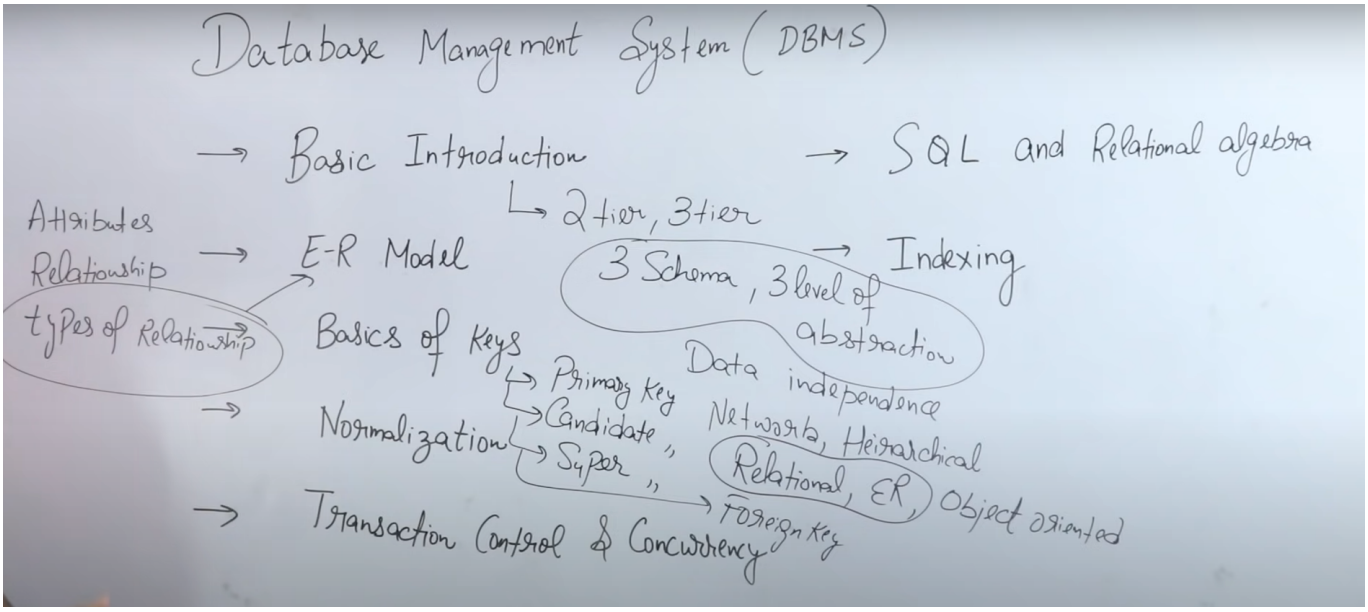
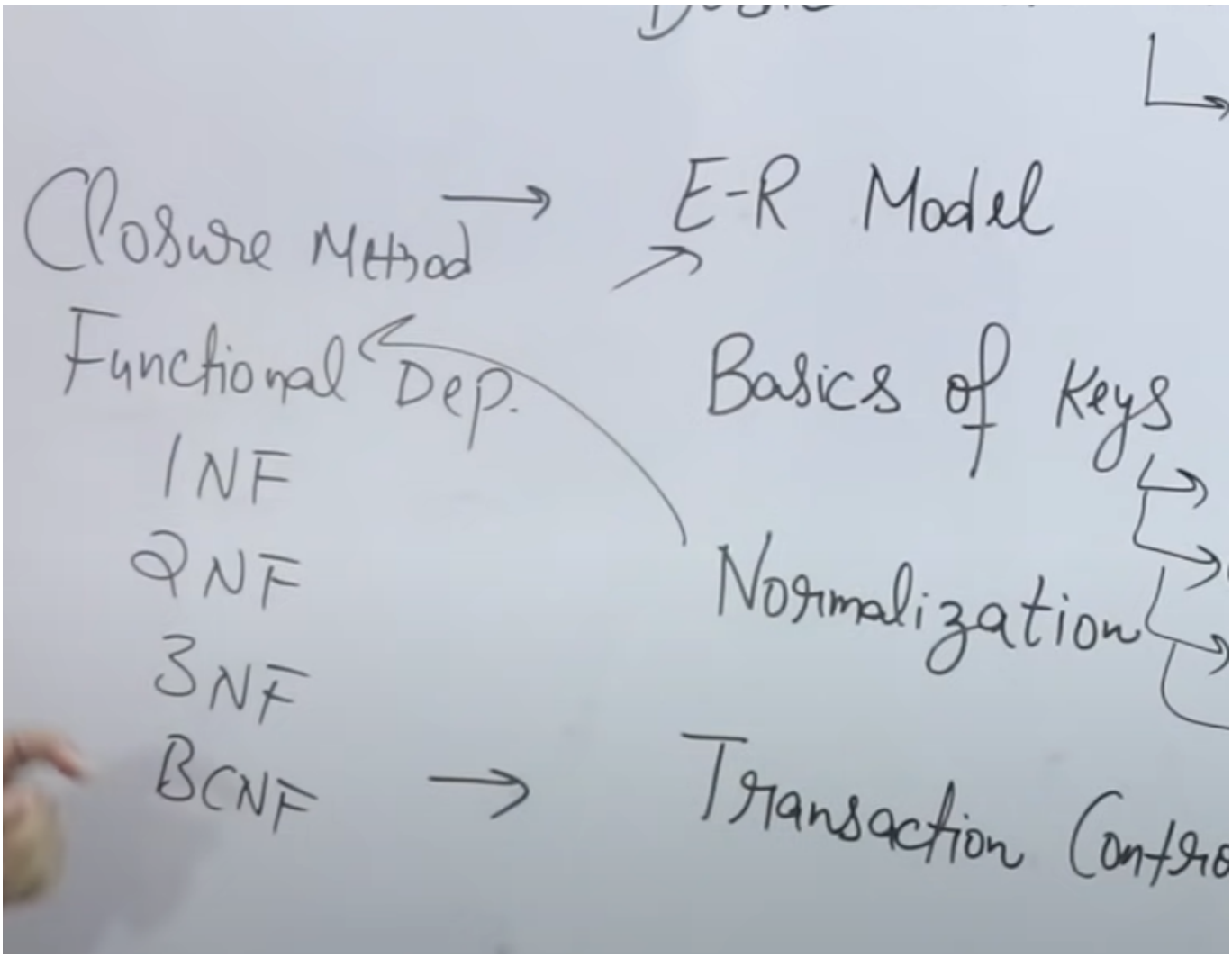
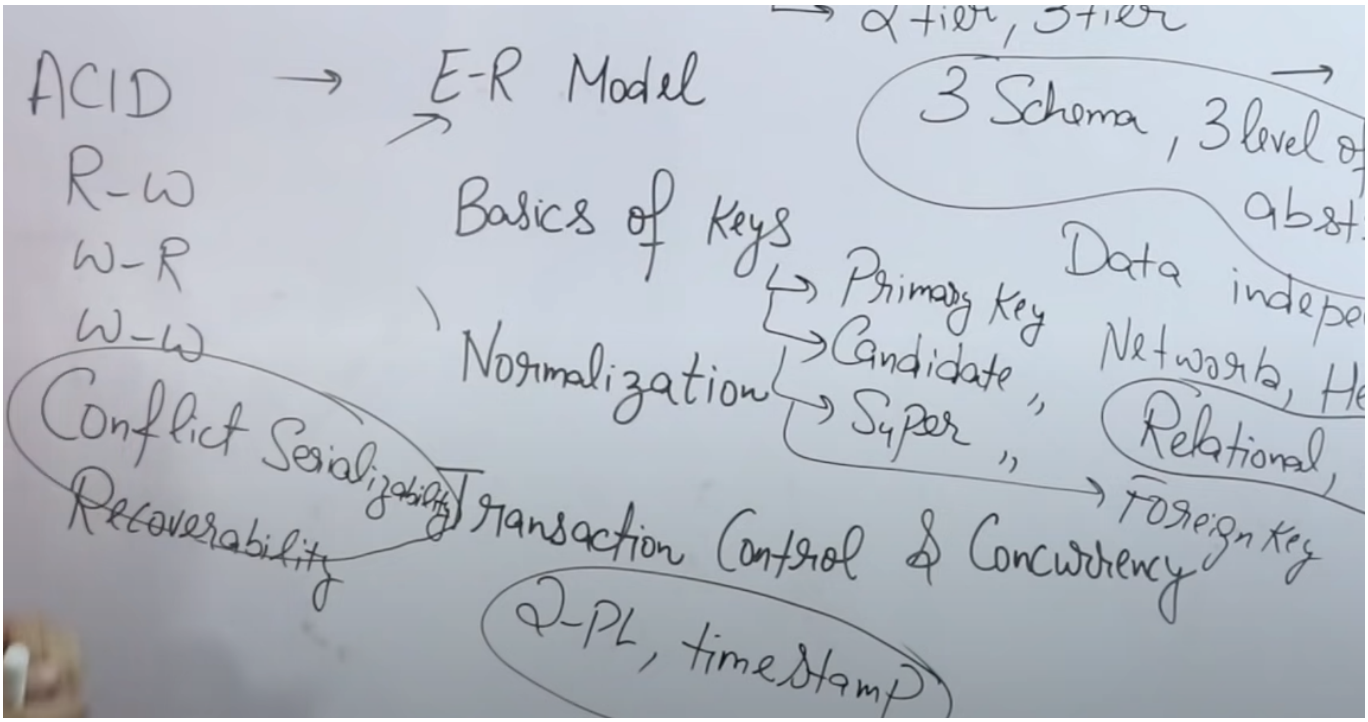
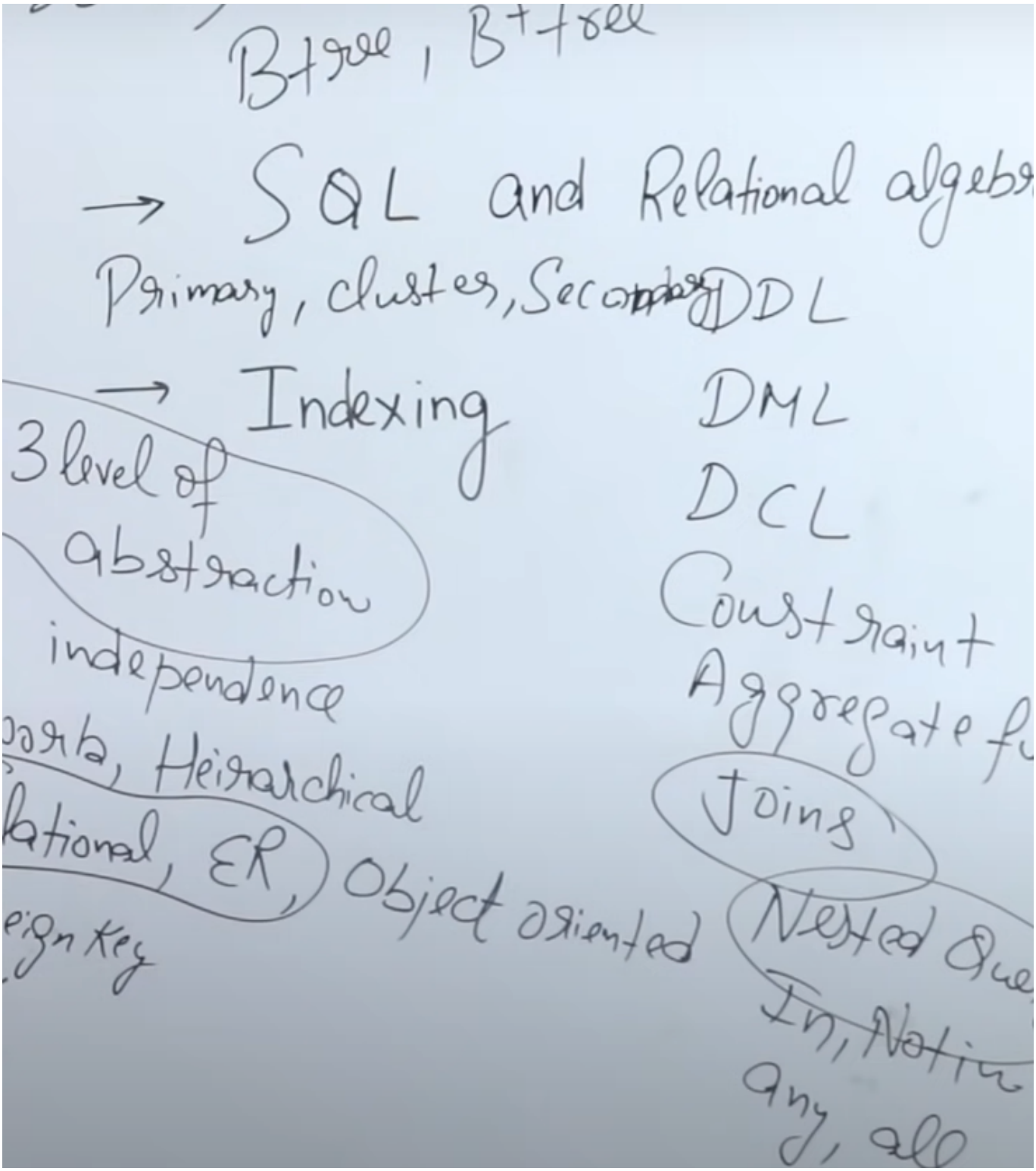
DBMS

Syllabus

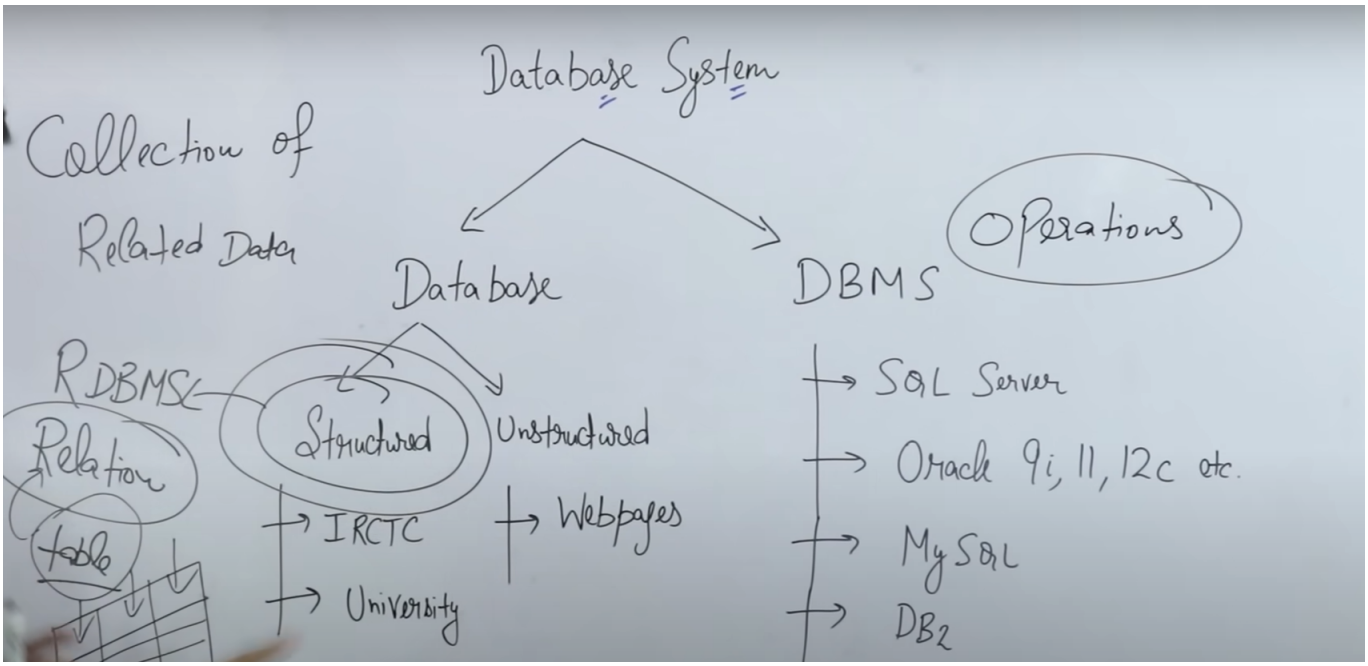





ACID
E-R Model
3 Schema, 3 level of
abst
R-W
W-R
W-W
Basics of Keys
Primary Key
Candidate "
Super "
Foreign Key
Data indepe
Networks, He
Relational,
Normalization
Conflict Serializab
Recoverability
Transaction Control & Concurrency
2-PL, timeStamp

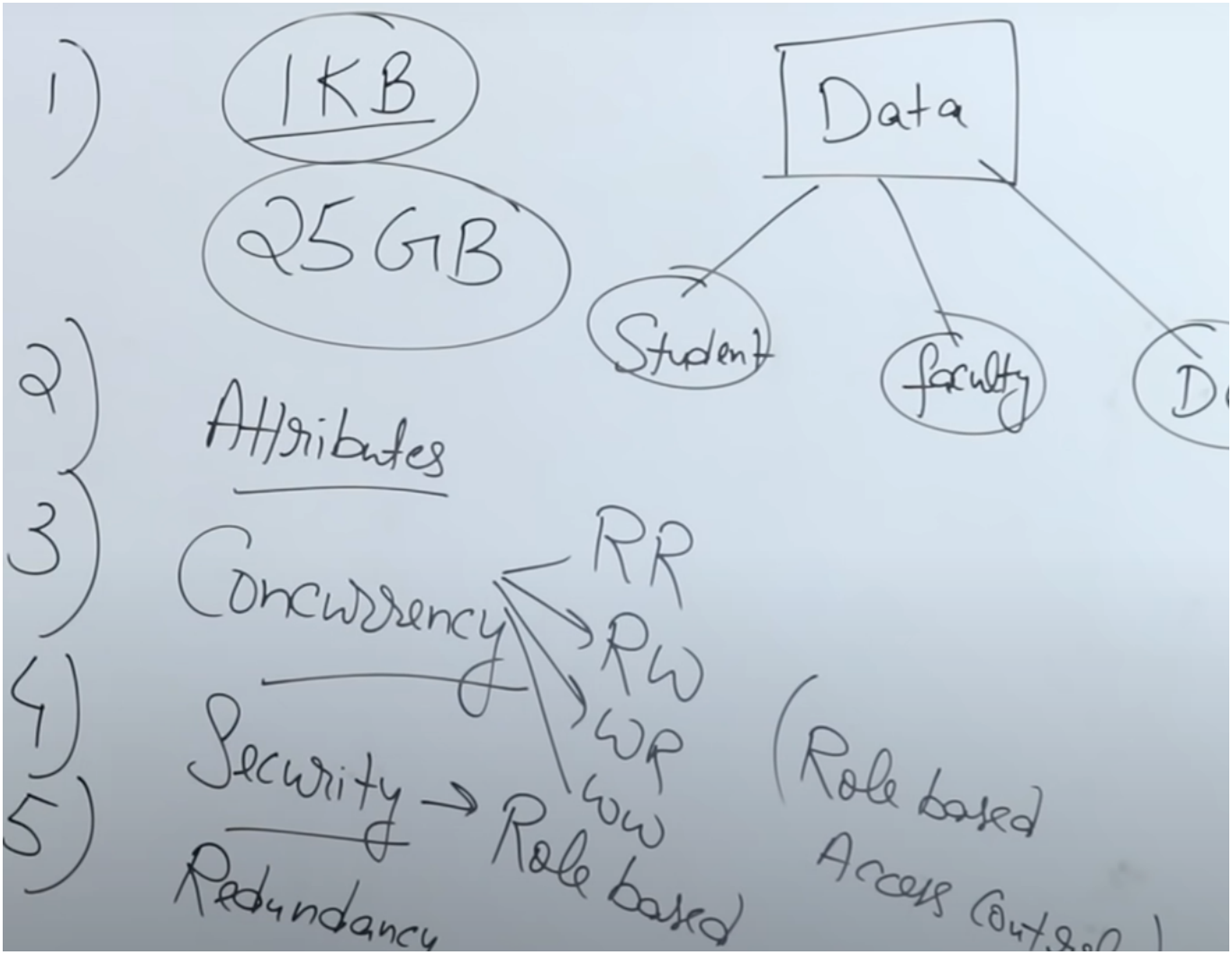

B tree, B+ tree
→ SQL and Relational algebra
Primary, cluster, Secondary
→ Indexing
DDL
DML
DCL
Constraint
Aggregate fu
3 level of abstraction
independence
Joins
Nested Que
In, Notin
any, all
Hierarchical
Object oriented
ER,
ign key

DBMS

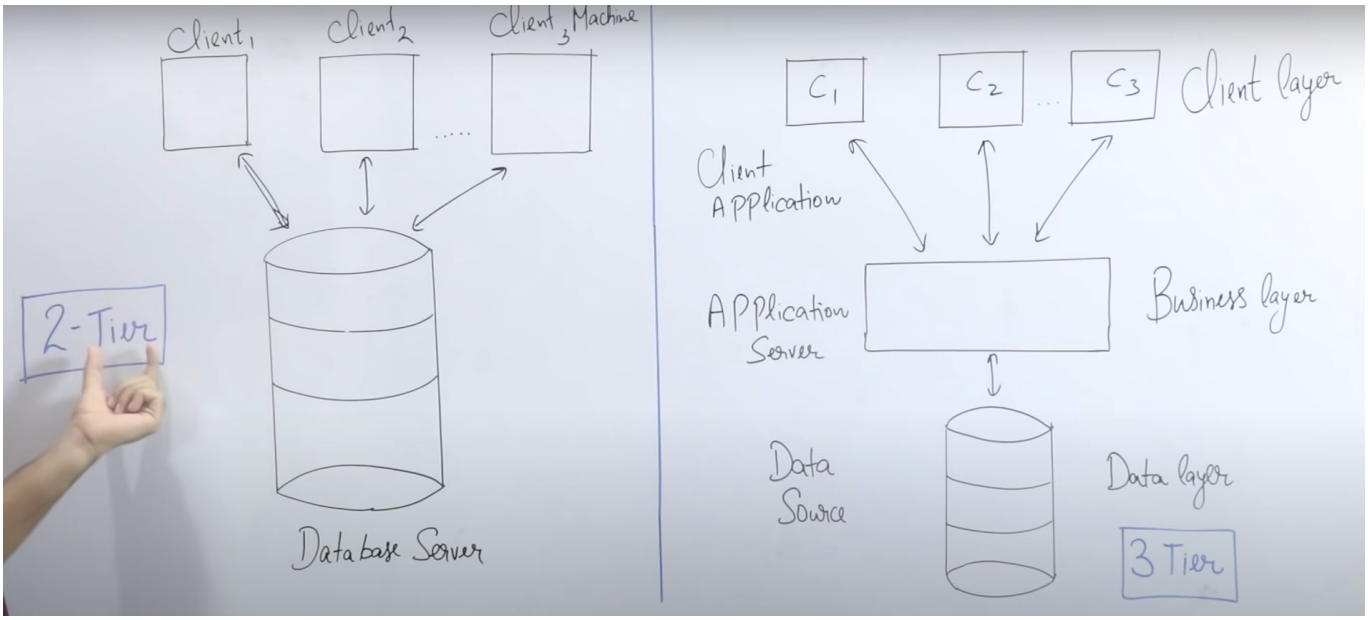


File System v/s DBMS

Why DBMS?

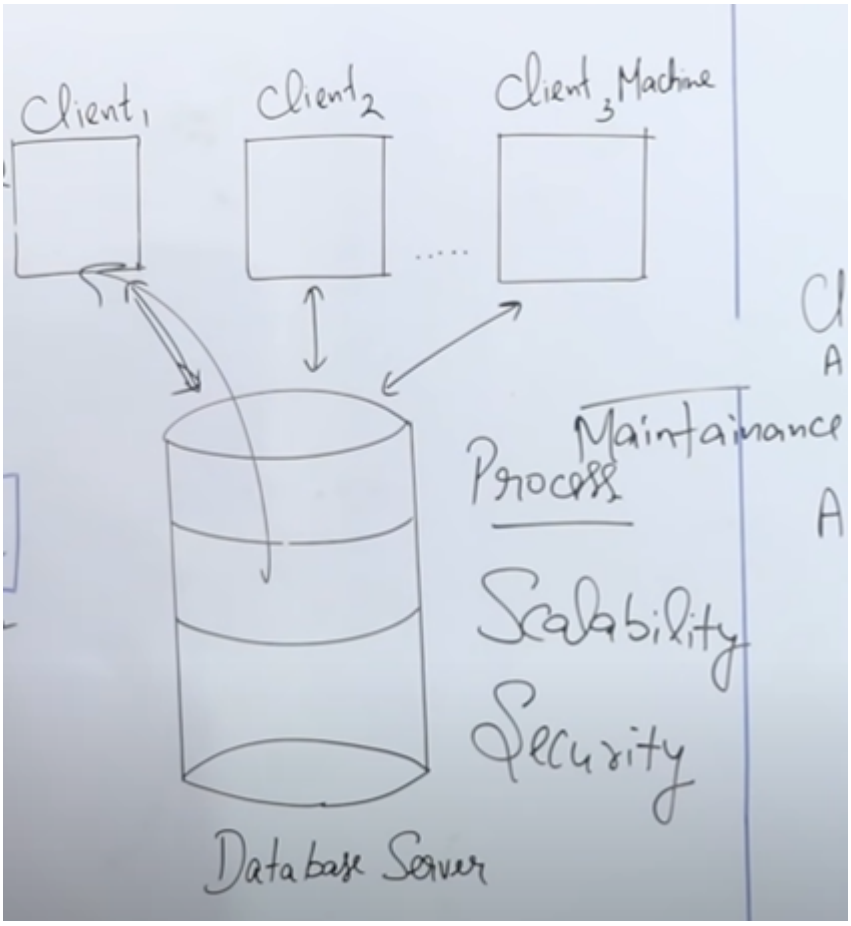


2-Tier & 3-Tier Architecture



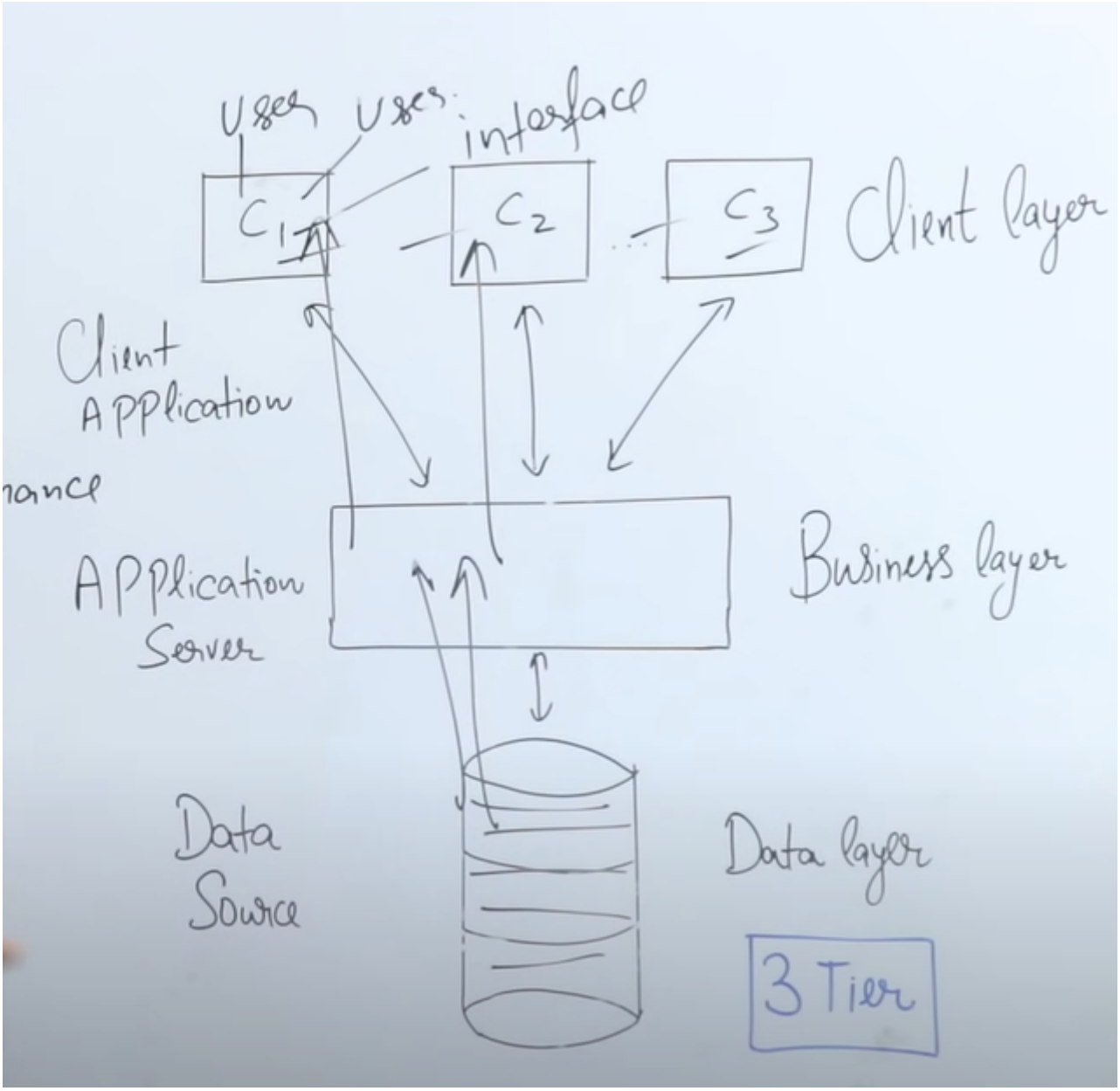
2 - Tier

Adv - Maintainence Disadv - Scalability, Security

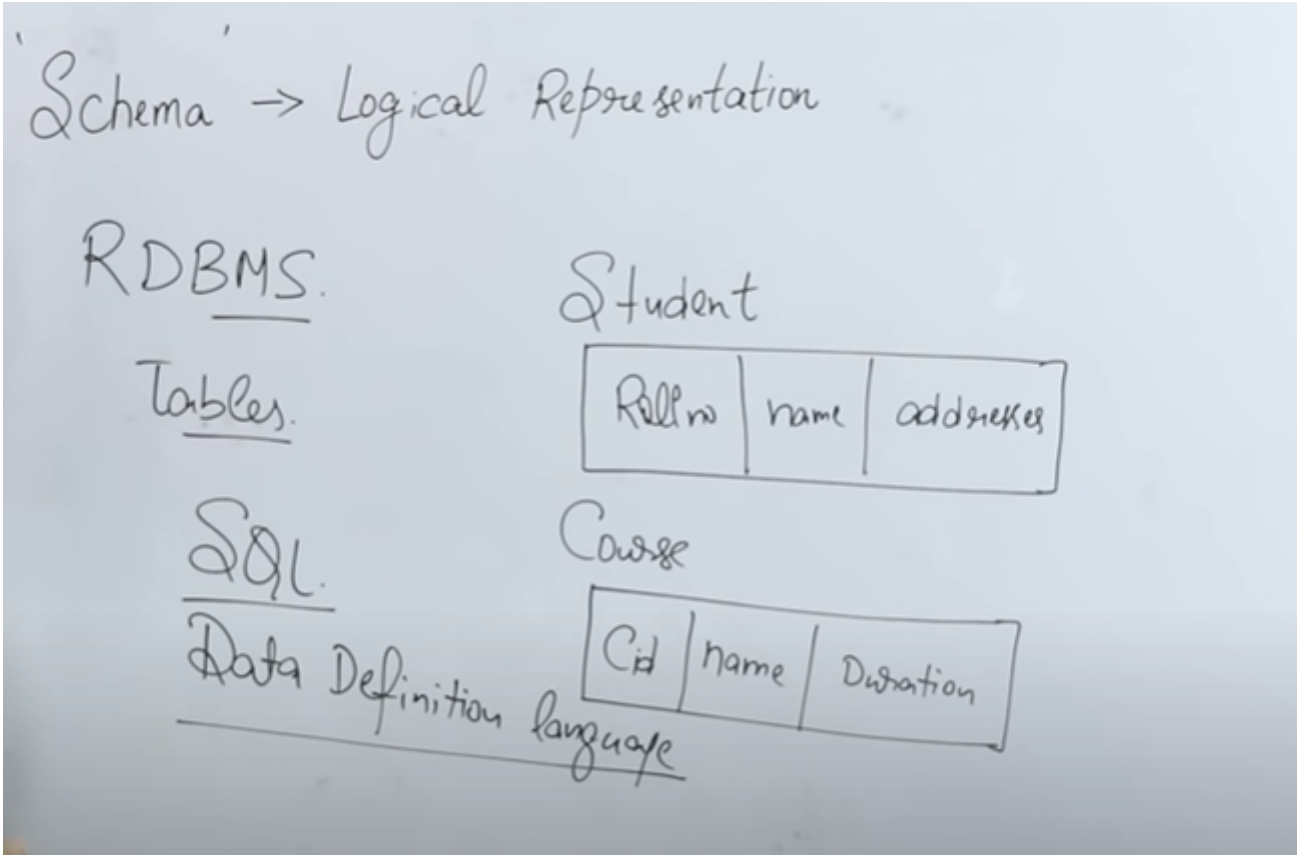


3 - Tier

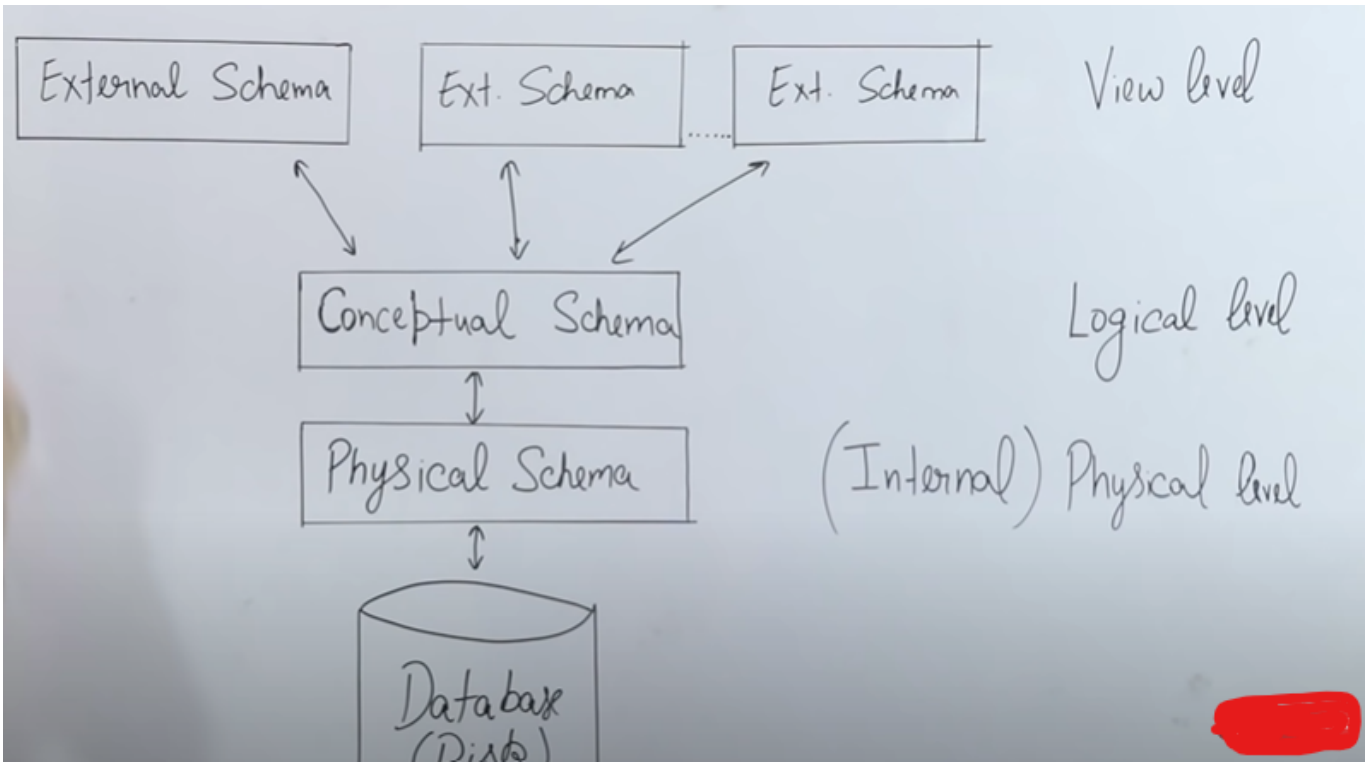
Adv - Scalability, Security Disadv - Maintainence

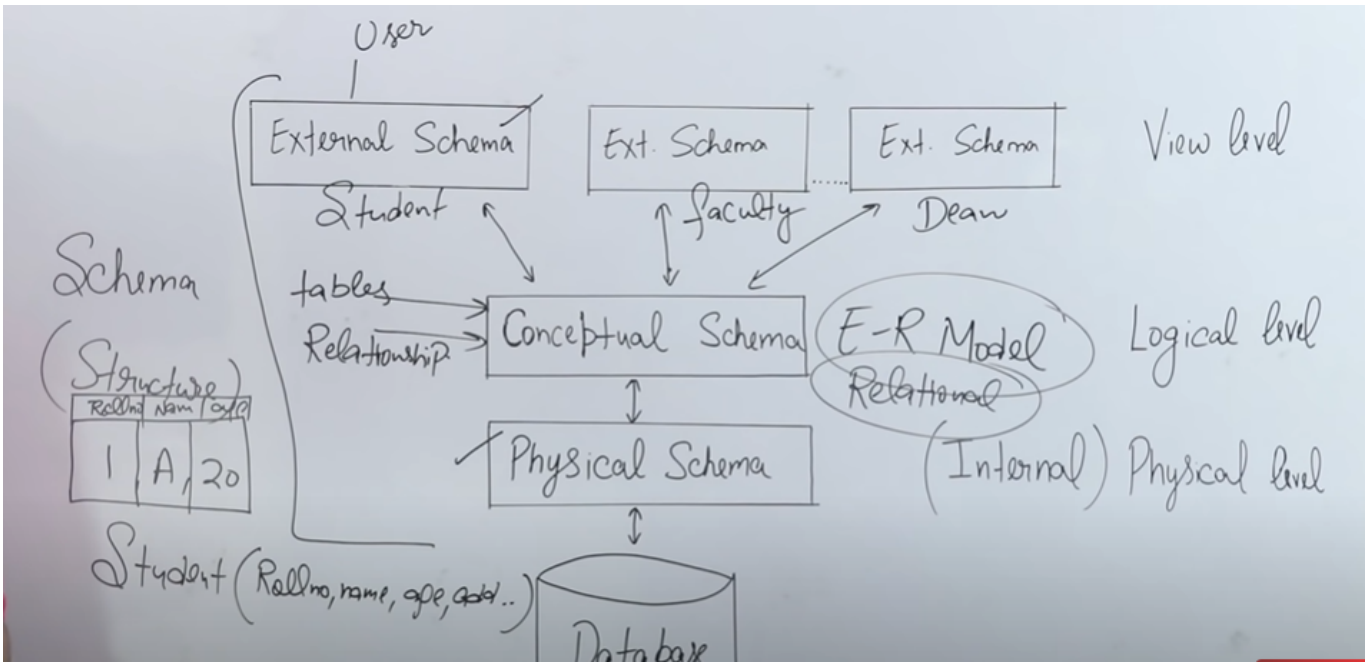


Schema

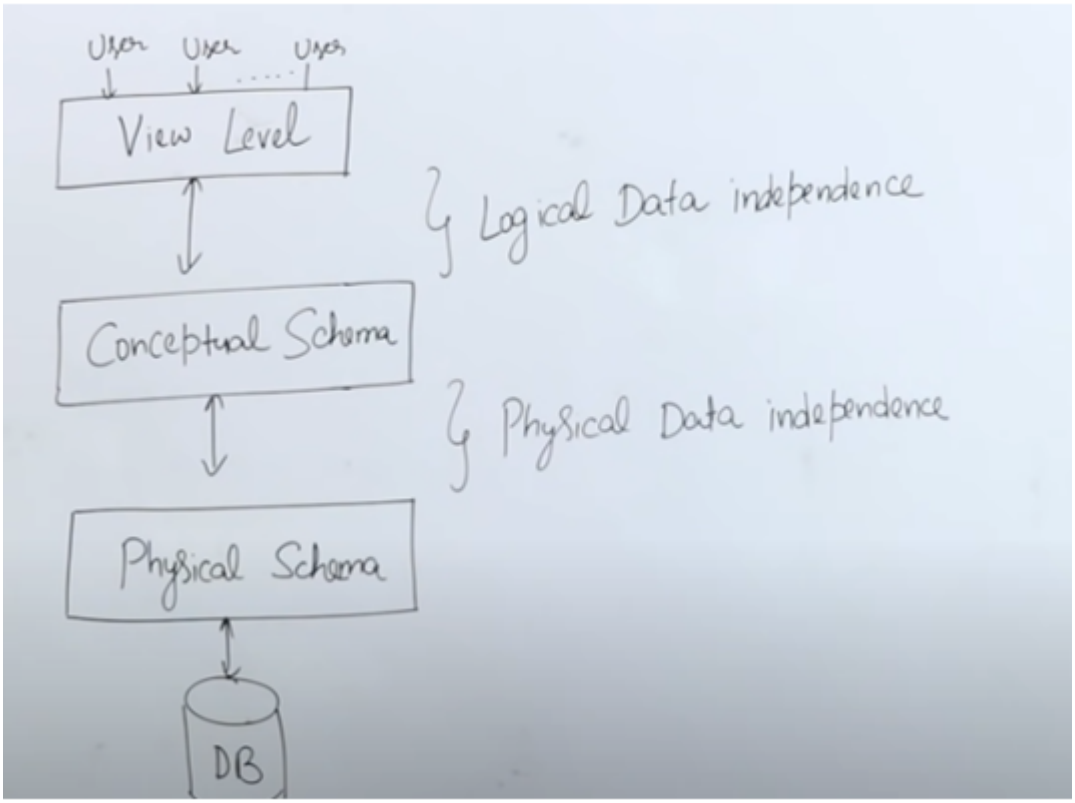


3 Schema Architecture

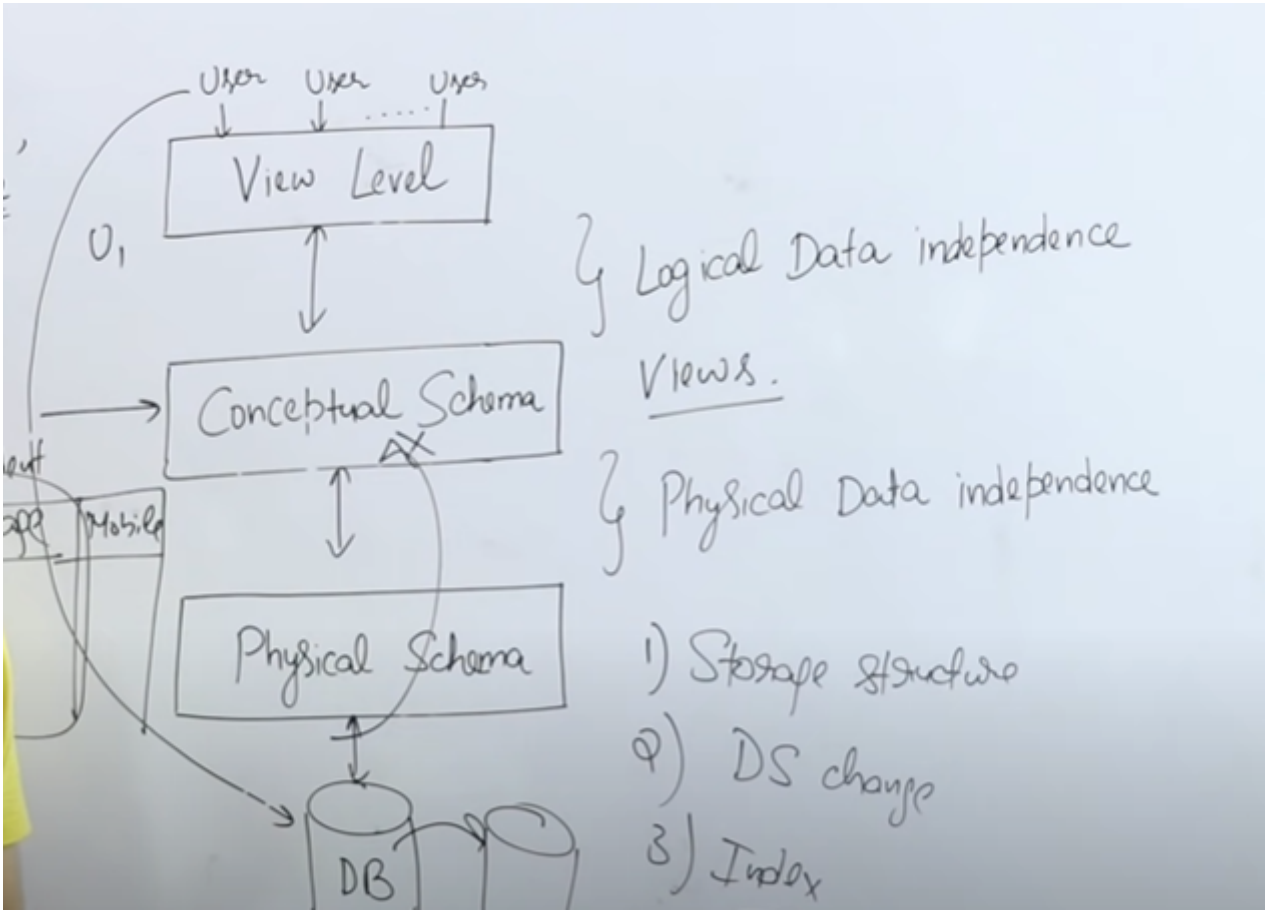




Data Independence



Logical and Physical Data Independence



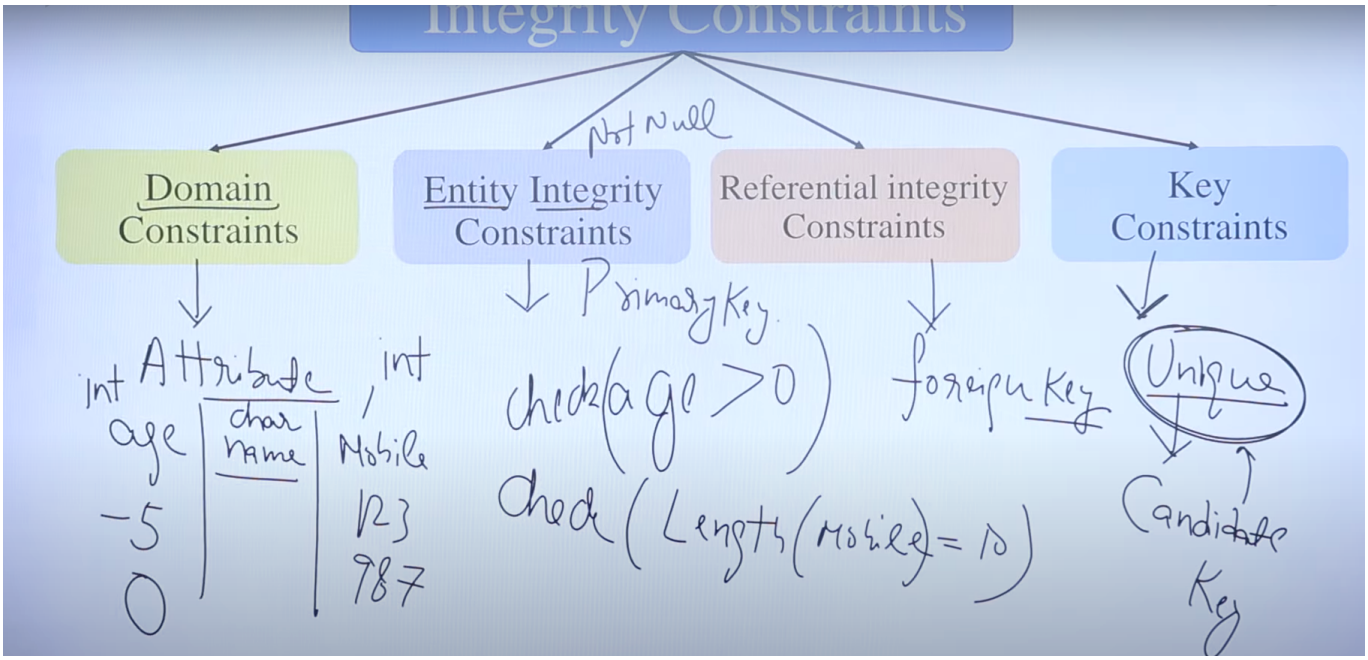
Integrity constraints

Integrity Constraints

Rules

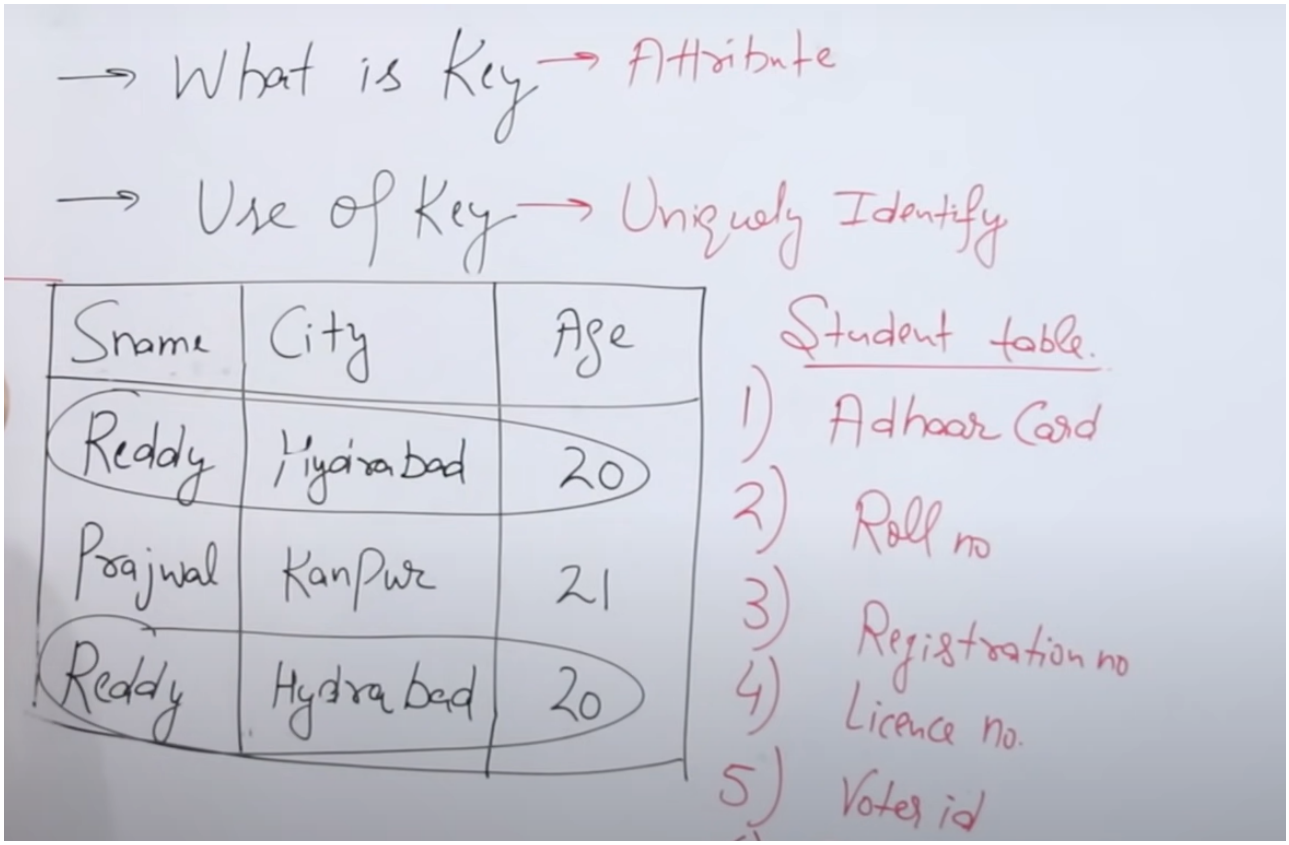
Integrity constraints are rules defined in a database to maintain data accuracy, consistency, and reliability.

They ensure that data stored in the database follows predefined rules or conditions.



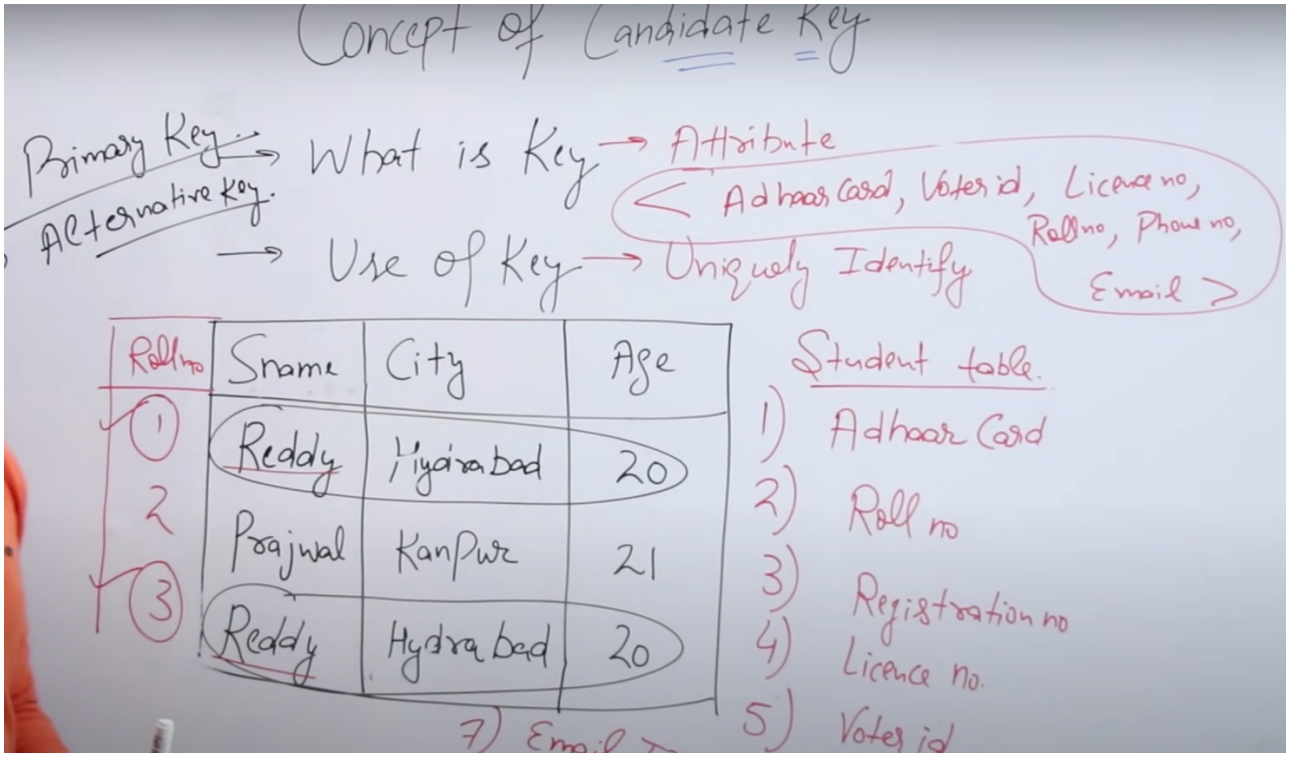
Key

- Any attribute



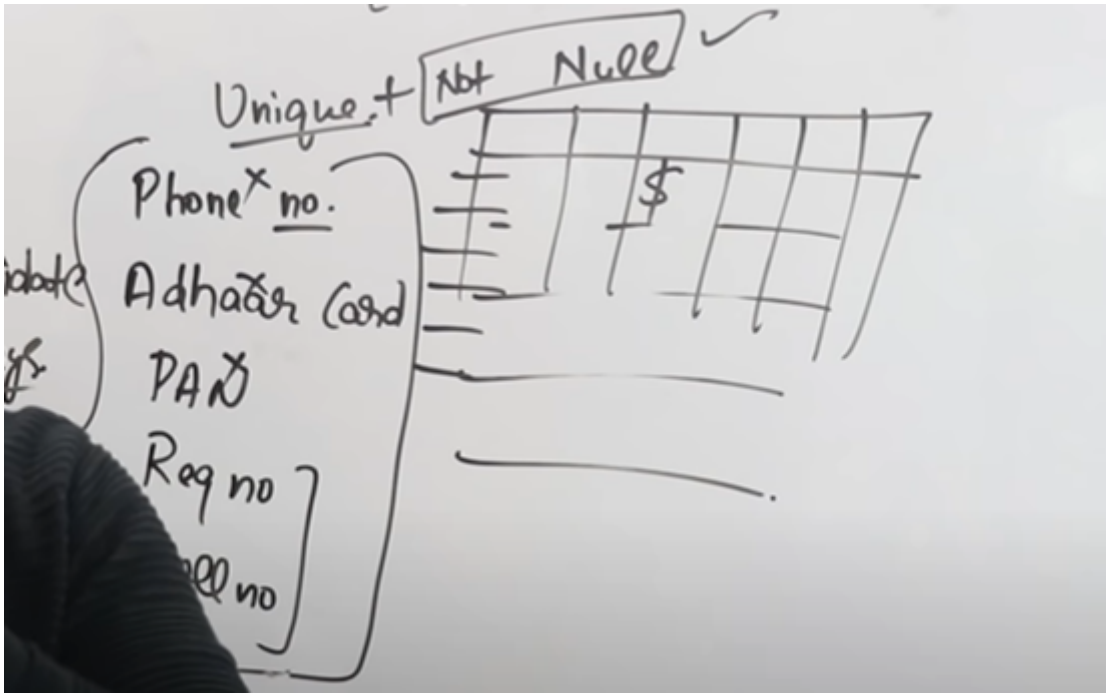
Candidate key

- Set of keys which uniquely identify
- From these choose a `primary key` and rest are `alternative key`



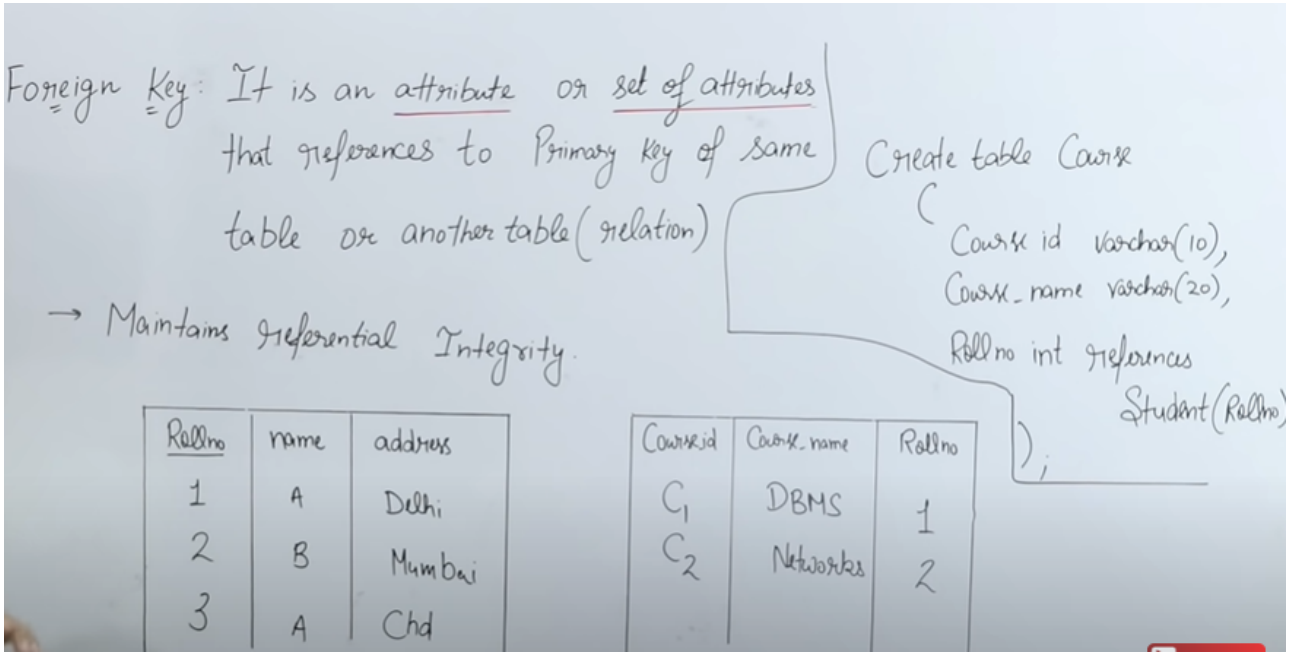
Primary key

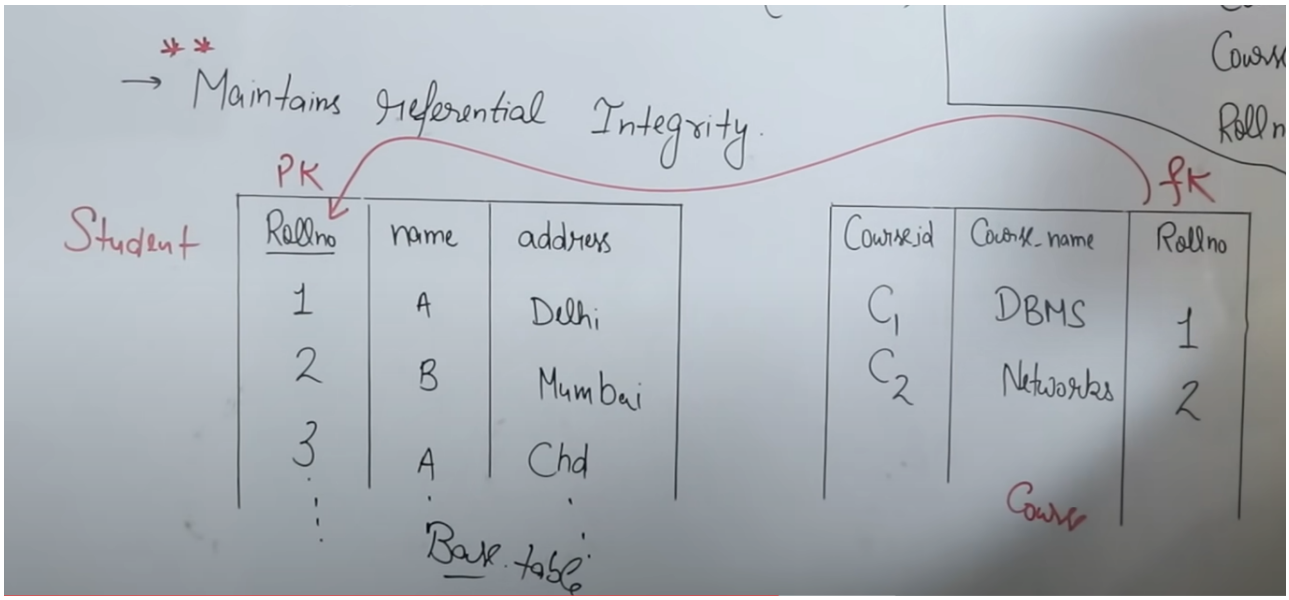
- `Unique + Not null`
- Only 1 Primary key in a db
- We give to user and not input



Foreign Key

- `Maintain Refrential Integrity`





- FK apart from PK not allowed

STUDENT

roll_no	name	age
1	abc	10
2	xyz	10
3	pqr	11

COURSE

course_id	name	roll_no
c1	dbms	1
c2	java	2
c3	os	4

- Multiple FK allowed
- FK and PK can have different name in referenced and referencing table

STUDENT

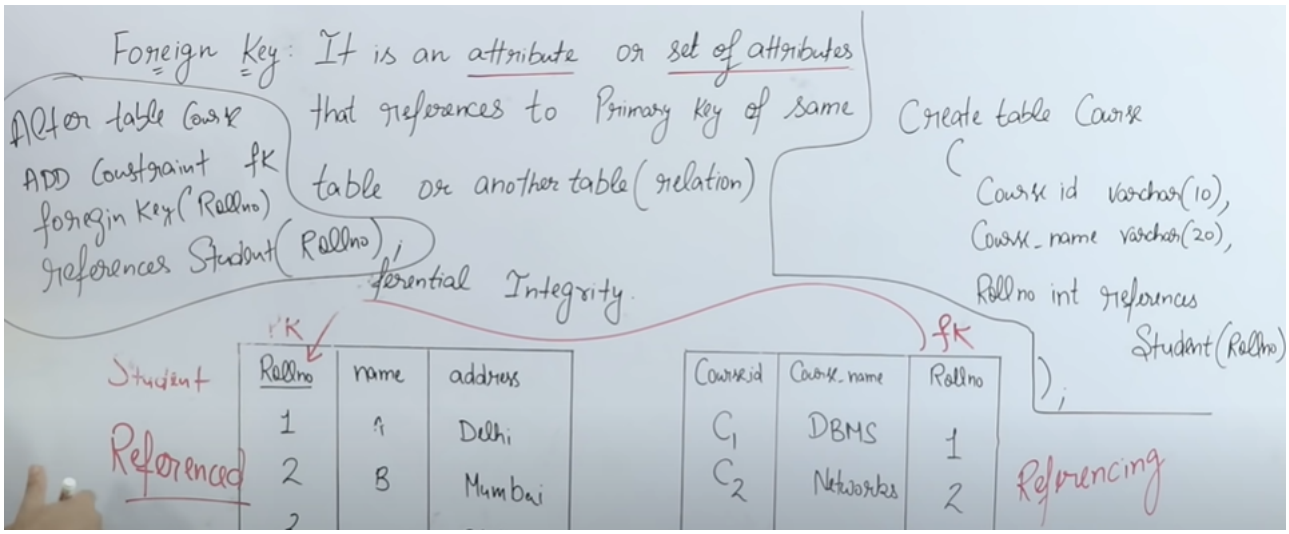
roll_no	name	age

COURSE

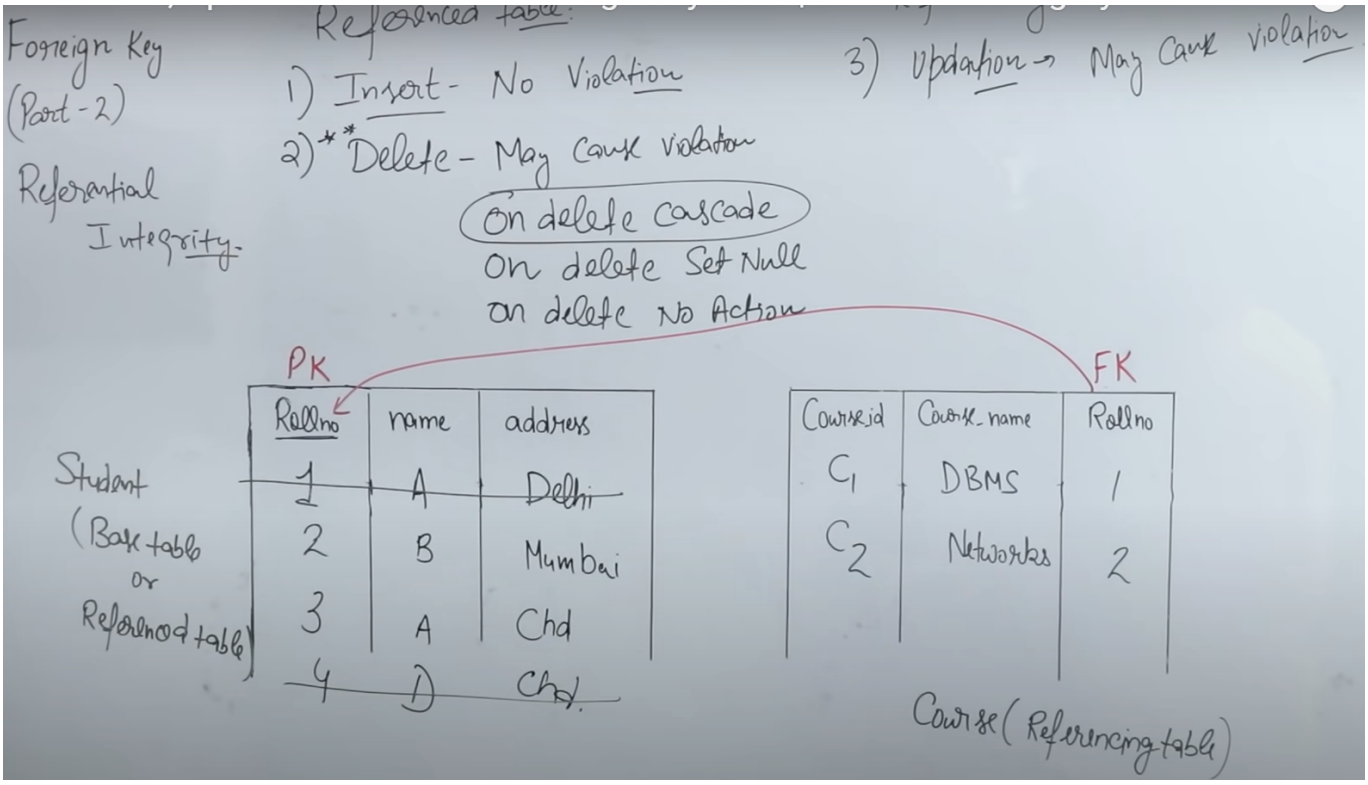
course_id	name	s_id	t_id

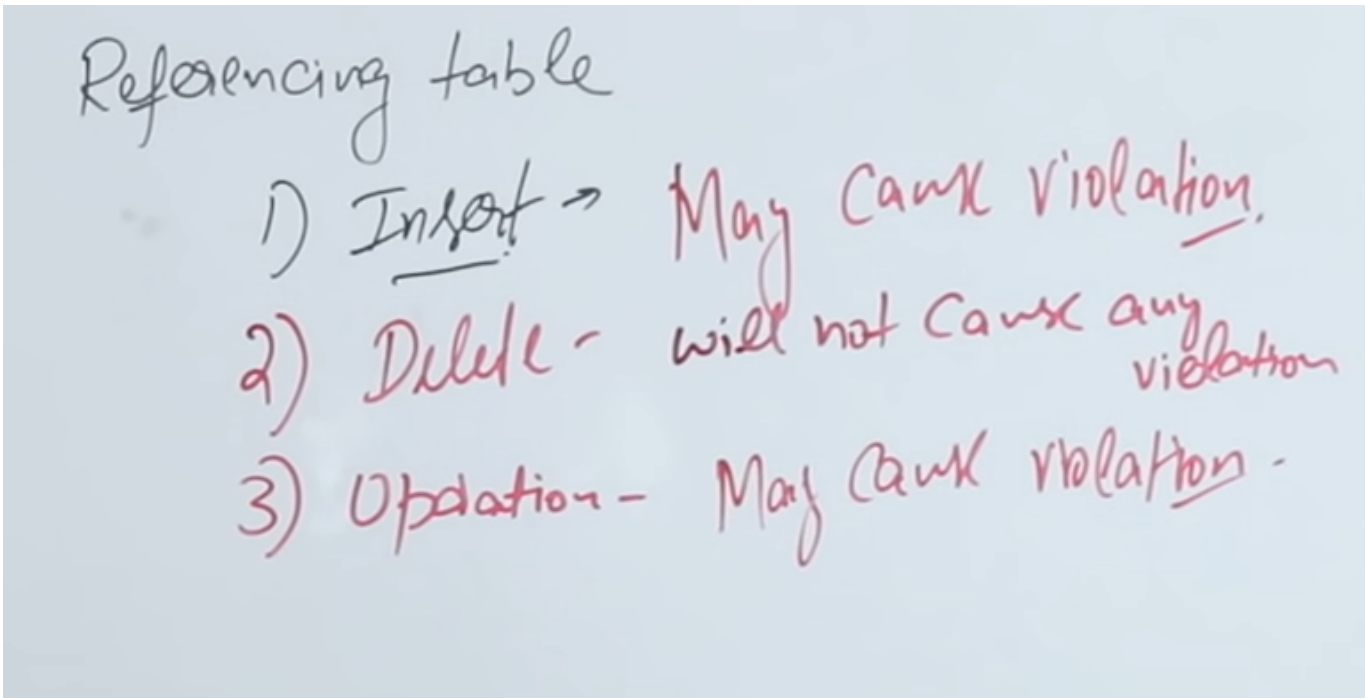
TEACHER

roll_no	name	age



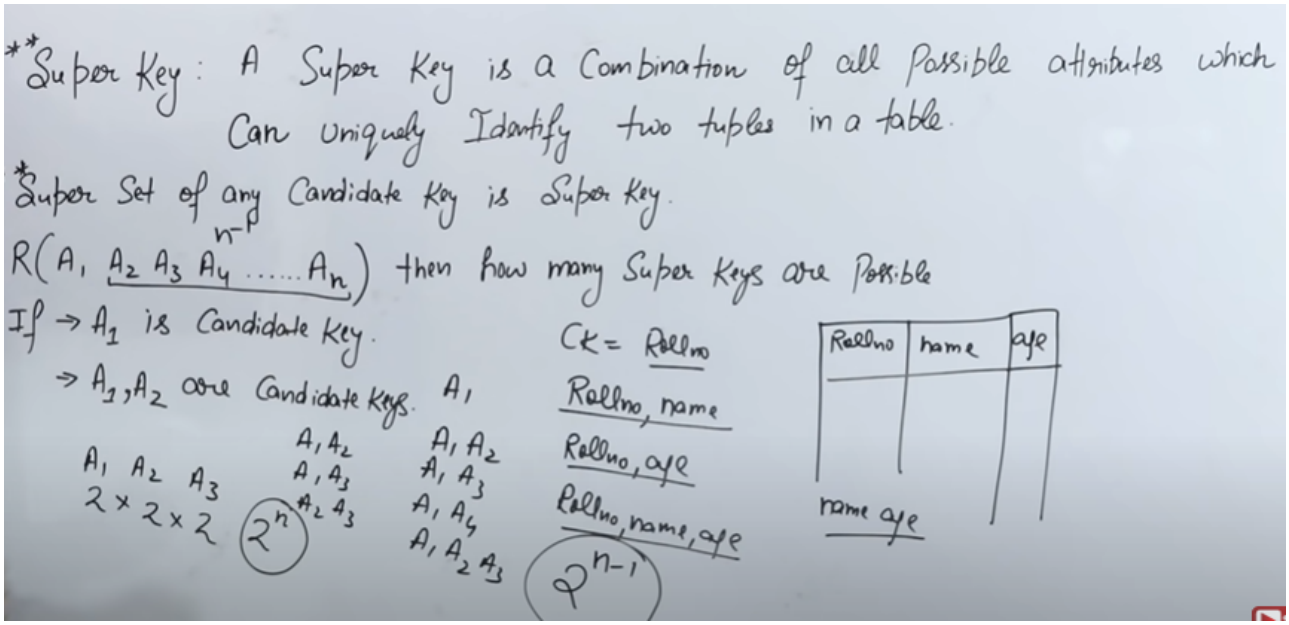
Insert, Update & Delete from Foreign Key table | Referential Integrity


Foreign Key
(Part-2)
Referential
Integrity
Referenced table
1) Insert - No Violation
2) ** Delete - May cause violation
On delete cascade
On delete Set Null
On delete No Action
3) Updation → May cause violation
PK
FK
Rollno
name
address
Student
(Back table
or
Referenced table)
1
A
Delhi
2
B
Mumbai
3
A
Chd
4
D
Chd
Course_id
Course_name
Rollno
C1
DBMS
1
C2
Networks
2
Course (Referencing table)

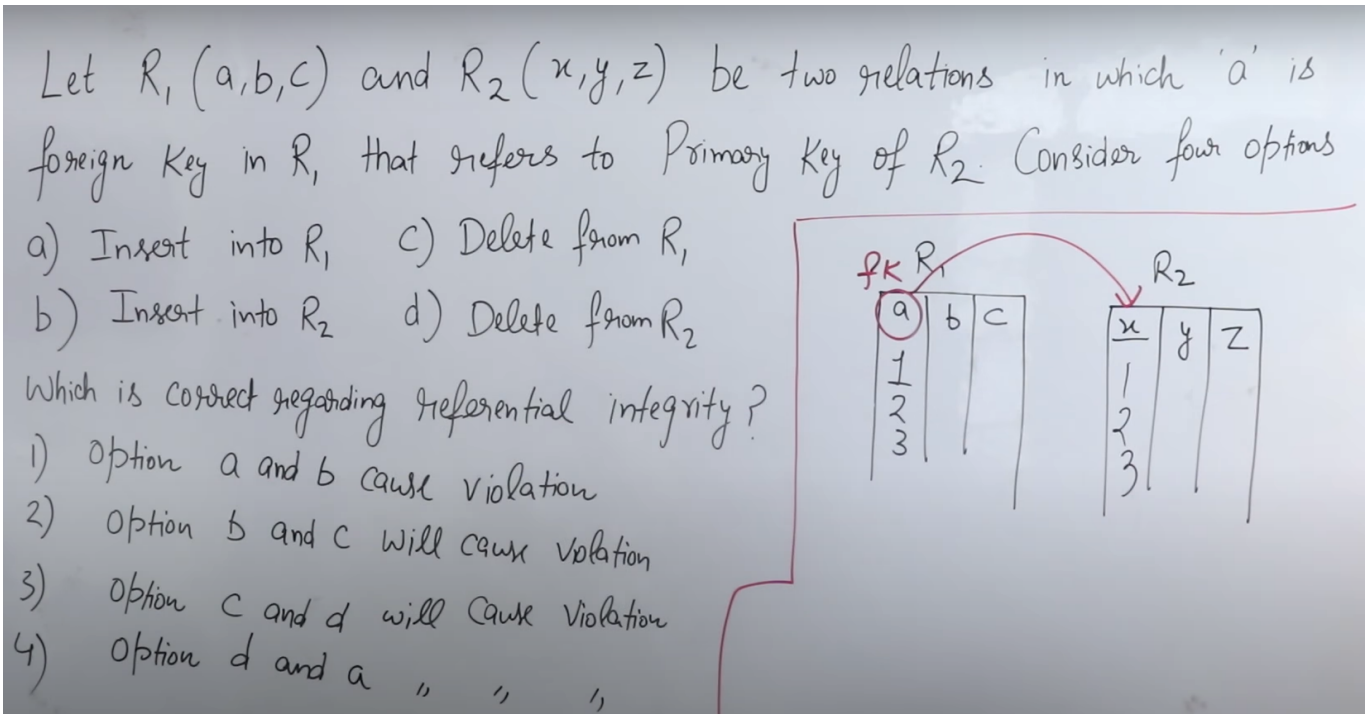

Referencing table
1) Insert → May cause violation.
2) Delete - will not cause any violation
3) Updation - May cause violation.

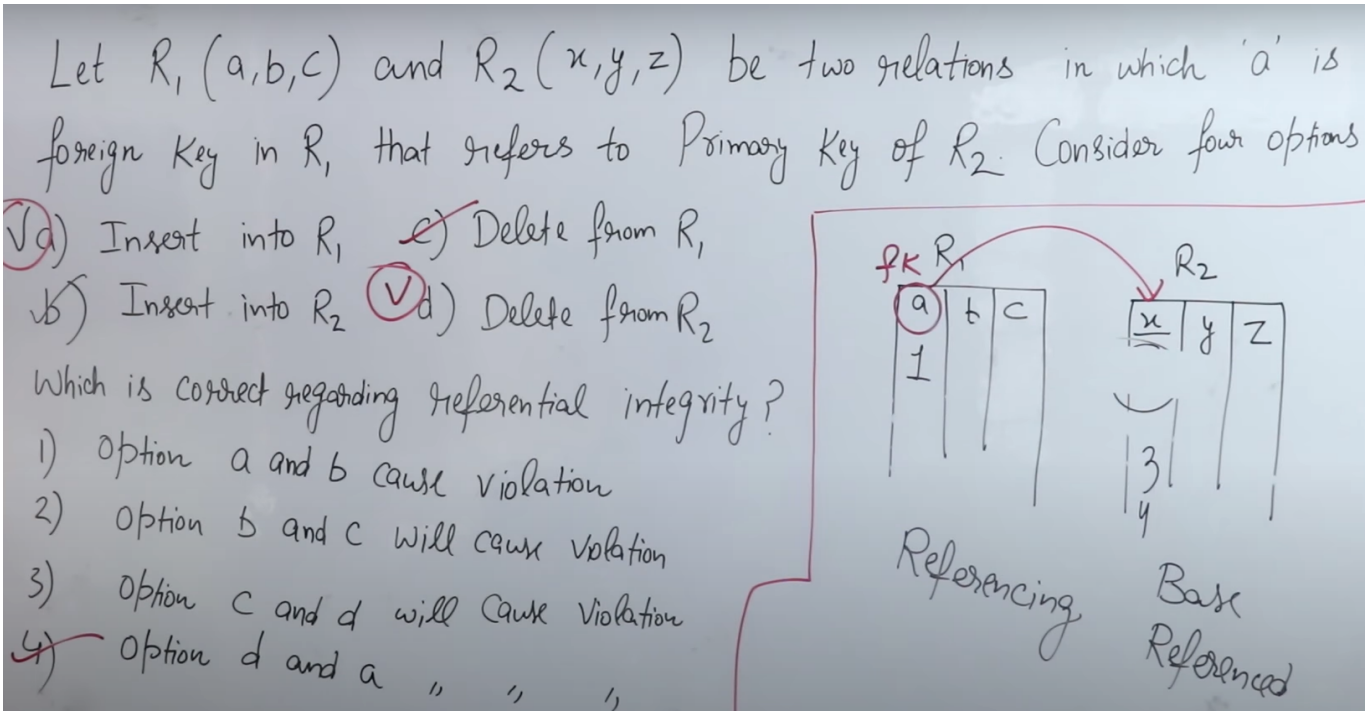
Super Key

- Candidate + some other key

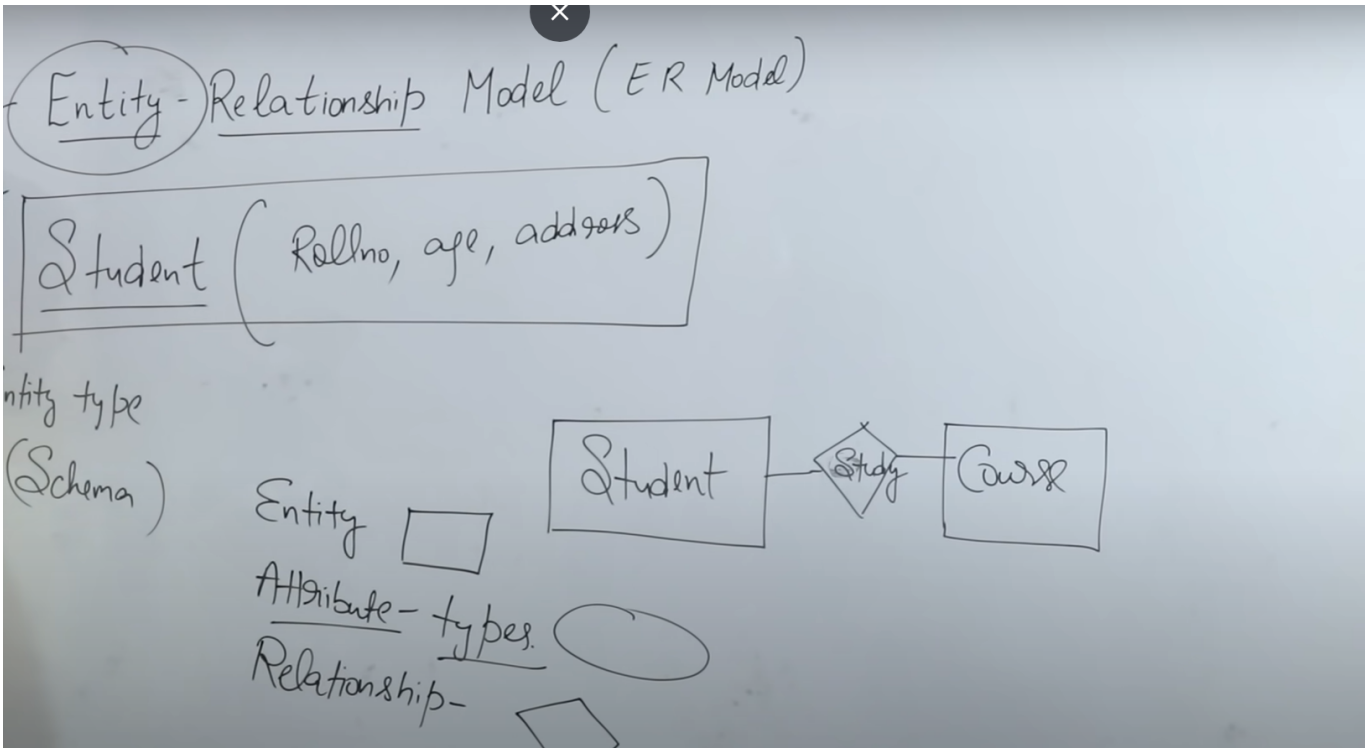


FK Question

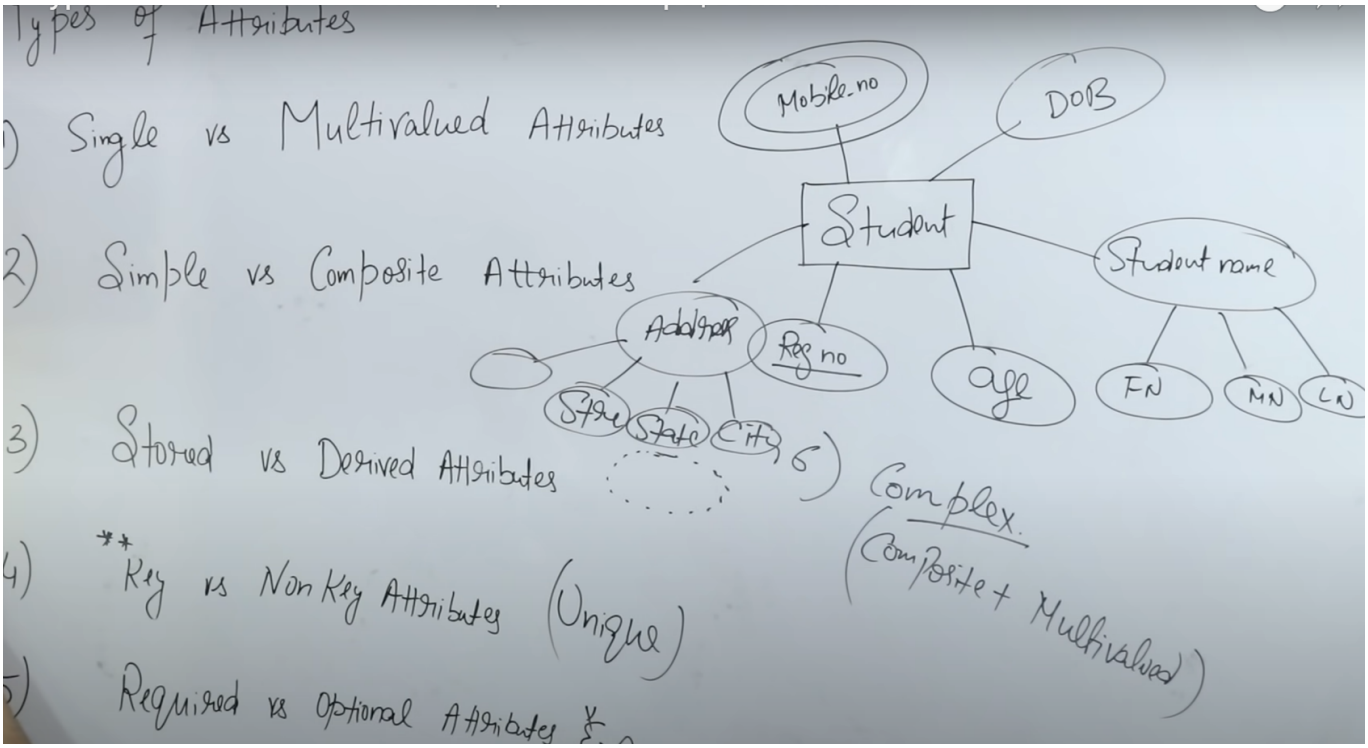

Let $R_1(a,b,c)$ and $R_2(x,y,z)$ be two relations in which 'a' is foreign key in R_1 that refers to Primary Key of R_2. Consider four options
a) Insert into R_1
b) Insert into R_2
c) Delete from R_1
d) Delete from R_2
Which is correct regarding referential integrity?
1) Option a and b cause violation
2) Option b and c will cause violation
3) Option c and d will cause violation
4) Option d and a " " "
fk R_1
a b c
1
2
3
R_2
x y z
1
2
3


Let $R_1(a,b,c)$ and $R_2(x,y,z)$ be two relations in which 'a' is foreign key in R_1 that refers to Primary Key of R_2. Consider four options
a) Insert into R_1
b) Insert into R_2
c) Delete from R_1
d) Delete from R_2
Which is correct regarding referential integrity?
1) Option a and b cause violation
2) Option b and c will cause violation
3) Option c and d will cause violation
4) Option d and a " " "
fk R_1
a b c
1
R_2
x y z
3
4
Referencing
Base Referenced

ER Model



Types of Attributes



- https://www.youtube.com/watch?v=Cn3RpA-QubY

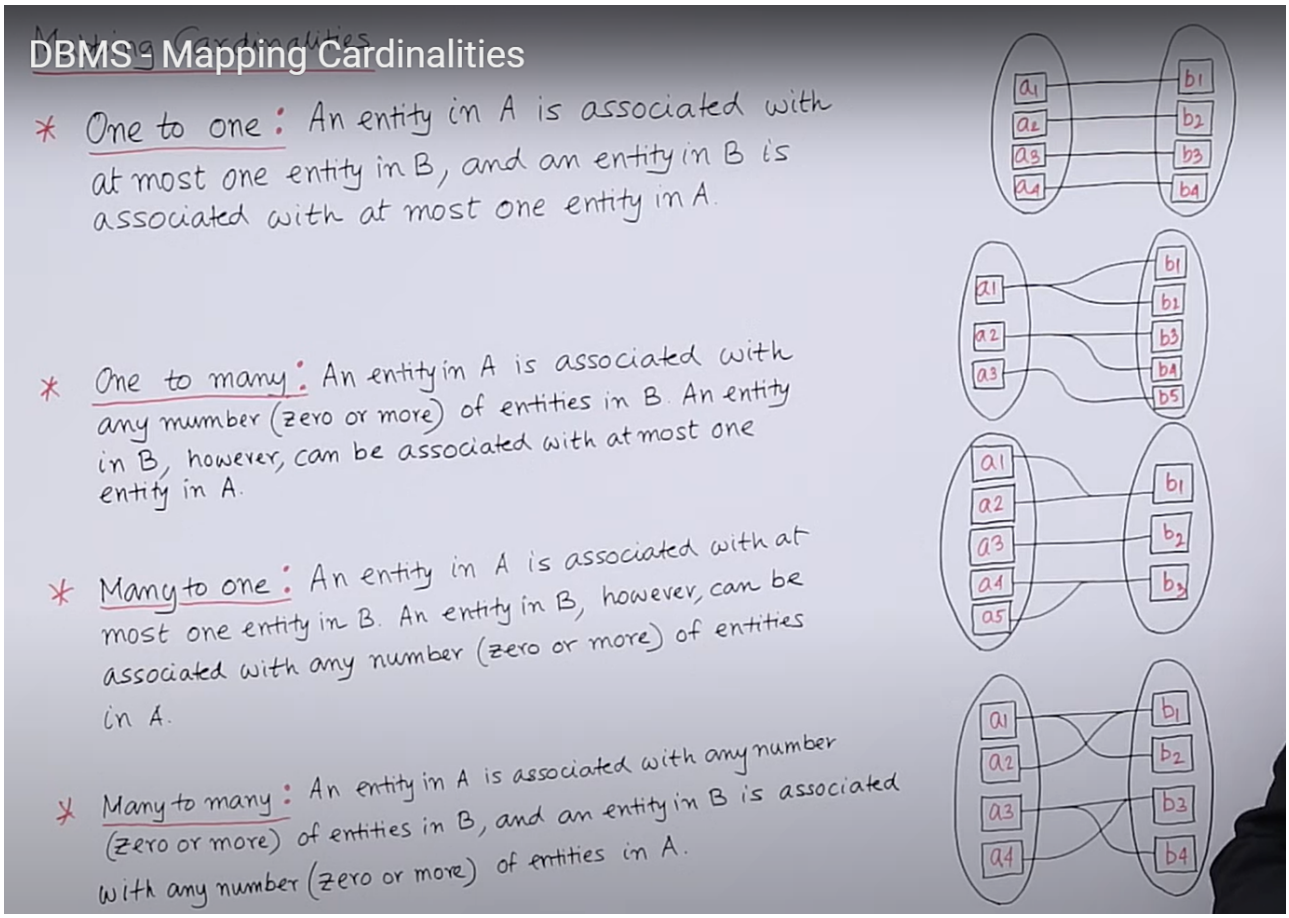
Steps to create ER Model

1. Idenity Entities
2. Determine their relationships
3. Identify attributes + Idenity PK

4. Cardinalities
5. Create with appropriate symbols

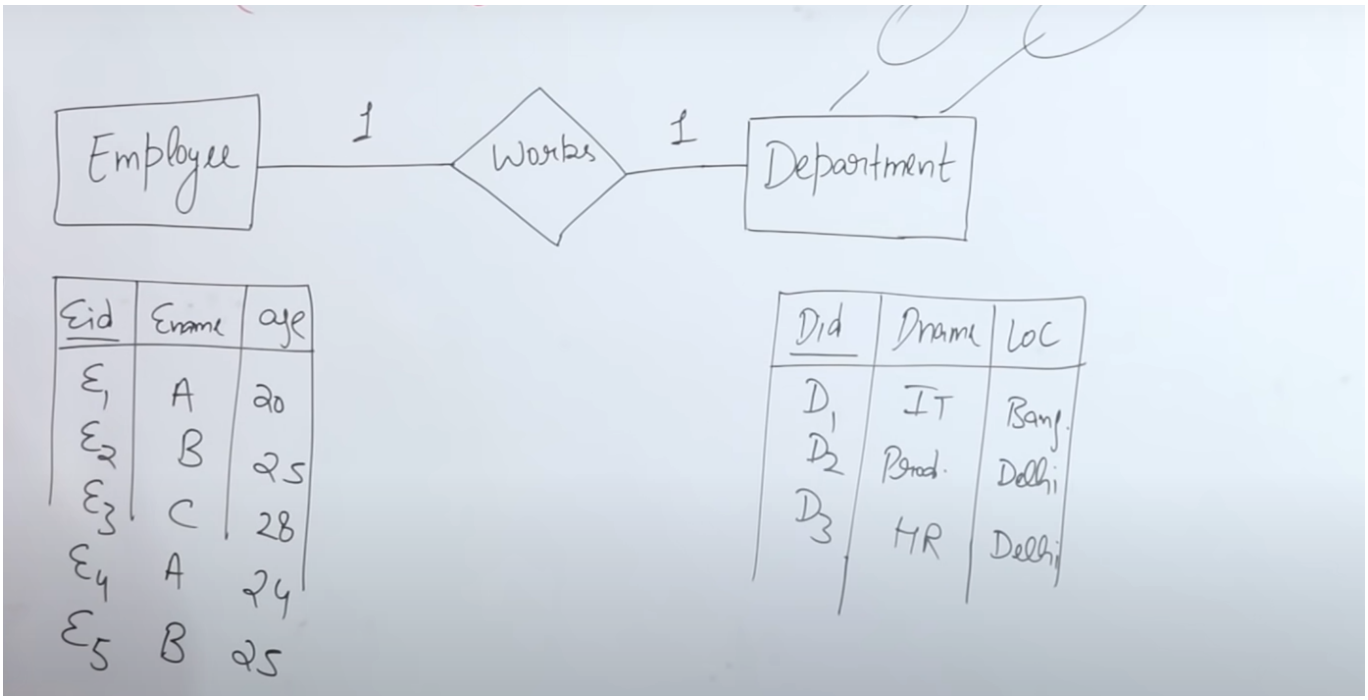
Degree of Relationship / Cardinality

1. 1:1
2. 1:M or M:1
3. M:N

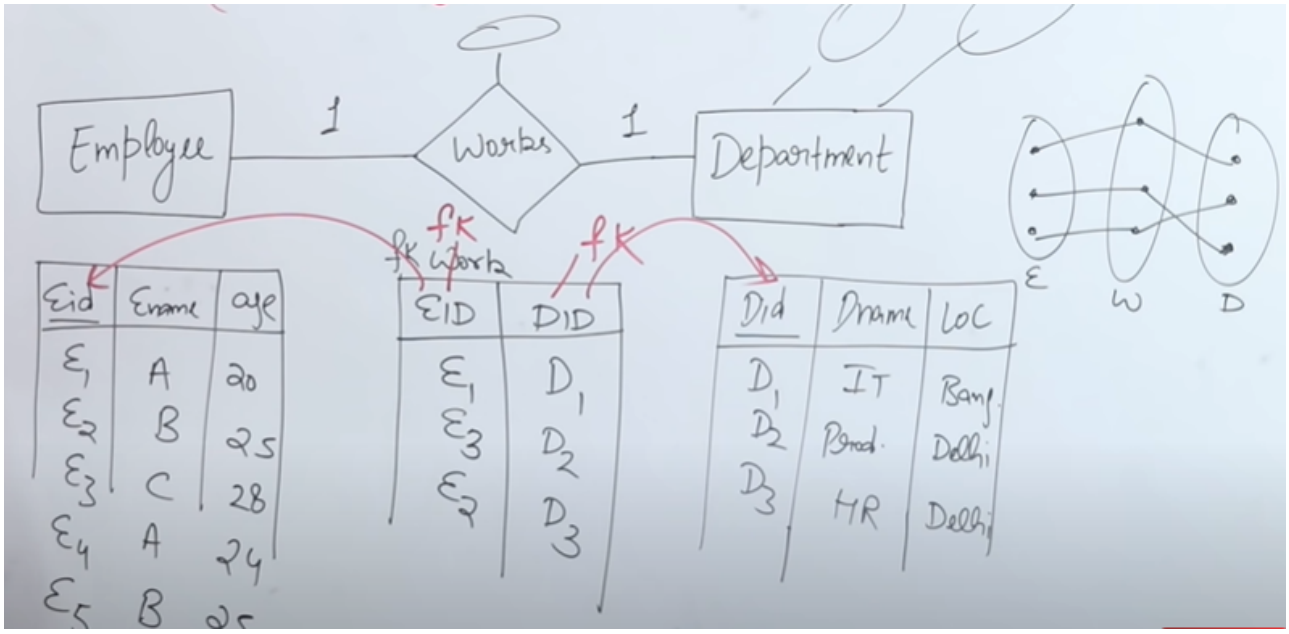


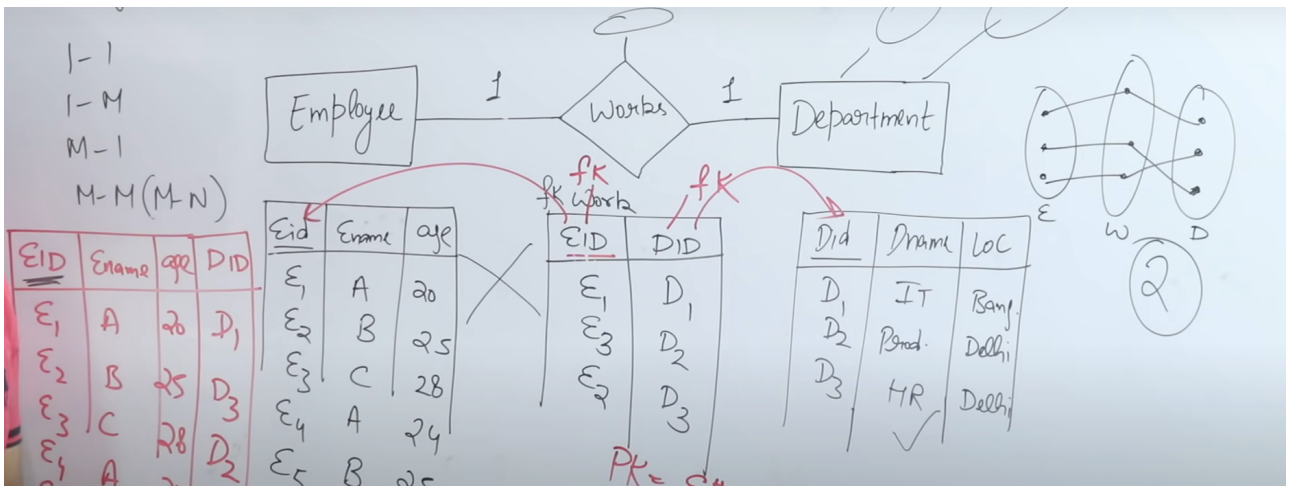
ER Model to Relational Model

1:1 Relationship

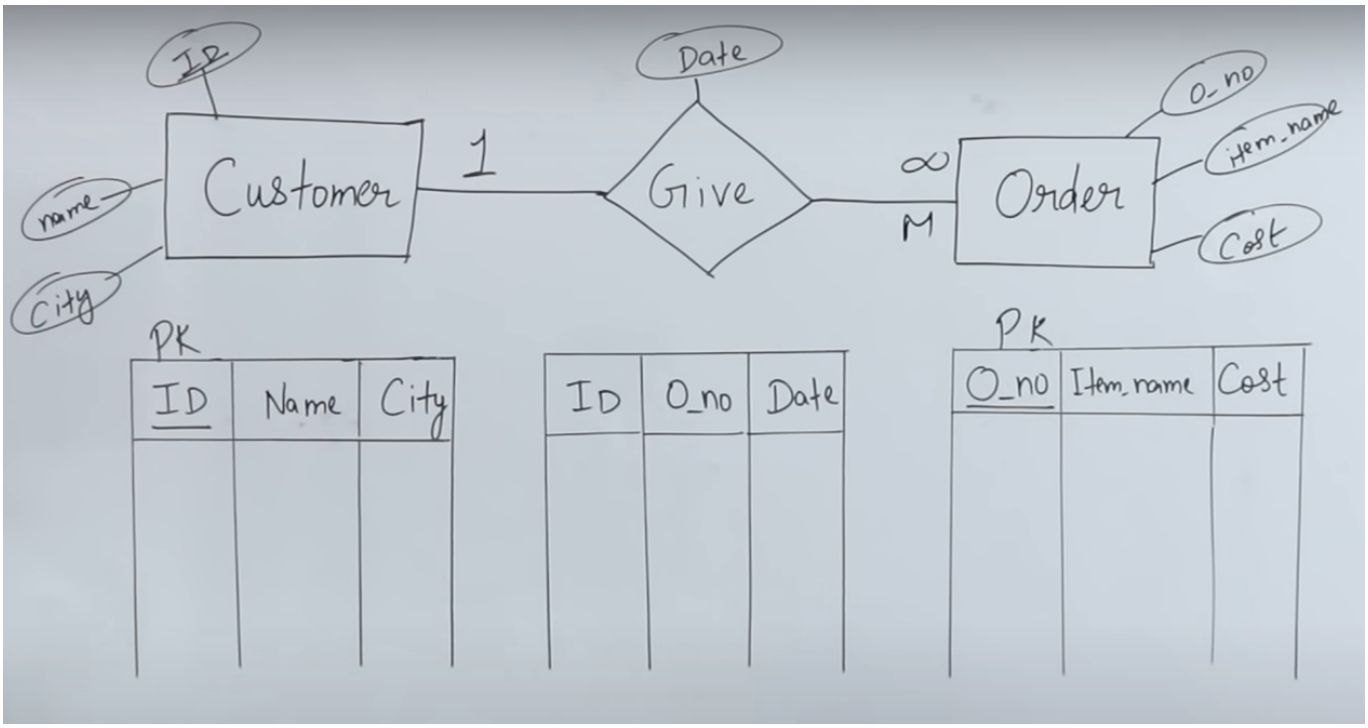


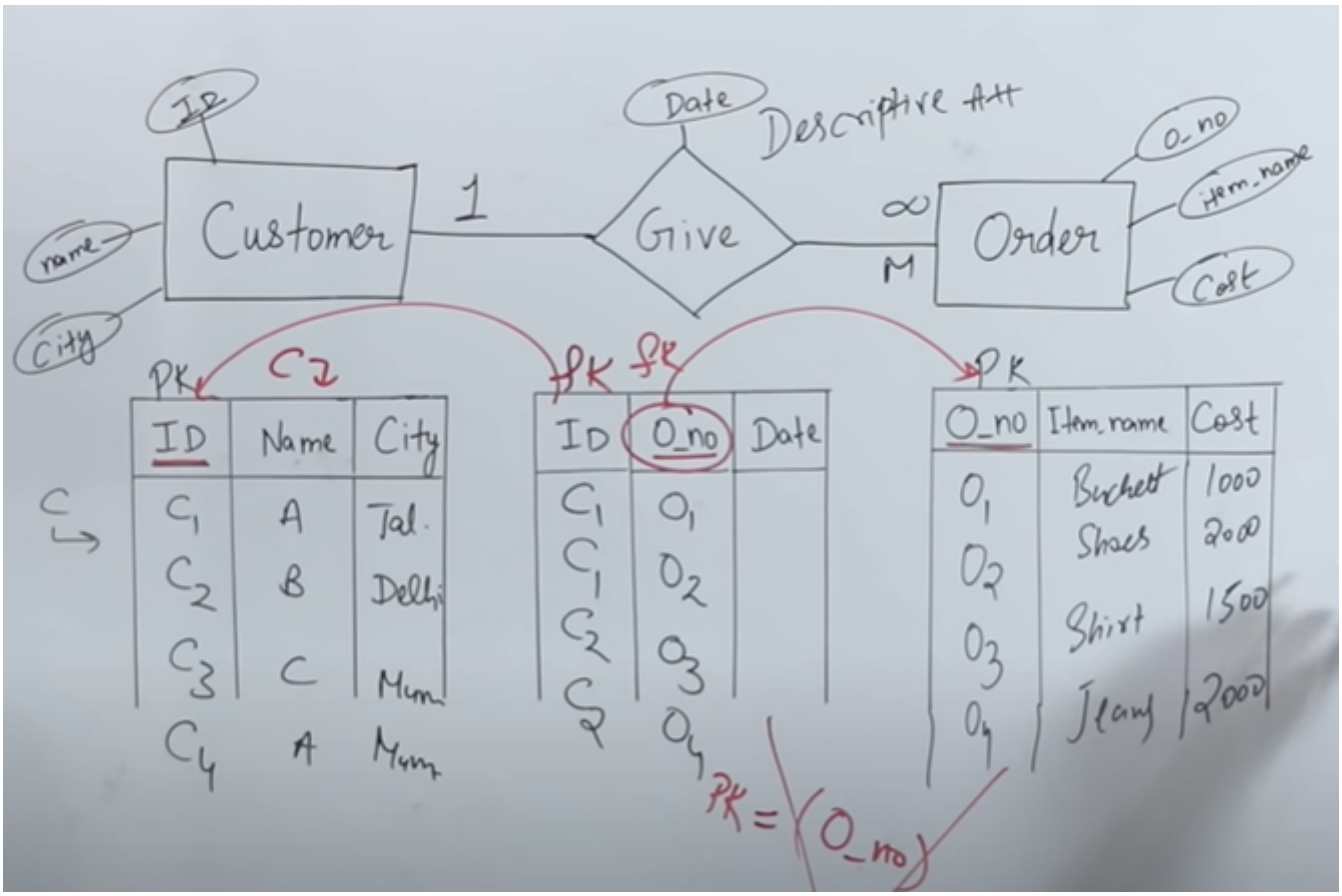
- PK - Either Eid or Did
- Yes can minimize
- Min 2 Tables





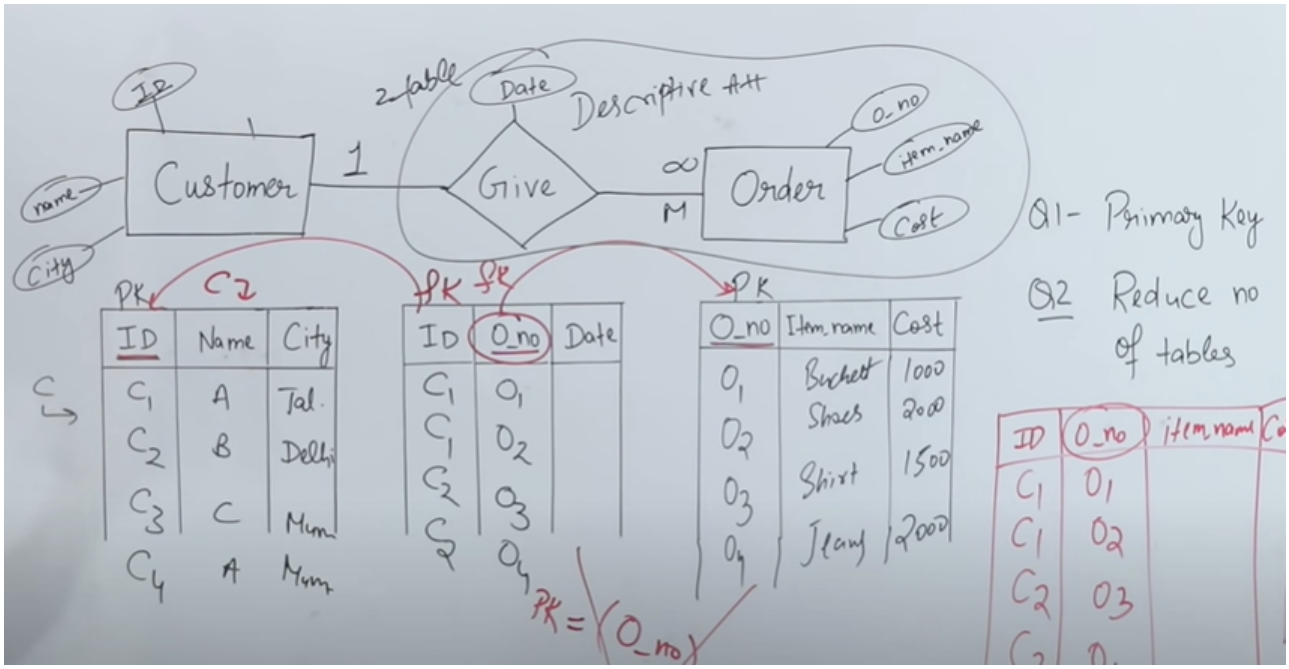
1:M



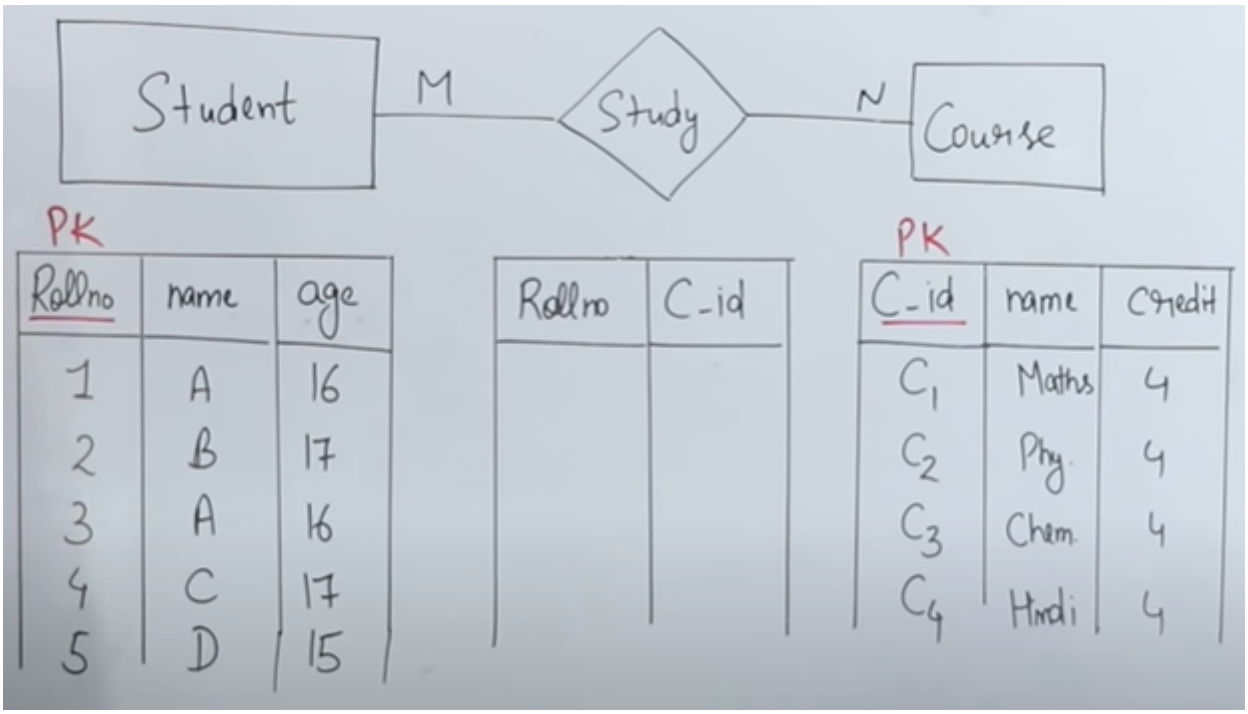


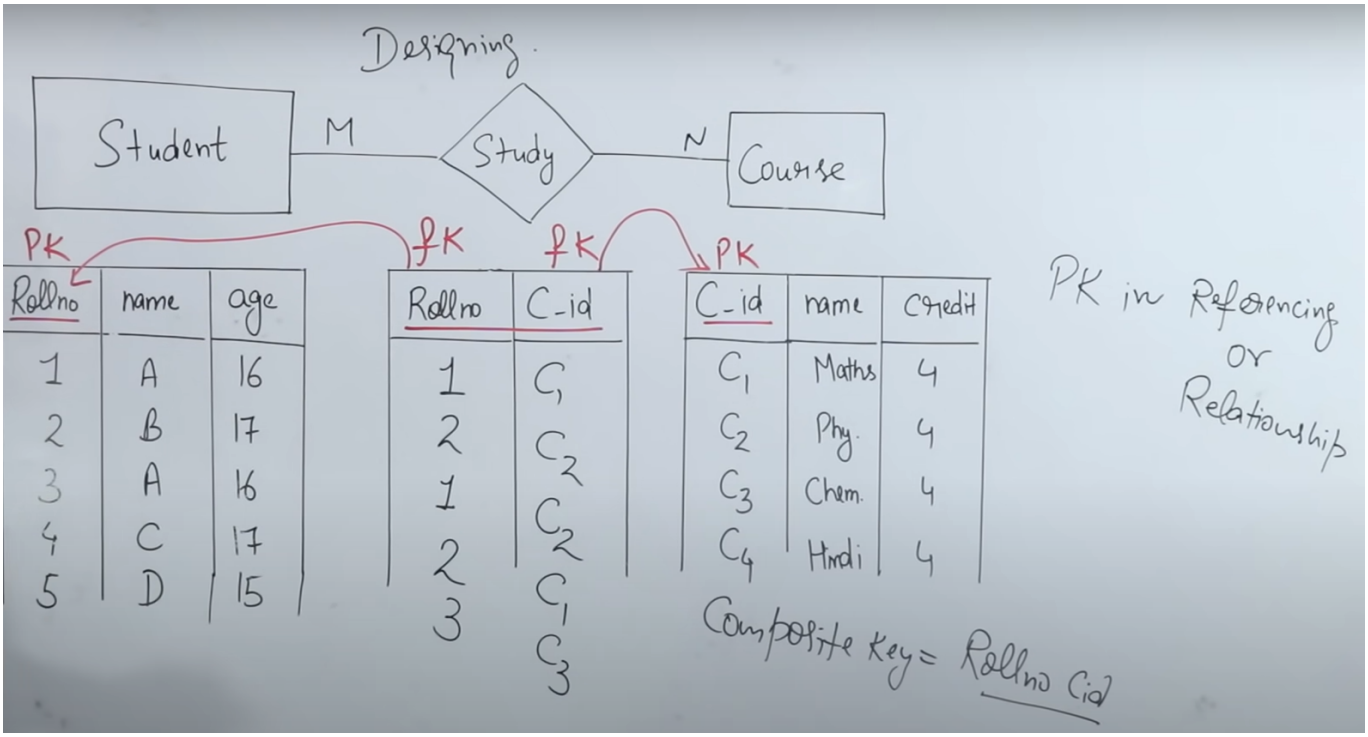
- PK - M side
- Yes can minimize (Relationship table with M side Table)

- Min 2 Tables



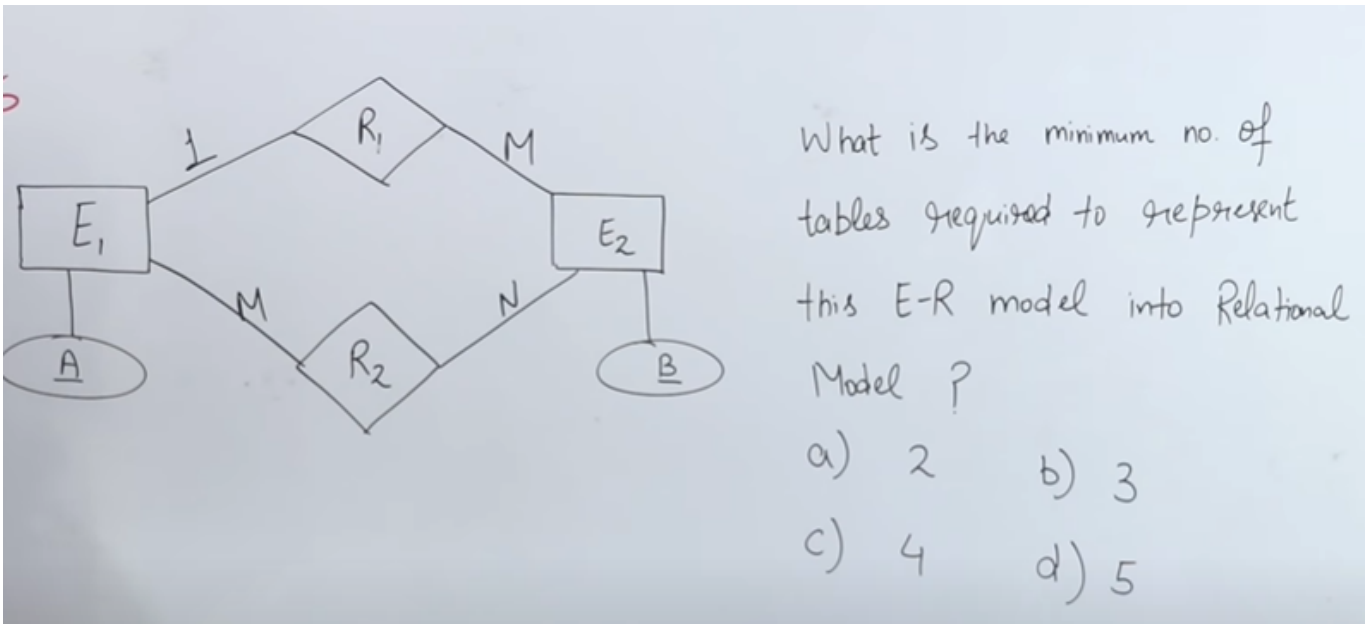
M:N

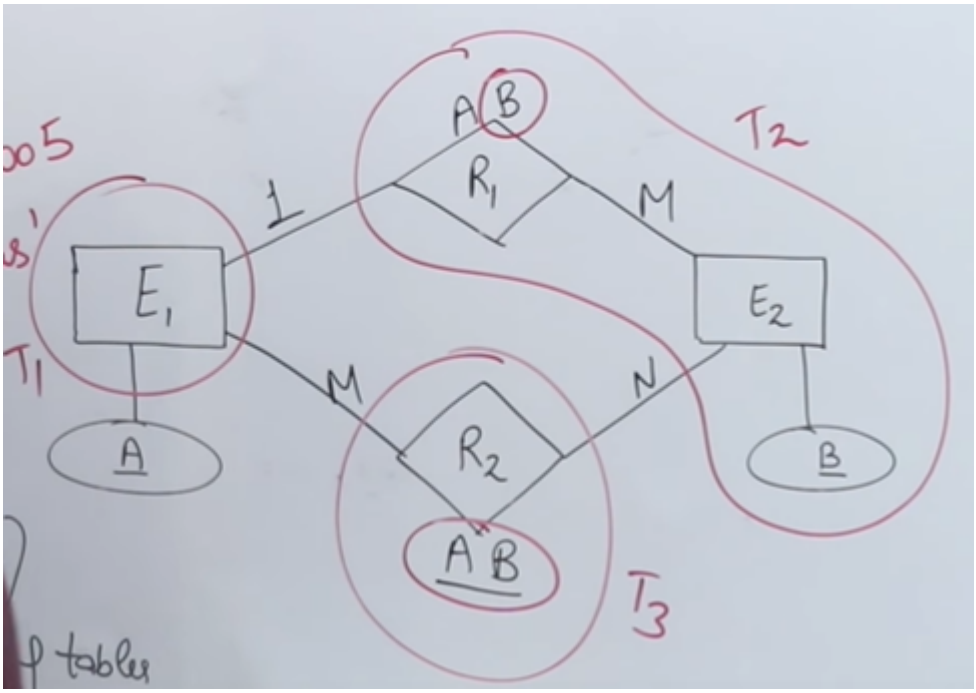




- PK - Composite : Both
- `No can't minimize`
- Min 3 Tables

Minimization Question



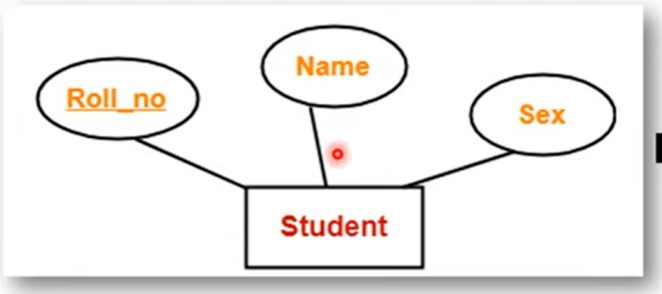


ER Model to Relational Model

Rules of Converting ER Diagram into Table

➢Rule 1: Strong Entity set with Simple attributes

- Attributes of the table will be the attributes of the entity set.
- The primary key of the table will be the key attribute of the entity set.



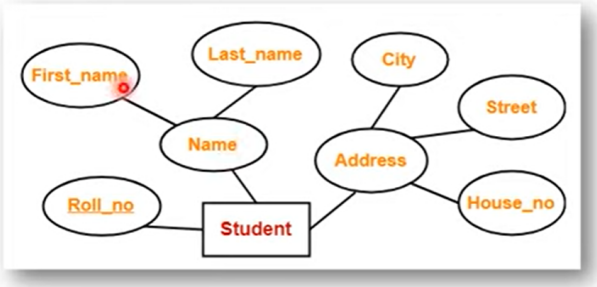
Roll_no	Name	Sex

Schema : Student (Roll_no , Name , Sex)

Rules of Converting ER Diagram into Table

➢Rule 2: For Strong Entity Set With Composite Attributes

- A strong entity set with any number of composite attributes will require only one table in relational model.
- While conversion, simple attributes of the composite attributes are taken into account and not the composite attribute itself.



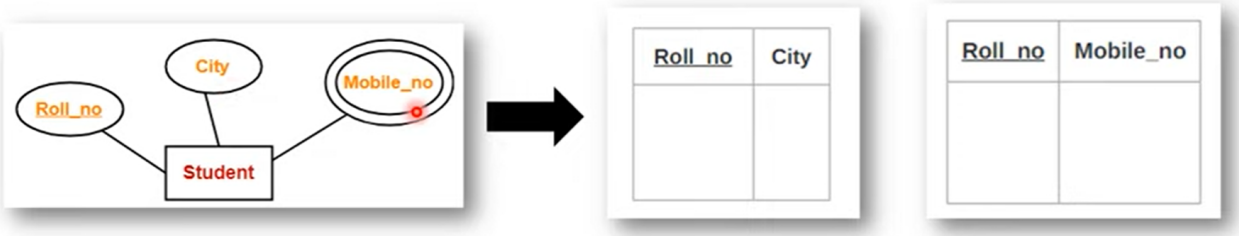
Roll_no	First_name	Last_name	House_no	Street	City

Schema : Student (Roll_no , First_name , Last_name , House_no , Street , City)

Rules of Converting ER Diagram into Table

➢Rule 3: For Strong Entity Set With Multi Valued Attributes

- A strong entity set with any number of multi valued attributes will require two tables in relational model.
- One table will contain all the simple attributes with the primary key.
- Other table will contain the primary key and all the multi valued attributes.

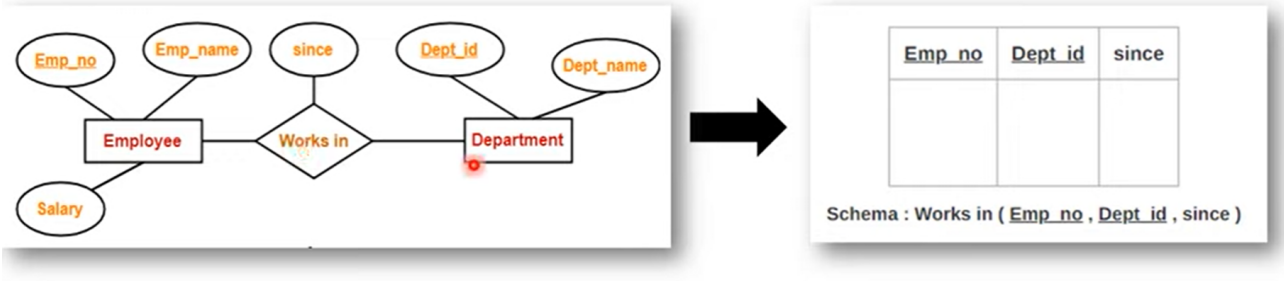


Rules of Converting ER Diagram into Table

➢Rule 4: Translating Relationship Set into a Table

Attributes of the table are-

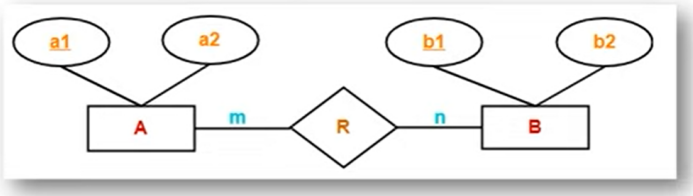
- Primary key attributes of the participating entity sets
- Its own descriptive attributes if any.
- Set of non-descriptive attributes will be the primary key.



Rules of Converting ER Diagram into Table

➢Rule 5: For Binary Relationships With Cardinality Ratios

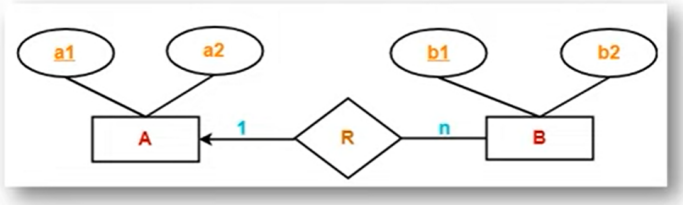
✓Case-1: For Binary Relationship With Cardinality Ratio m:n



Here, three tables will be required-

1. A (a1 , a2)
2. R (a1 , b1)
3. B (b1 , b2)

✓Case-2: For Binary Relationship With Cardinality Ratio 1:n



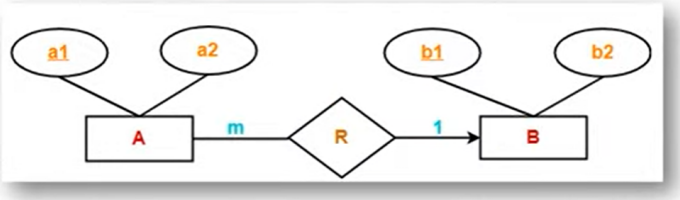
Here, two tables will be required-

1. A (a1 , a2)
2. BR (a1 , b1 , b2)

Rules of Converting ER Diagram into Table

➢Rule 5: For Binary Relationships With Cardinality Ratios

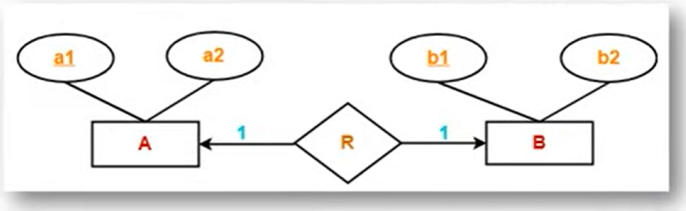
✓Case-3: For Binary Relationship With Cardinality Ratio m:1



Here, two tables will be required-

1. AR (a1 , a2 , b1)
2. B (b1 , b2)

✓Case-4: For Binary Relationship With Cardinality Ratio 1:1



Way-01:

1. AR (a1 , a2 , b1)
2. B (b1 , b2)

Way-02:

1. A (a1 , a2)
2. BR (a1 , b1 , b2)

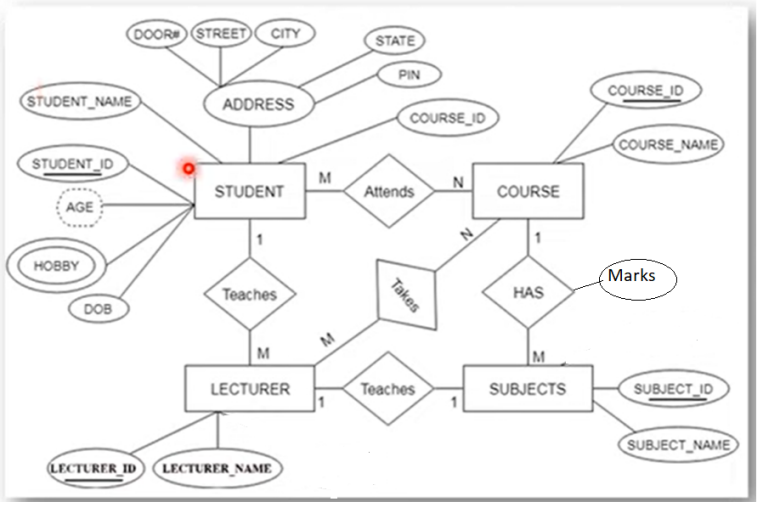
Attributes

1. Single - Oval - Directly add as column
2. Multivalued - double oval - Make seperate table alongwith PK
3. Composite - many ovals from a oval - Add individual attributes as col
4. Derived - dashed oval - Do not add in table
5. Complex - composite + multivalued - Seperate table + individual attributes + PK

Steps :

1. Tables for Individual Entities Weak Entity(Double Rectangle) : Add PK of strong entity

2. Table for multivalued(if any)
3. Relationships
    1. M:M (Seperate table)
    2. 1:M (M side table : add PK of 1 side)
    3. 1:1 (Any of the side : add PK of another side)
4. Descriptive attribute : Add desc attribute + PK of both



1. Student : 9 values
2. Student_Hobby : 2 values (PK of student+Hobby)

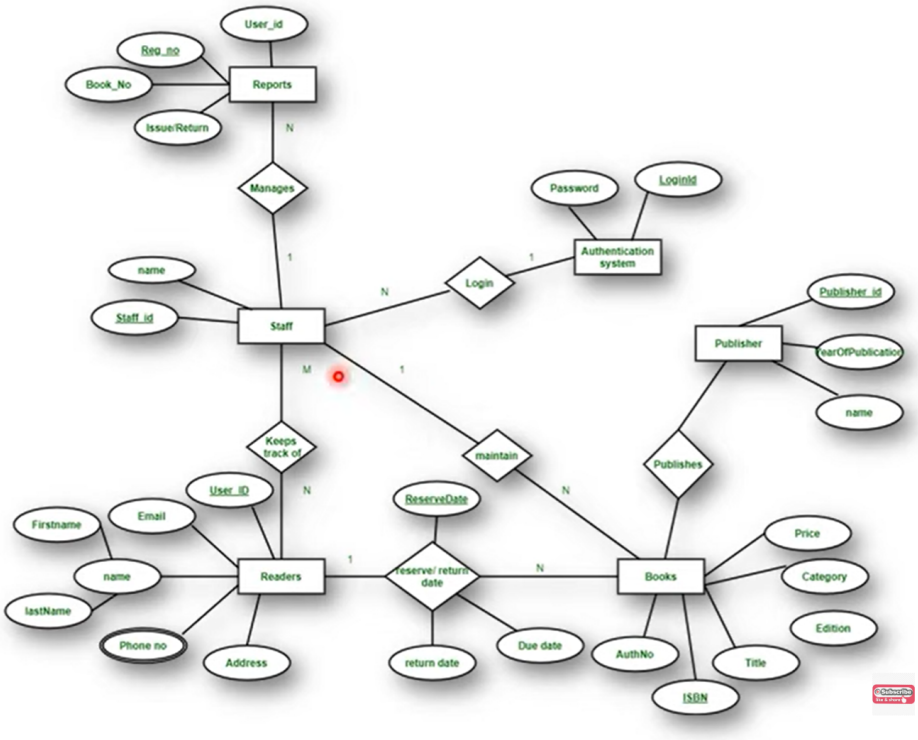
3. Course : 2
4. Subjects : 2 + pk of course
5. Lecturer : 2 + pk of student + pk of subjects

6. Attends : Student_id + Course_id
7. Takes : lec_id + course_id

8. Course_has_Subjects : PK of course+ PK of subjects

HW

ER Diagram of Library Management System



ER Diagram of Bus Ticketing System