

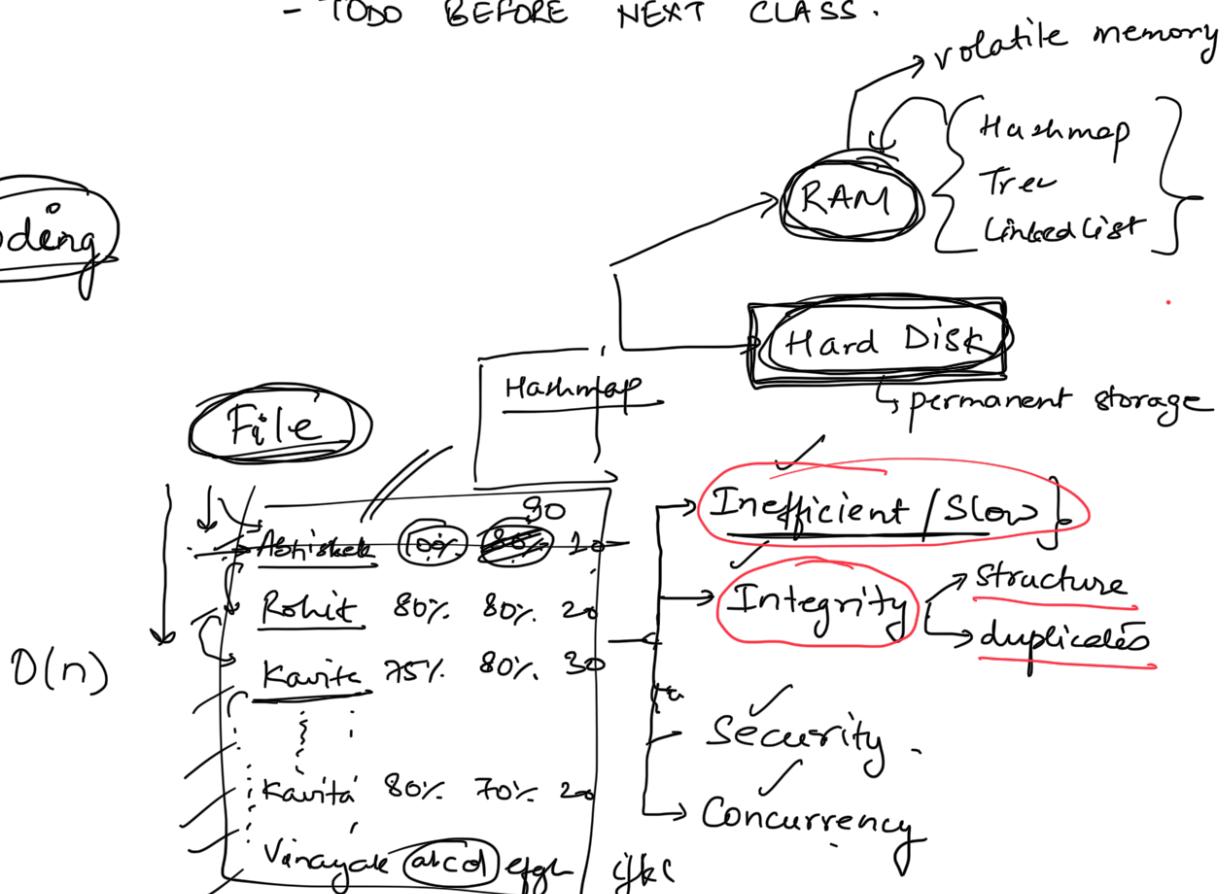
DATABASES & SQL - I (INTRO)

AGENDA :

- WHY?
- CONTEXT
- TYPES OF DATABASE
- RELATIONAL DB:
 - KEYS

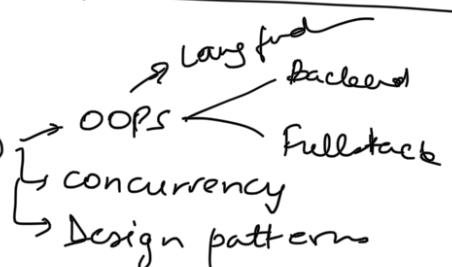
- TODO BEFORE NEXT CLASS.

Coding



Database

① 9 classes (SQL)

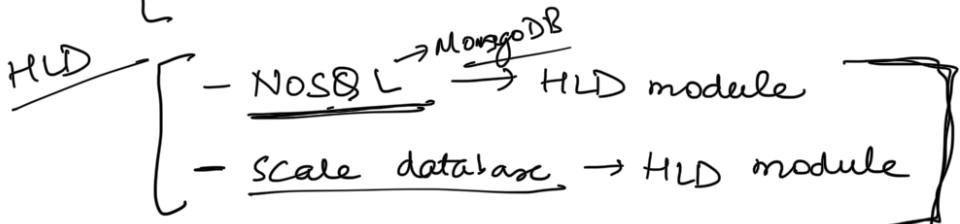


RAM → 8GB

HDD → 1TB

② Post DB & SQL,

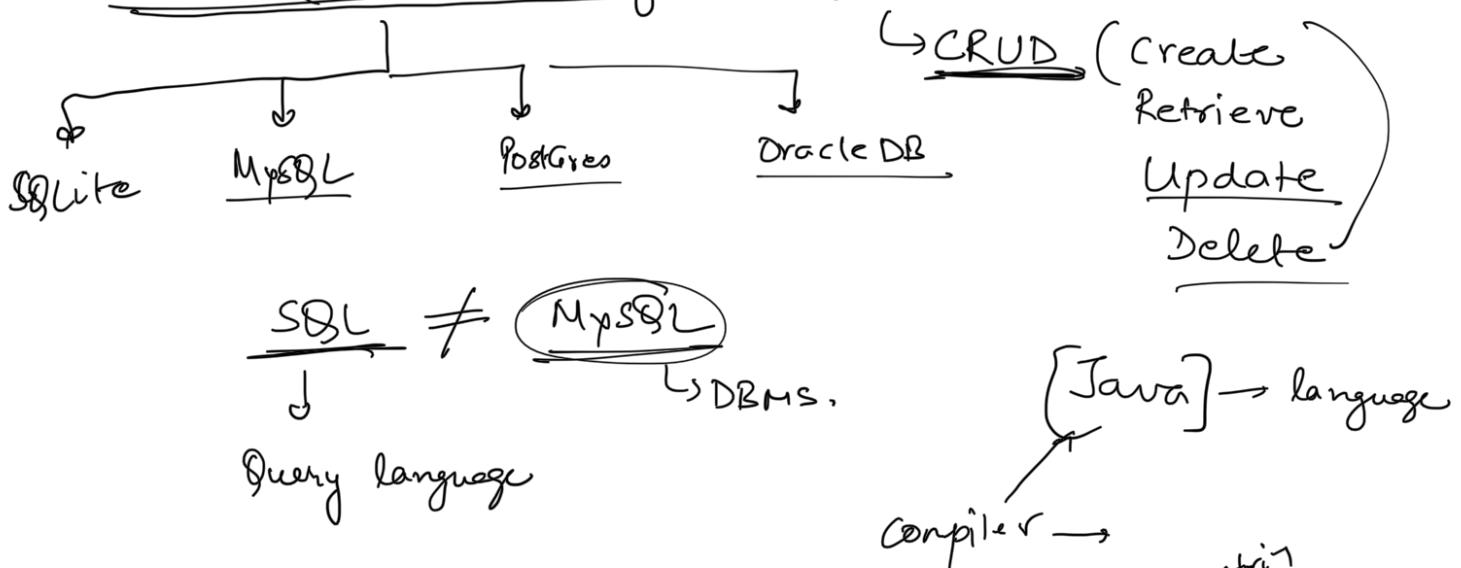
- DB
- Index in a DB
- SQL → complicated
 - Query optimisations
- Schema design
- Transactions



③ 1st class → No code writing.

Database → collection of related data

DBMS (Database Management System):



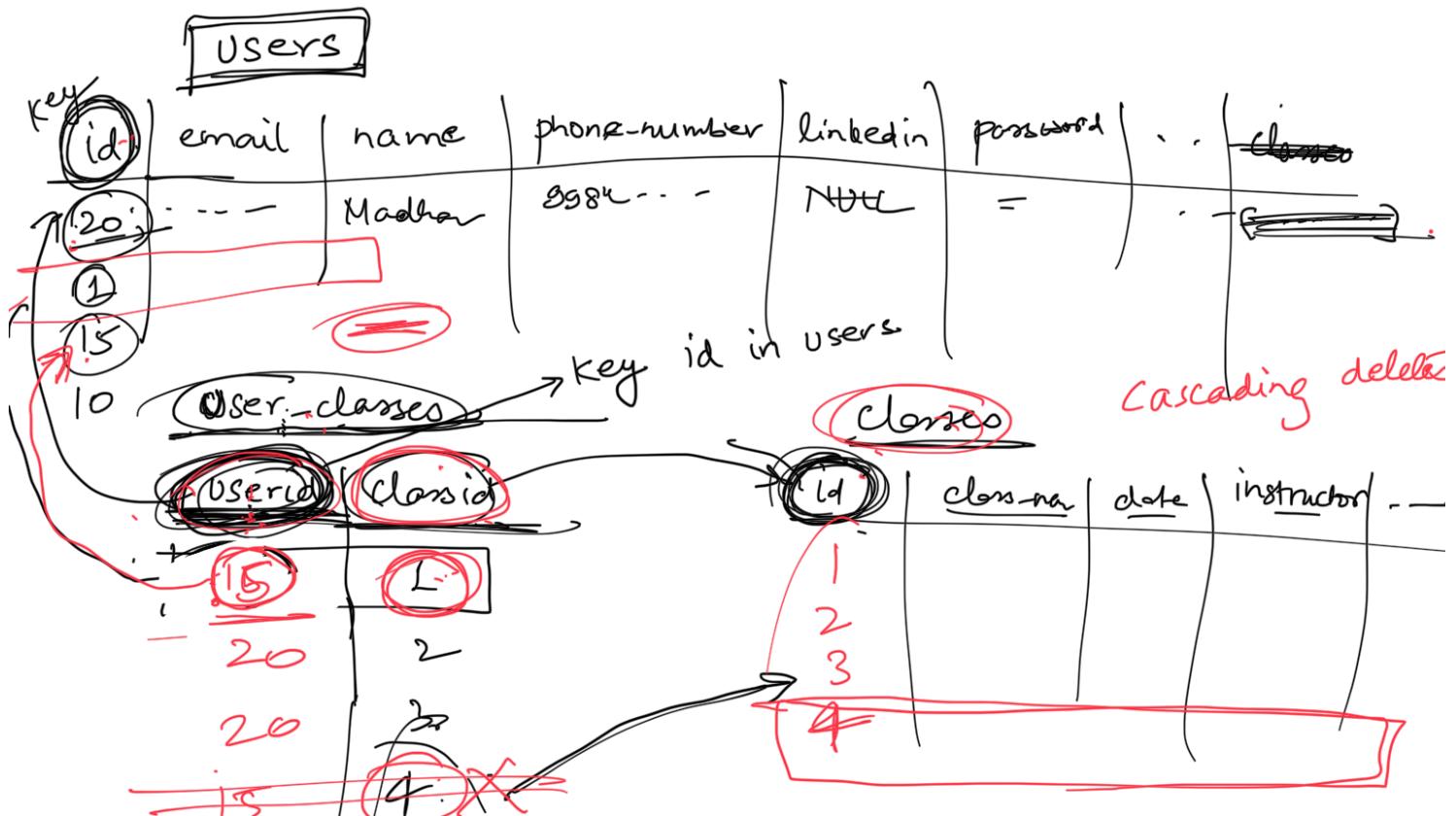
① Relational Databases

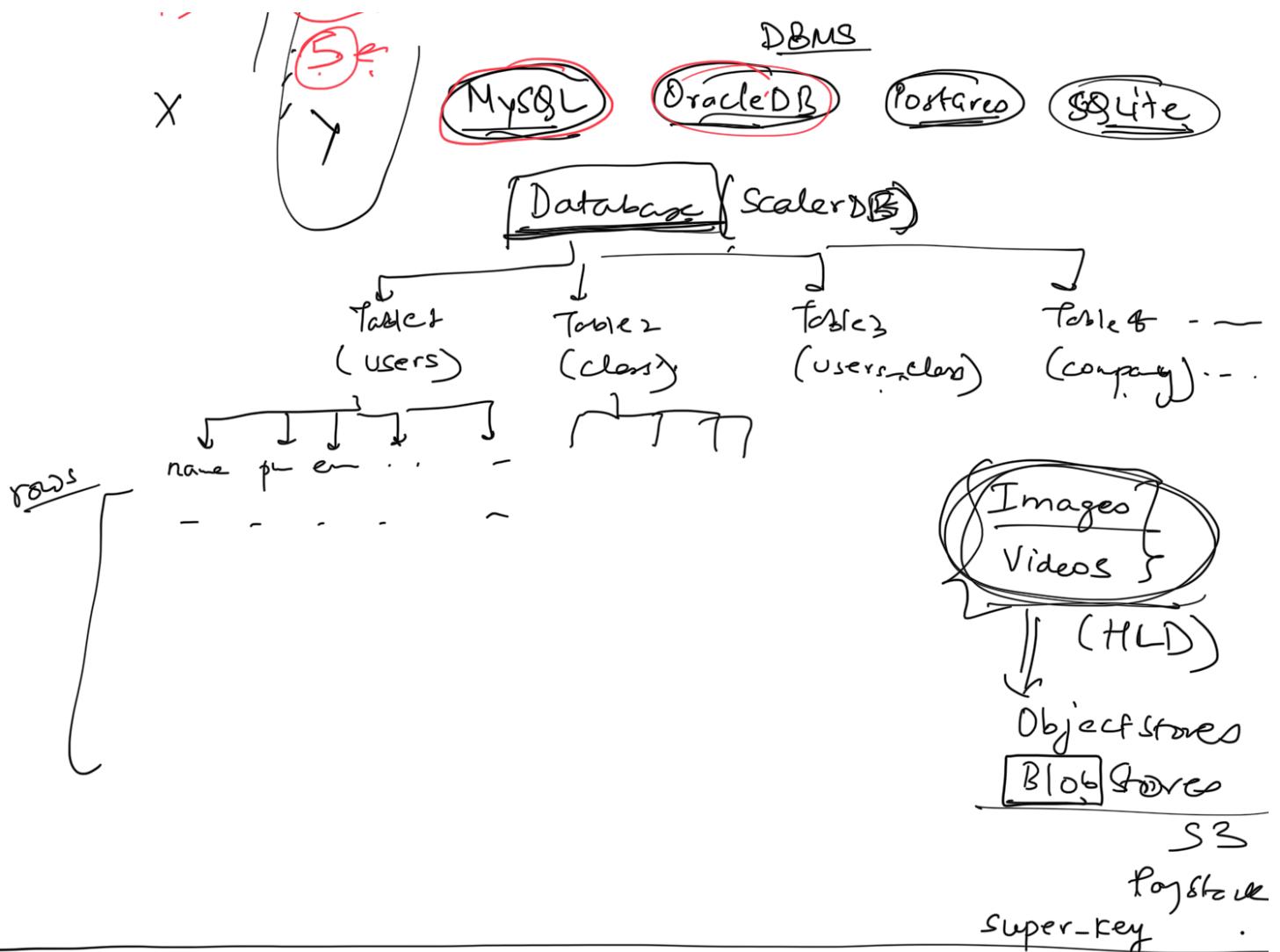
Table

Name	Email	Tl.No	Addr	PSP
Abc1	abc1@abc.com	9876543210	123 Main St	High
Abc2	abc2@abc.com	9876543211	456 Elm St	Medium

② NoSQL databases

- ↗ Key-value
 [json → file (document DB)
 XML]
- GraphDB
- ① Each column should have a type
 ② No 2 rows should be identical
 ③ Values in a row+col should be Atomic
- phone-numberL
 Ph-No1 Ph-Nb2
 ↓
 phone-number
 [Ph-No1, Ph-No2]
- row1 == row2
 if and only if
 entry[1][0] = entry[2][0]
 AND
 entry[1][p1] = entry[2][p2]
 AND
 ;
 ④ Title of a column has to be unique





Keys in Relational DB :

~~(Name, Attendance, email)~~

Student		Name	Ph No	Attendance	PSP	Subject	Year	email
id	string	Name	string	double	double	integer		
abc123@gmail.com	Ashish							

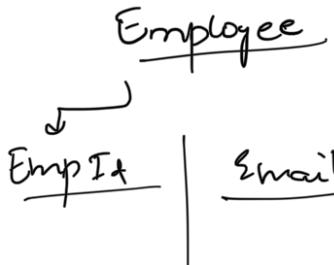
(Email, Name) → unique → key Super_key

(Email, Name, phone-number) → Superkey

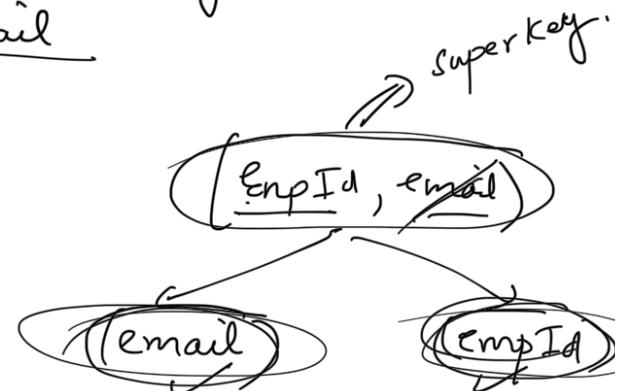
Name → X a key

(email) → Key

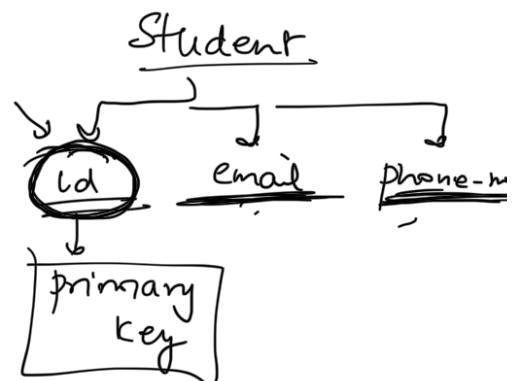
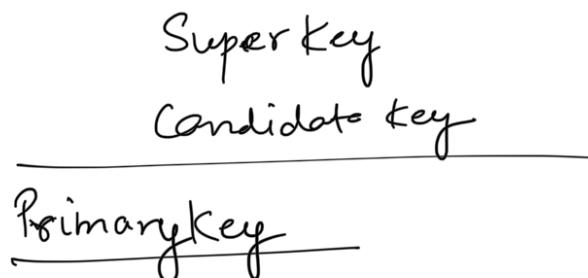
Key / Superkey



Candidate-key
email



Both b and c



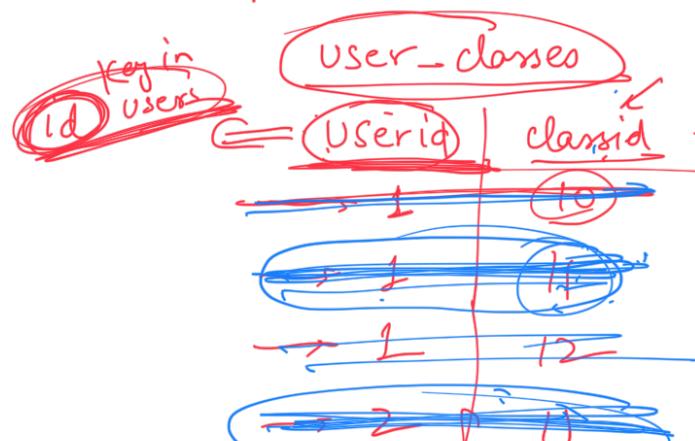
Foreign Key

Users

<u>(id)</u>	<u>name</u>	<u>email</u>
1	Abhishek	a@b.c
2	Naman	b@b.c
3	Abhishek	c@b.com

classes

<u>(id)</u>	<u>name</u>	<u>desc</u>	<u>inst</u>
10	sorting	=	=
11	string	=	=
12	searching	=	=



Cascading deletion
Abhishek

12

	2	12
	3	12
<i>primary key</i>		X
<i>Users</i>		
<i>id</i>	name email class-id class-name desc.	
1	Aboish ab@l.u 10 Sorting --	
1	Abo ab@l.u 11 Sorting-2 --	
1	- - - - 12 Ec --	

