

1. Inconsistency → because of ^(duplication) redundancy.
2. No relation
3. Search will take longer time.
4. Slow
5. Security
6. Backup.

DBMS:

↳ Solve all the above issues.

→ Relational DBMS (SQL DBMS)
↳ MySQL, MS-SQL, Oracle, Postgress

→ Non Relational DBMS (NoSQL)

⇒ HLD

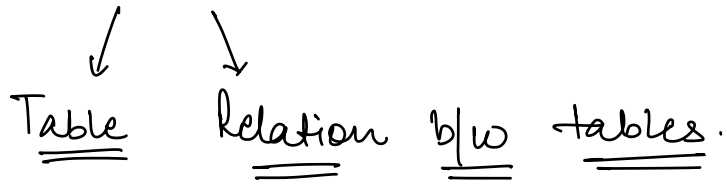
↳ Mongo DB

↳ Cassandra

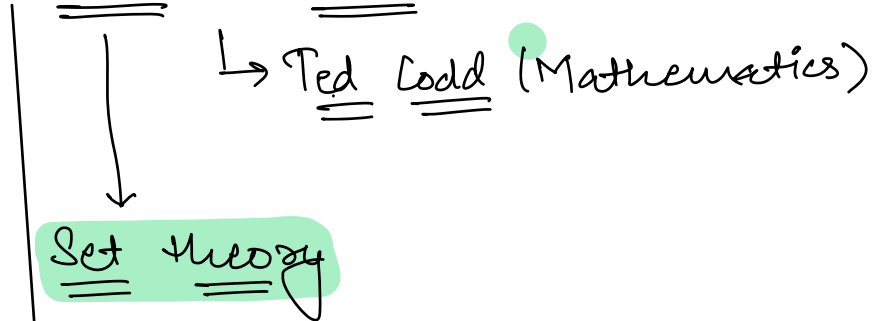
↳ Redis

↳ Graph based ⇒ Neo4J.

* Relational DBMS

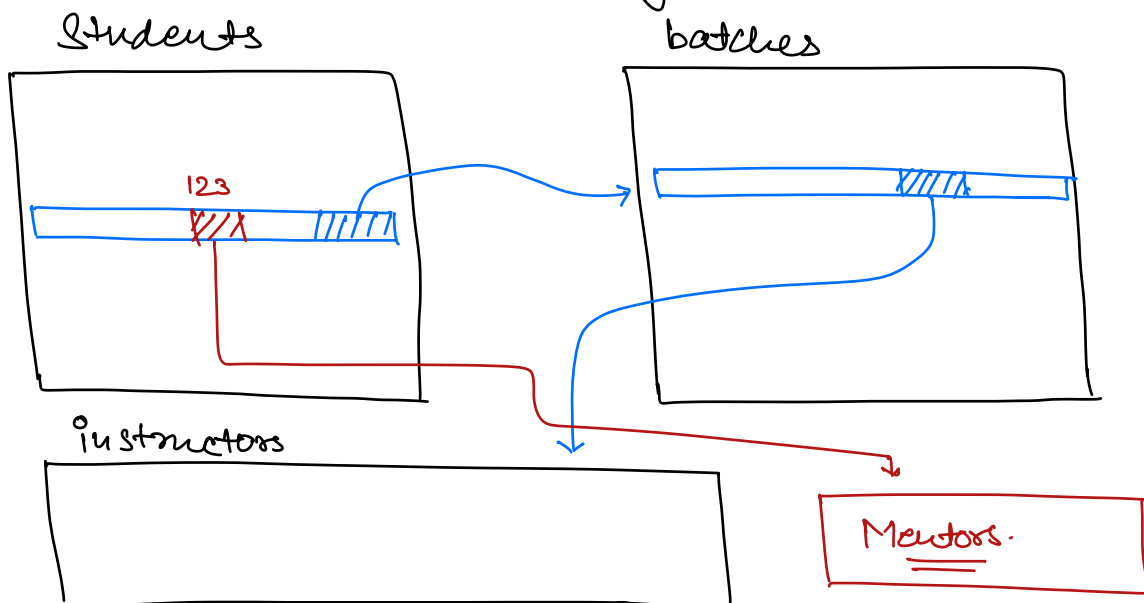


Relational Database



→ Collection of interrelated data stored in the form of relations (table).

⇒ Collection of relations (tables), Each table will have a set of tuples (row) & each tuple can be related to a tuple of some other table.



* Set Theory

1. Table \equiv Set of tuples.

↓
No duplicates.

\Rightarrow No two tuples (rows) will be completely same in a table.

[1 1 2 3]

[1 2 3]

↓

[1 2 3]

↓

[2 1 3]

\Rightarrow Order of tuples doesn't matter.

2. Order of column's doesn't matter.

f-name, l-name, <u>Phno</u> , email, batch
--

Note :

f-name, l-name, email, batch \Rightarrow |

Select f-name, l-name,
email, batch - - -

--

* Order of the columns should never be assumed.

3: Value in a cell should be Atomic.

↓ Which can't be divided further.
Single.

(Array / List ---)
x x

name	email	ph-no's
xyz	—	[—, —, —, —]

MySQL8 ⇒ JSON

⇒ SQL: Structured Query Language.

MySQL: DBMS

How to identify a row uniquely in a table.

Students

f-name, l-name, email, ph-no.

Key : Attribute / Set of attributes to uniquely identify a row.

Super Key

Attribute(s)	Super Key	<u>CK</u>	PK
f-name	X	X	X
f-name, l-name	X	X	X
f-name, l-name, email	✓	X	X
f-name, email	✓	X	X
email, ph-no	✓	X	X
email	✓	✓	○
ph-no	✓	✓	○

Candidate Key

↳ Super key of Minimum size

⇒ If we remove an attribute from super key then the remaining set of attributes is NO longer a SK. = Candidate key.

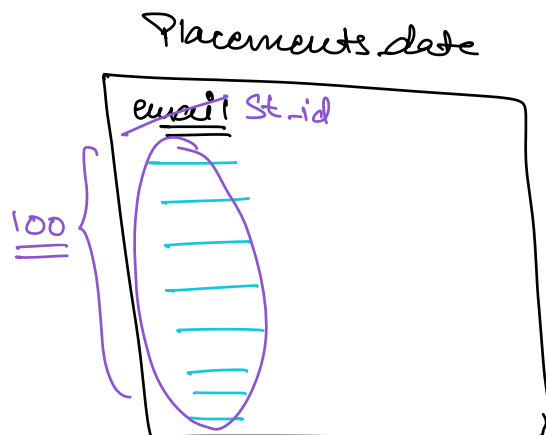
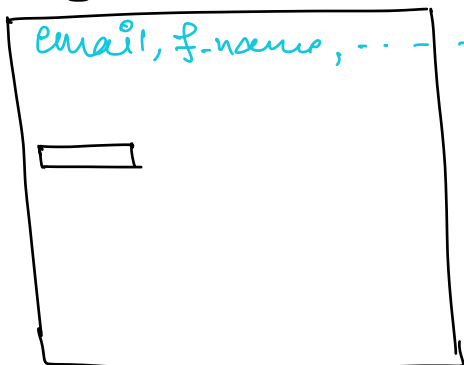
⇒ If we have any redundant attributes in SK then it won't be a CK.

If there are NO redundant attributes in SK then it is a CK.

Every Candidate key is a S.K but NOT vice-versa.

Primary Key: Any candidate chosen by DB owner.

email → PK
Students



Issues

1. Email id can change.

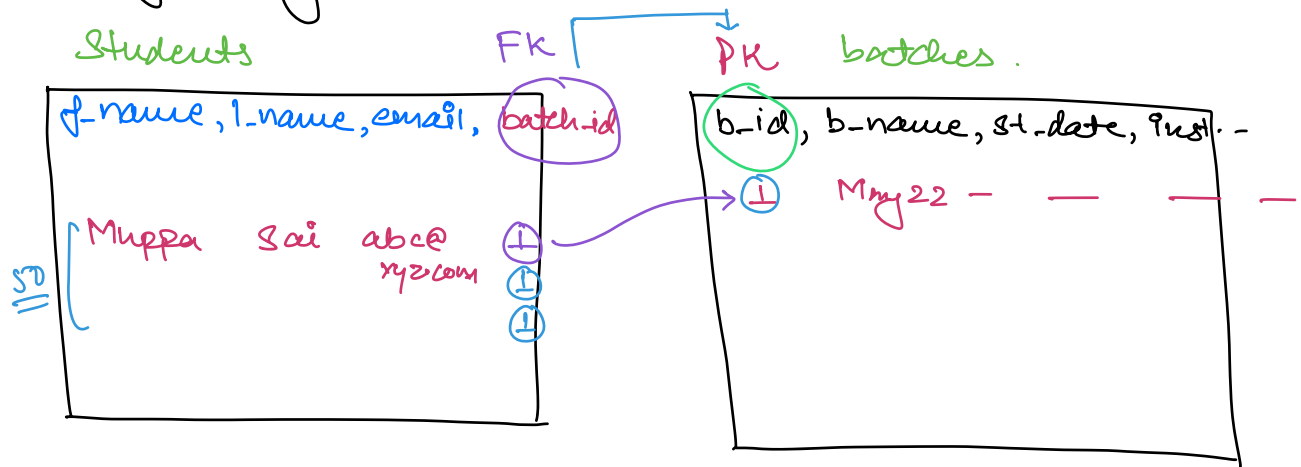
2. Space

* Any kind of user info shouldn't be treated as PK, because it can change over the time.

⇒ Add a new column ⇒ Student-id.

(AUTO INCREMENT)

Foreign Key



⇒ batch-id in students table is referring to batch-id in batches table.

⇒ FK

Ex

st-id	exam-id	marks
1	1	70
1	2	80
1	3	60
2	1	80
2	2	90
2	3	92

st-id \Rightarrow SK \times

exam-id \Rightarrow SK \times

(st-id, exam-id) \Rightarrow SK \checkmark

CK \checkmark

CK \times

