Data: Raw and isolated facts about an entity (recorded)

Eg: Text, audio, video, map etc.

Information: Processed, meaningful, usable data.

What is data and what is information depends upon the user/observer too. {Assignments by students in college example}

Database: It is a collection of similar/related data.

DBMS: S/W used to create, manipulate & delete databases.

Disadvantages of file System

① Data Redundancy

② Data Inconsistency

③ Difficulty in accessing the data
{ Queries are not available }

④ Data isolation. { can depend on 1 particular Lang like C/C++ etc }

⑤ Security Problem { Access etc }

⑥ Atomicity Problem { 0 or 1 }

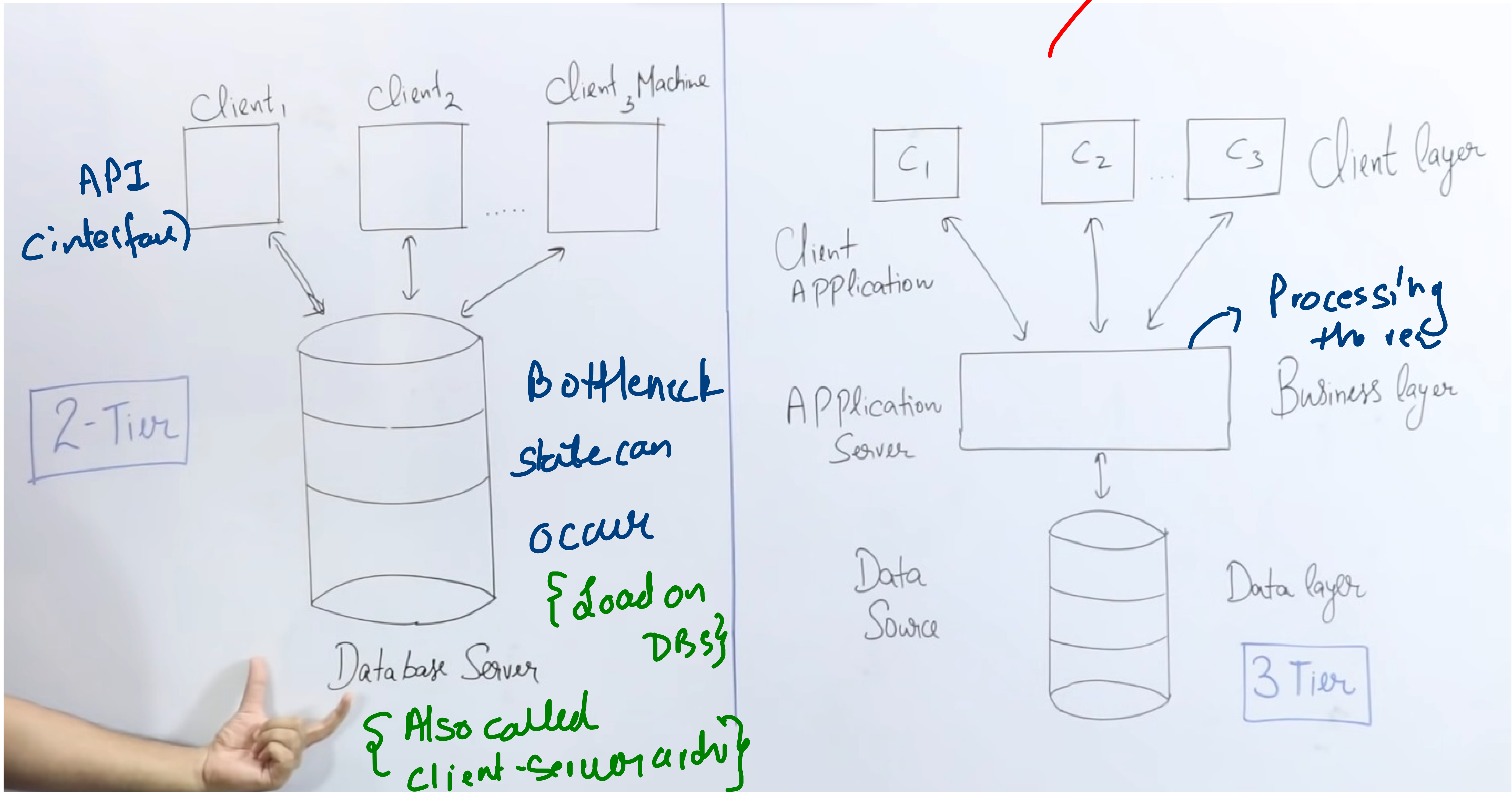
⑦ Concurrent access anomalies

⑧ Integrity problem { Some automatic triggers etc NOT possible }

OLAP	OLTP
(online analytical processing)	(online transaction processing)
→ historical data	→ current data
→ subject oriented	→ application oriented
→ decision making	→ day to day operations
→ TB, PB	→ MB, GB
→ CEO, MD, GM	→ clerks, managers
→ R 95% | 5%	→ R/W

2 Tier and 3 Tier Architecture.

Maintainance is difficult.



2 Tier Architecture Advantages: Easy to design & maintain because of limited authorized users. Limited clients & data.

Disadvantage: Not scalable. Large data and large no of clients cannot be handled. The problem is also with this large no of users accessing data at any time from anywhere. Security is also not good because of direct interaction of Client with the database server.

What is Schema?

Representing the data logically is called as a schema.

SQL → DDL commands are used to implement a schema.

{Data definition Language}

Schema can be one table or a collection of more tables.

Ques:- Define the following terms:

Relation Schema: [Table]. Represents name of the relation with its attributes.

↳ $\boxed{\text{Emp (emp\_id, ename, Age)}}$ Schema.

Relational Database Schema: Collection of one or more relation schema.

Emp (emp_id, ename, Age)
Dept (d_id, dname)
→ Company (D.B Schema)

Relation Instance: set of tuples/rows of relⁿ at a particular instance of time

emp_id	ename	Age
1	A	10
~~2~~	~~B~~	~~30~~
3	C	40

at time t_1 → I_1

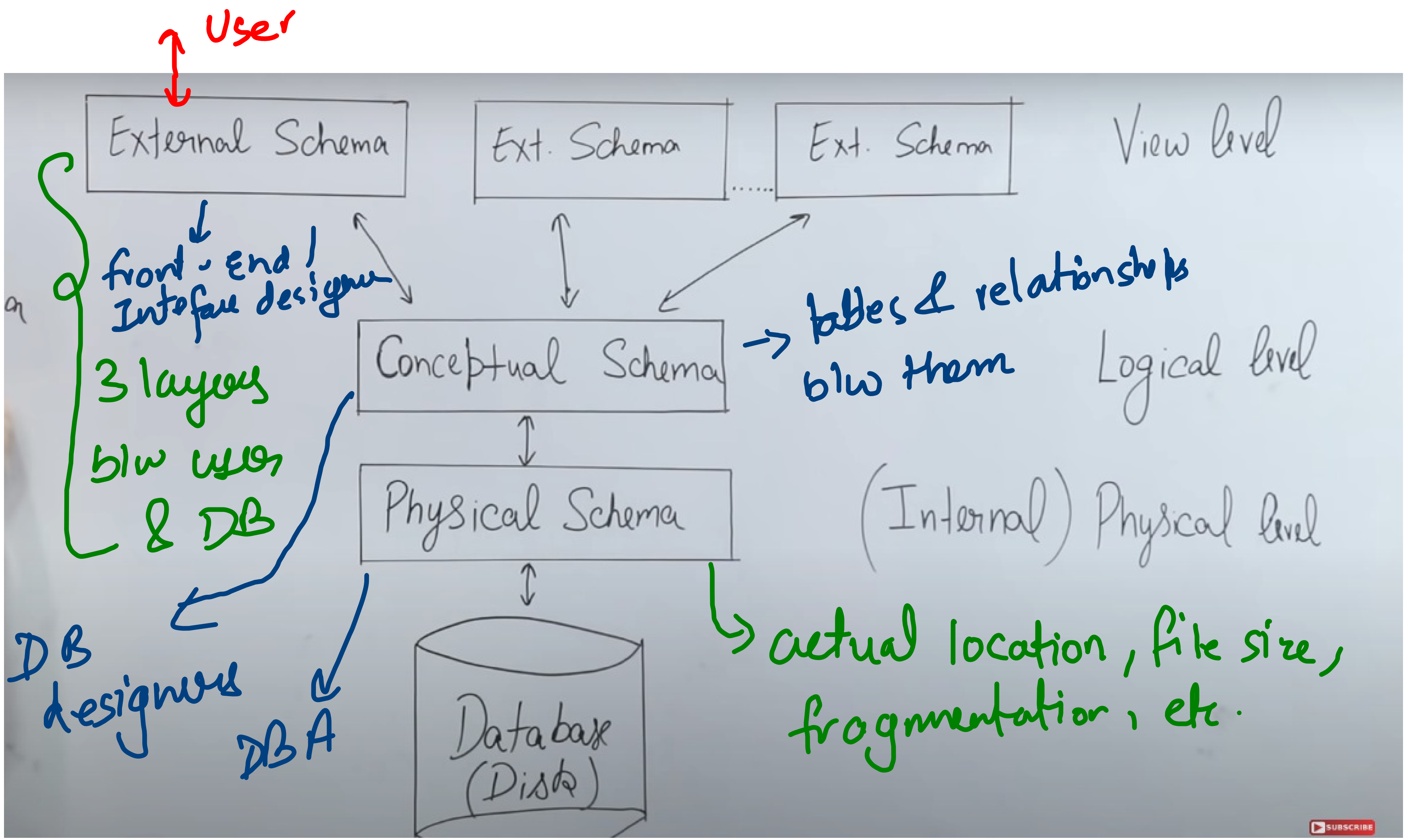
at time t_2 →

Relation Cardinality: No. of tuples/rows in a relation.

Cardinality of Emp at I_1 = 3
" " " " I_2 = 2
} no. of rows.

Relation Degree: No. of Attributes/Columns in a relation.

Three Schema Architecture (Three level of Abstraction)



User should not be able to directly interact with the data.

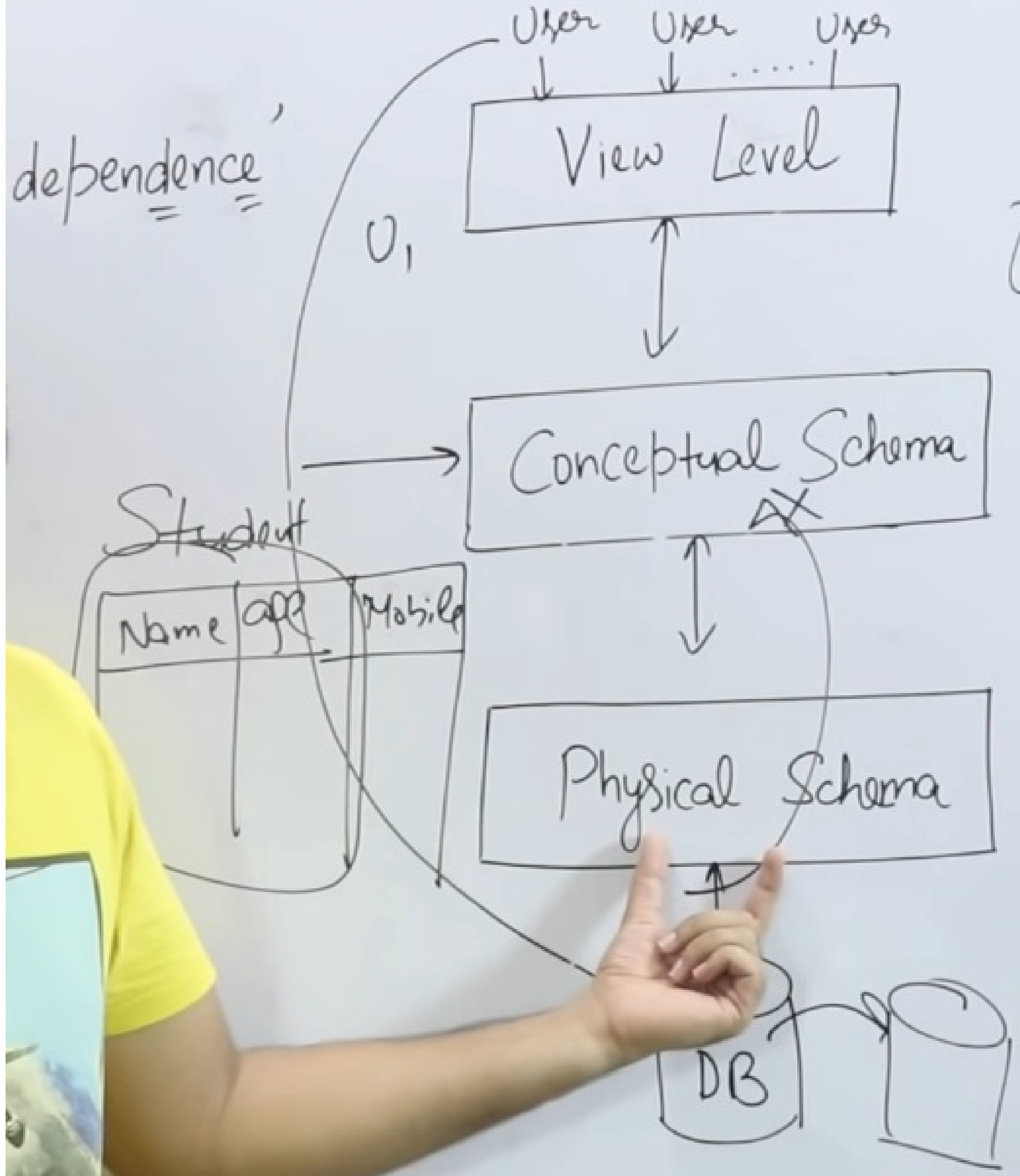
User should not know where the data is stored.

Data Independence

Make user independent of the data using 3 Layers of Abstraction.

Views can be used to implement logical data independence ie making changes in the conceptual schema do not reflect in View Level!

Physical data independence means that any change in the physical layer should not be reflected in the logical layer.
{conceptual schema}



Logical Data independence

Views.

Physical Data independence

1) Storage structure

2) DS change

3) Index

ER Diagrams

2.1 ER Diagram

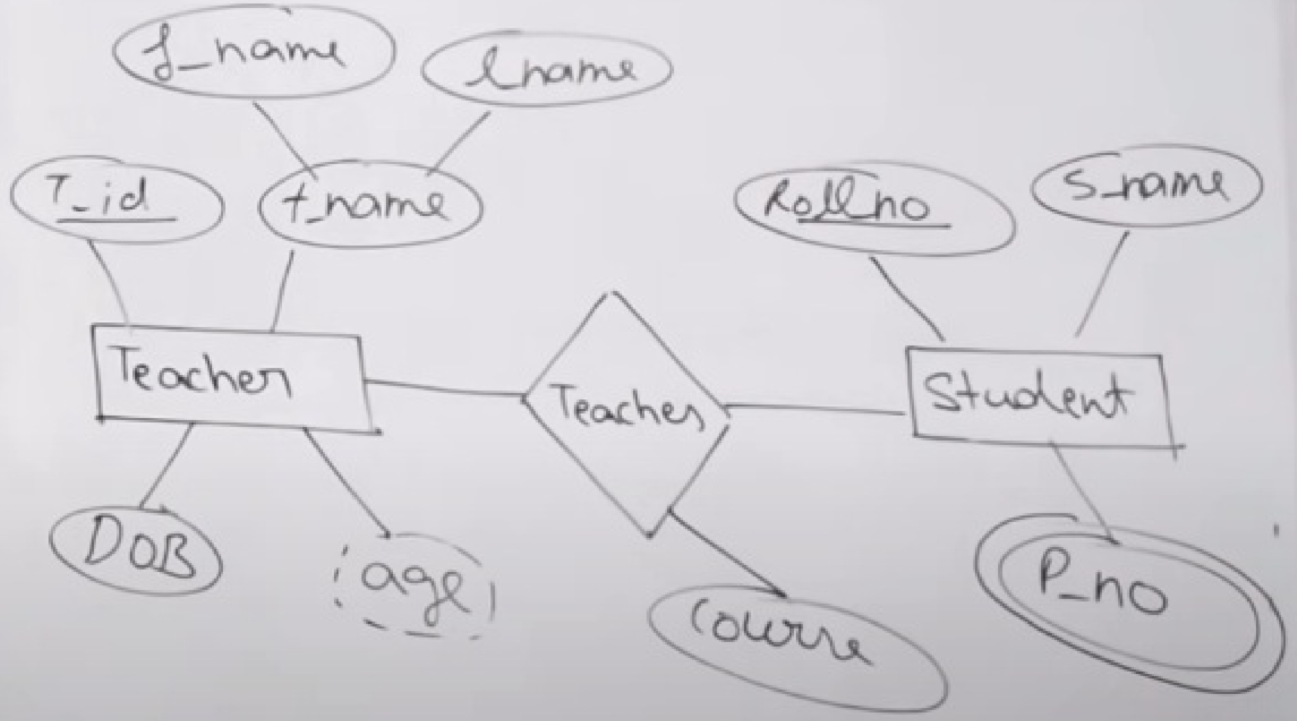
→ Introduced by Dr Peter chen in 1976.

→ A non-technical design method works on conceptual level based on the perception of real world

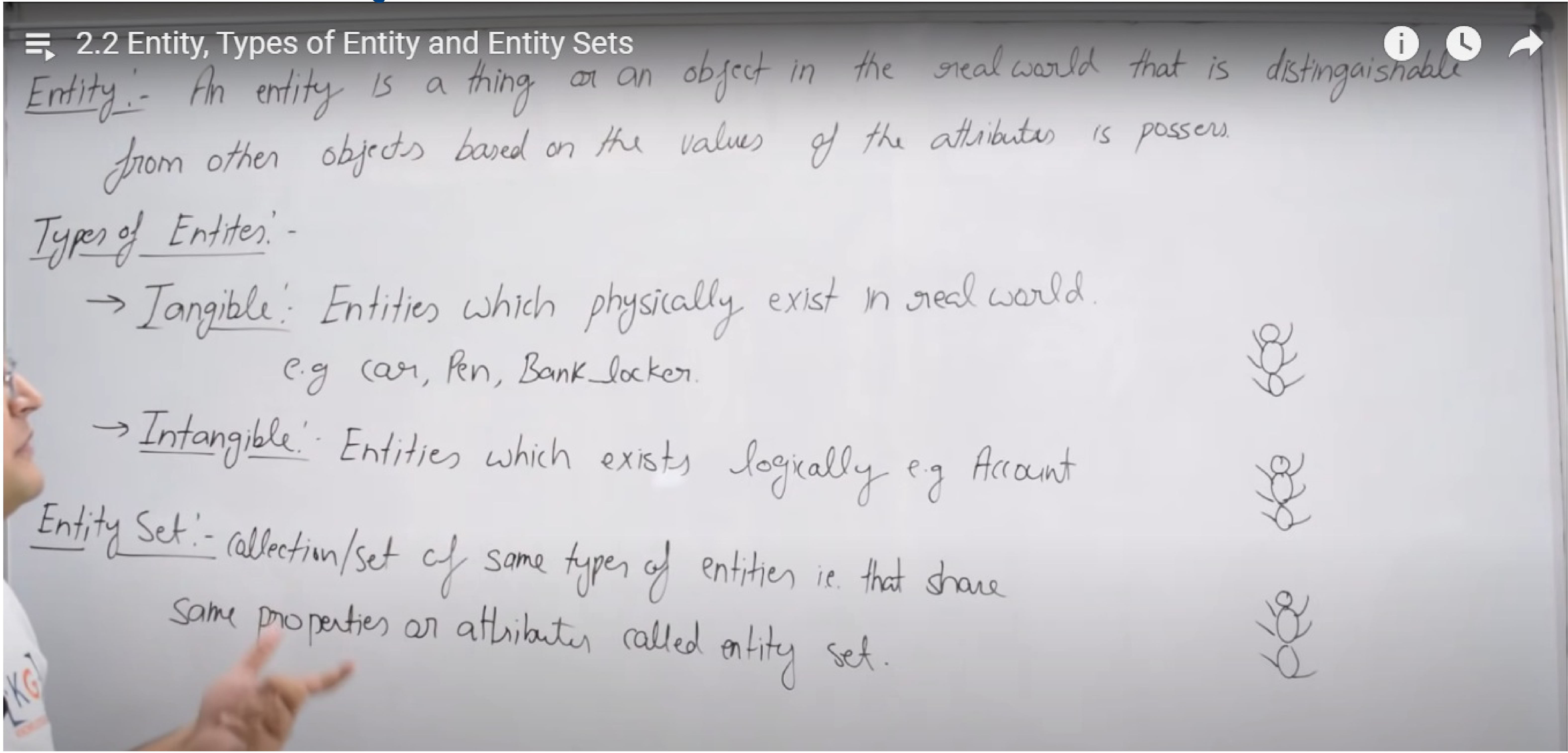
→ Consists of collection of basic objects called entities and of relationships among these objects and attributes which defines their properties.

→ free from ambiguities and provides a standard and logical way of visualizing data

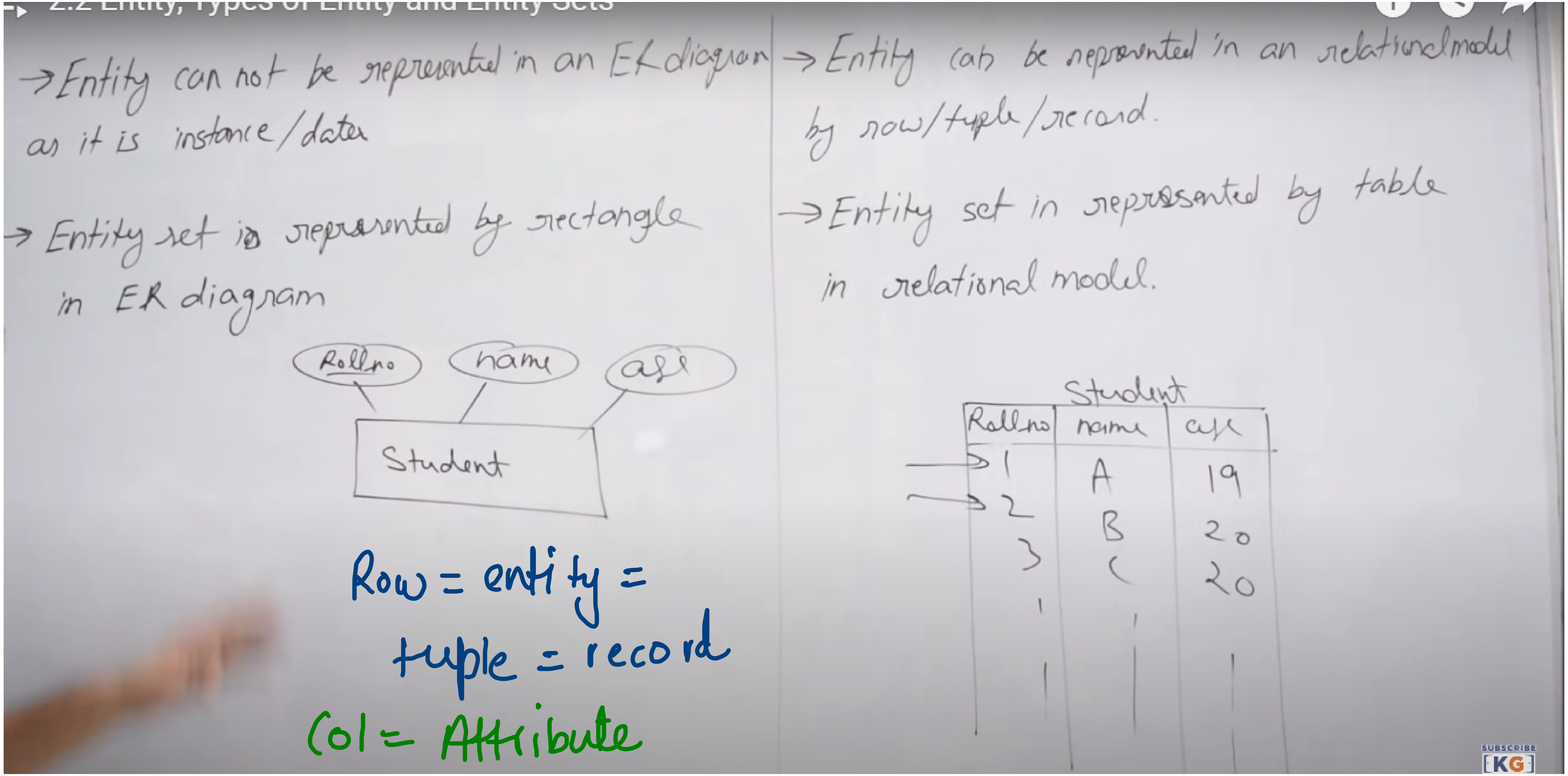
→ basically it is a diagrammatic representation easy to understand even by non-technical user.



What is Entity & diff b/w Entity & Entity Set



ER diag me entity nahi, entity set ko represent karte hai.



Row = entity = tuple = record

Col = Attribute

Types of Attributes

Attributes:- are the units that describe the characteristics of entities.

→ For each attribute there is a set of permitted values called domains.

→ In ER diagram represented by ellipse or oval, while in relational model by a separate column.

Types

Simple - composite

→ Simple cannot be divided further represented by simple oval

→ Composite can be divided further in simple attribute, oval connected to a oval

Single-multivalued

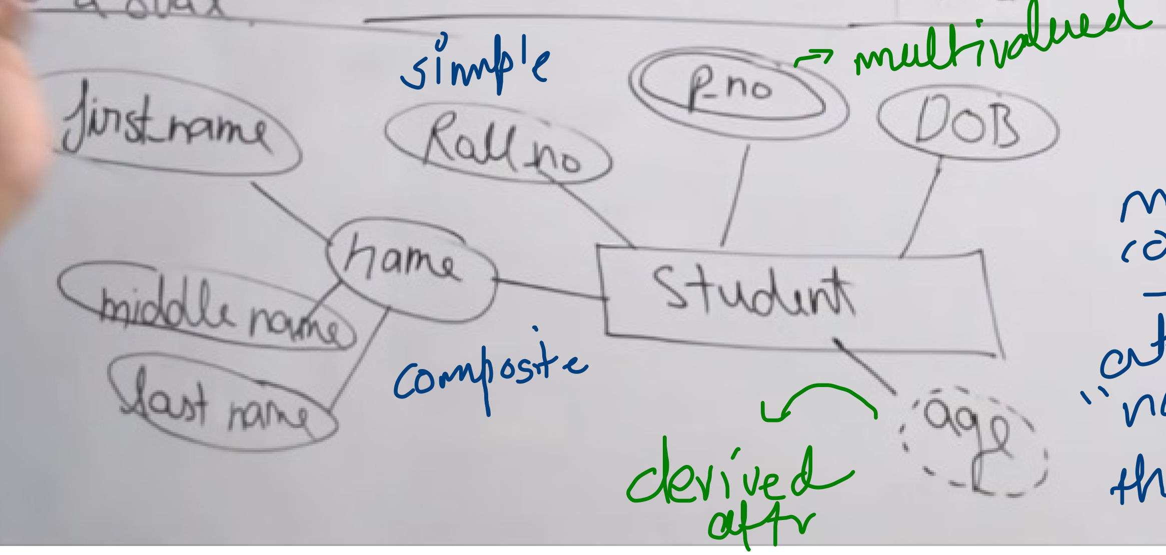
→ Single can have only one value at a instance of time.

→ multivalued → can have more than one value at a instance of time

Stored - derived

→ Stored - how value is stored in the data base.

→ Derived → how value can be computed in run time using stored attributes.



in relational model, composite attribute "name" is not there.

Student

Roll no	f_name	M_name	L_name	DOB	Age

Roll no	P_no

value will be computed during Run Time.

Relationship :-

Relationship :- is an association between two or more entities of same or different entity set.

→ no representation in er diagram as it is an instance or data.

→ In relational model represented either using a row in a table.

Relationship type/set :- a set of similar type of relationship

→ In er diagram represented using a diamond

→ In relational model either by a seperate table or by seperate column (foreign key)

→ Every relationship type has three components

(i) Name

(ii) Degree No of entity sets participating in a relationship

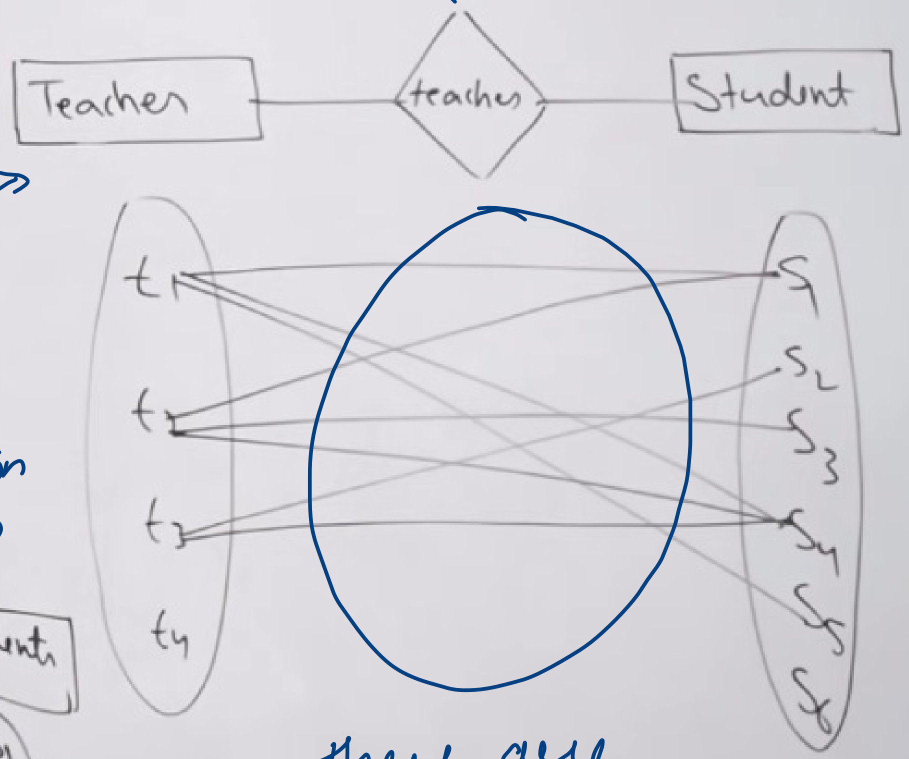
(iii) Cardinality ratio / Participation constraints

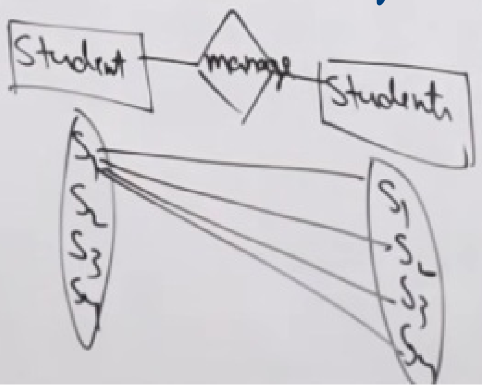
1 to Many

Many to Many etc.

Binary Relation

this is a relation type or relation set

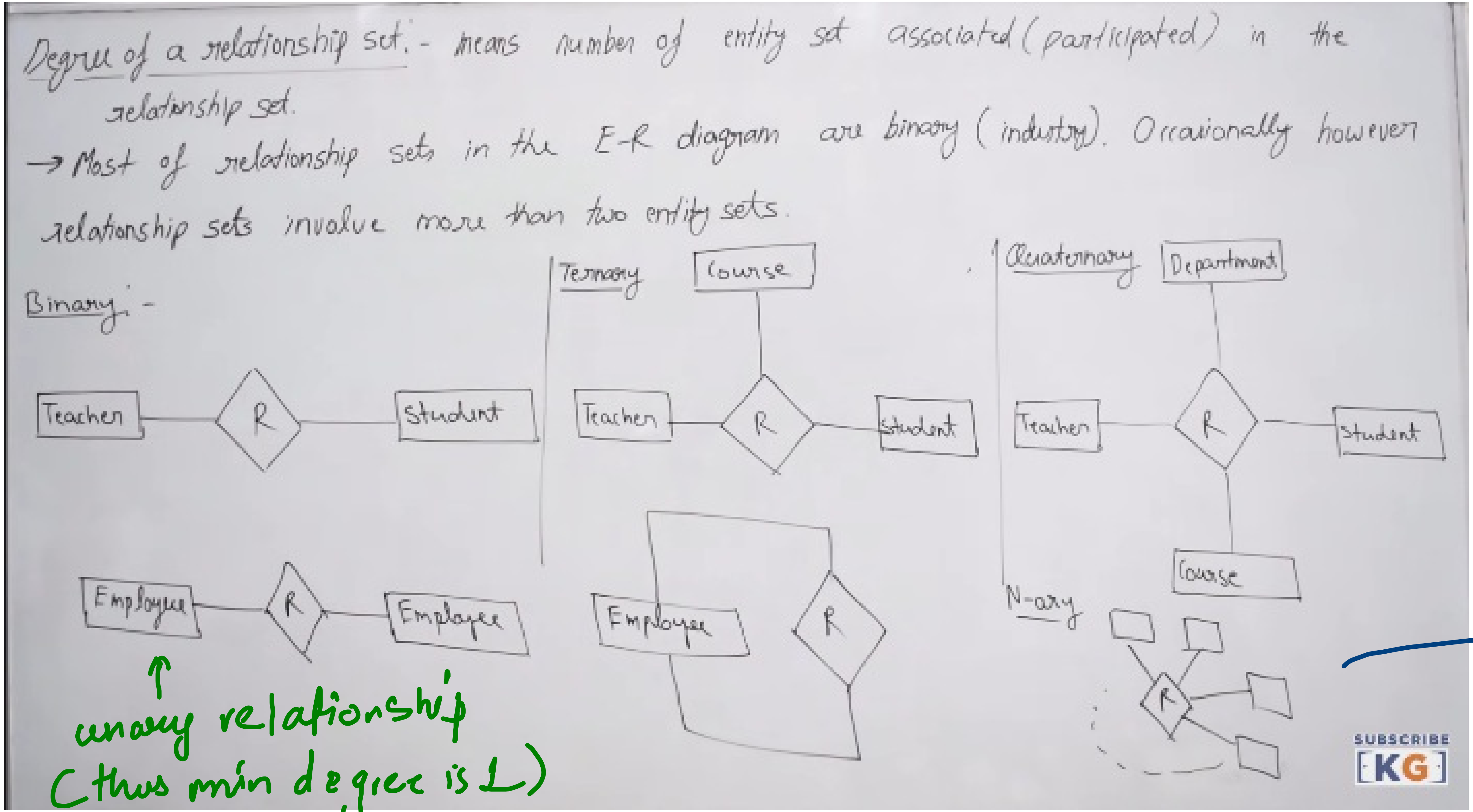




these are all relations

We know the diff b/w relationship & relationship type. However, we will say relationship only most of the times.

Degree of a relationship set



↑ unary relationship (thus min degree is 1)

→ Theoretically it is possible

Mapping Cardinalities and Cardinality Ratio.

Mapping Cardinalities/Cardinality ratio:- Express the number of entities to which other entity can be related via a relationship.

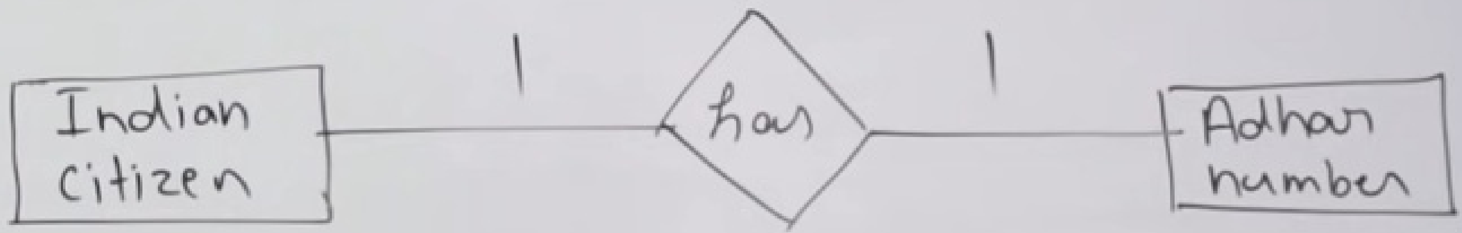
→ can be used in describing relationship set of any degree but is most useful in binary relationship

→ 1 : 1 (one to one)

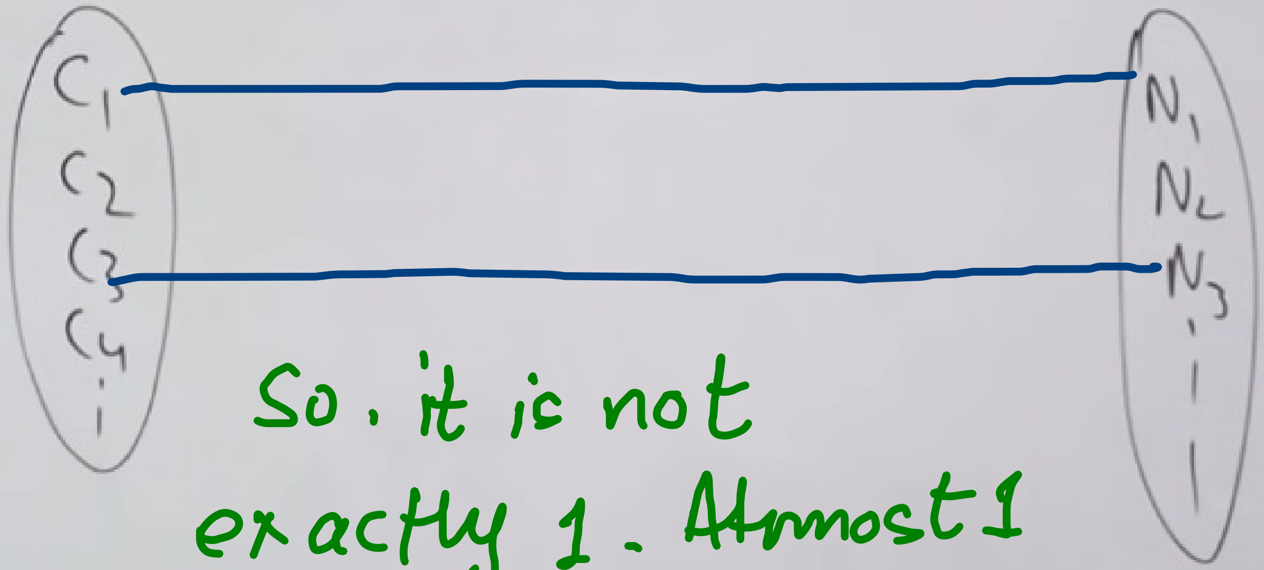
→ 1 : N (one to many)

→ N : 1 (many to one)

→ M : N (many to many)



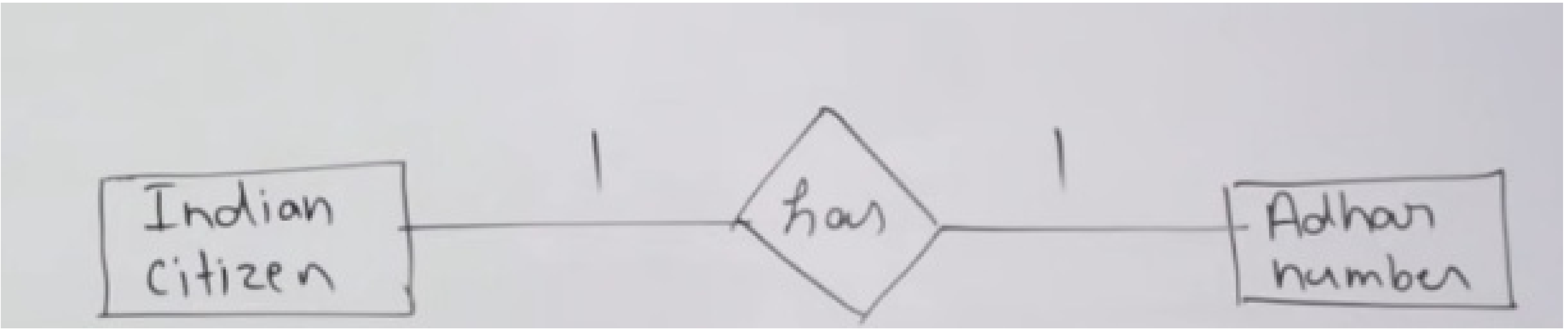
One to one relationship

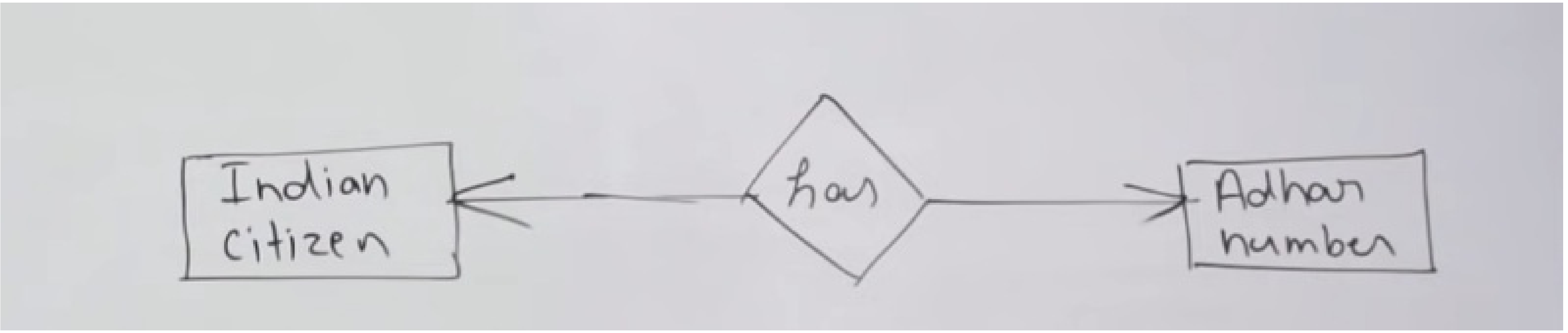


So. it is not exactly 1. Atmost 1

i.e some entities might not have any relation added!

One to One

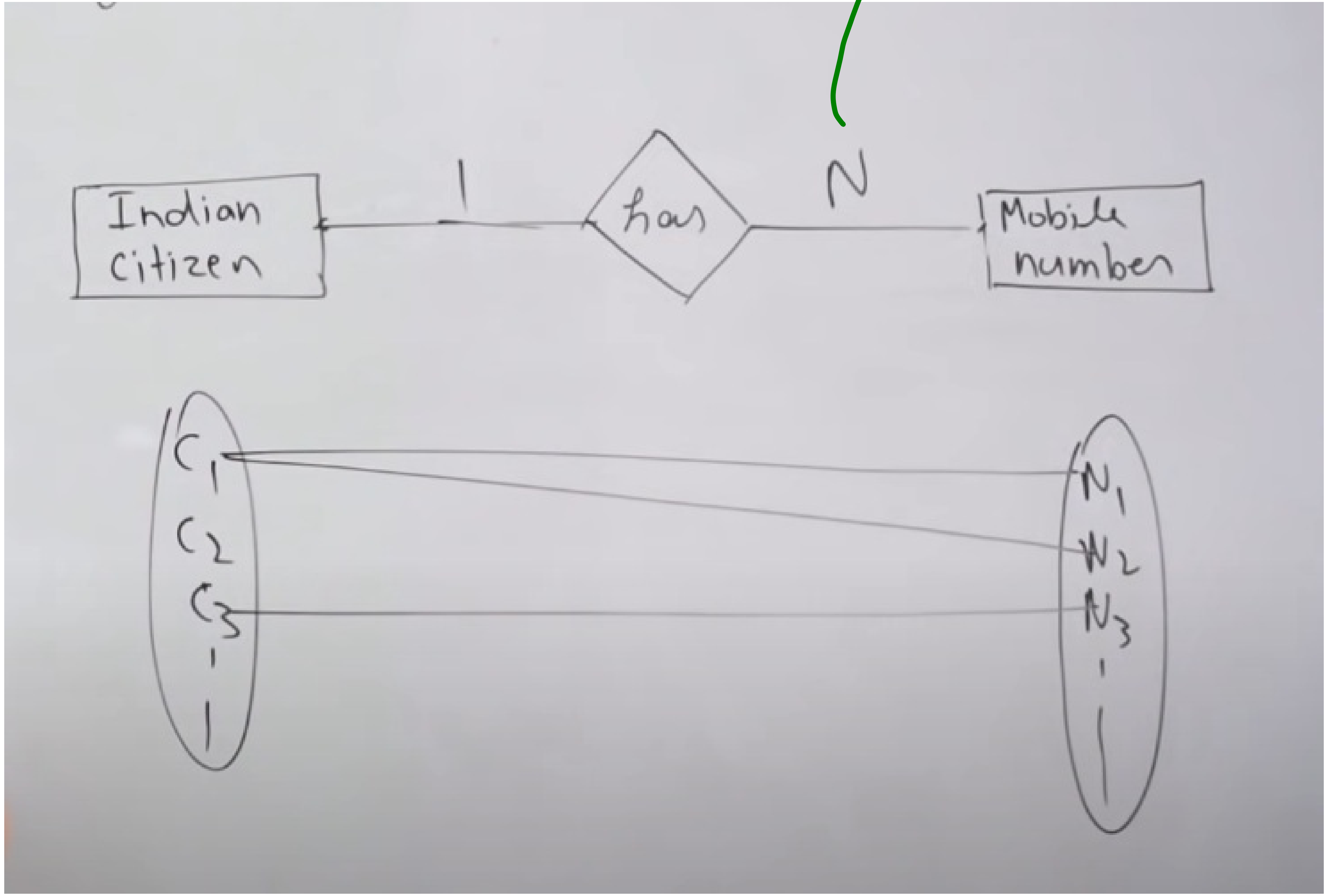




Ya to 1 likha hua hona chahiye ya edges dikhni chahiye.

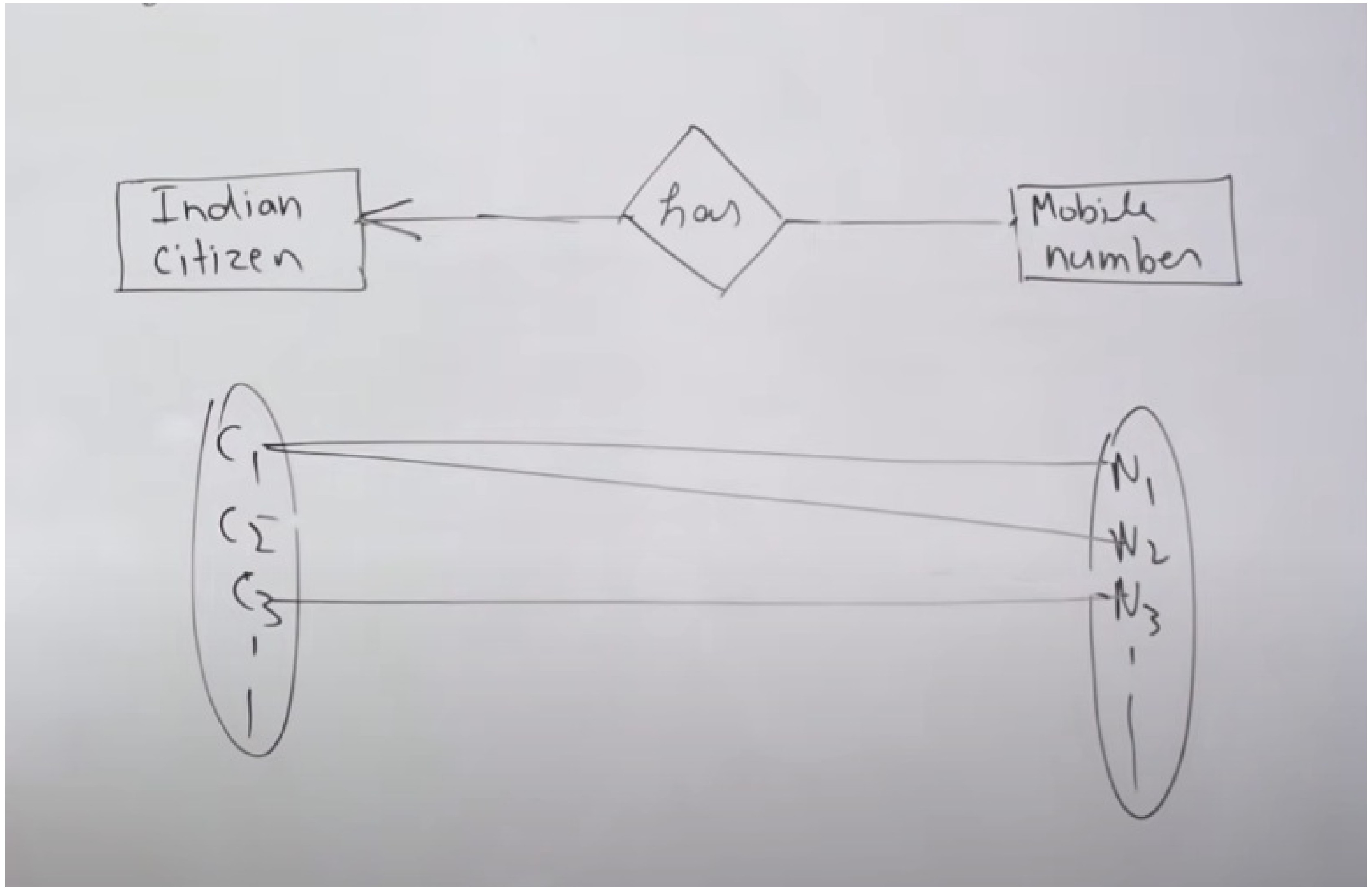
One to Many

N ki jagah * bhi likha mil sakta hai.



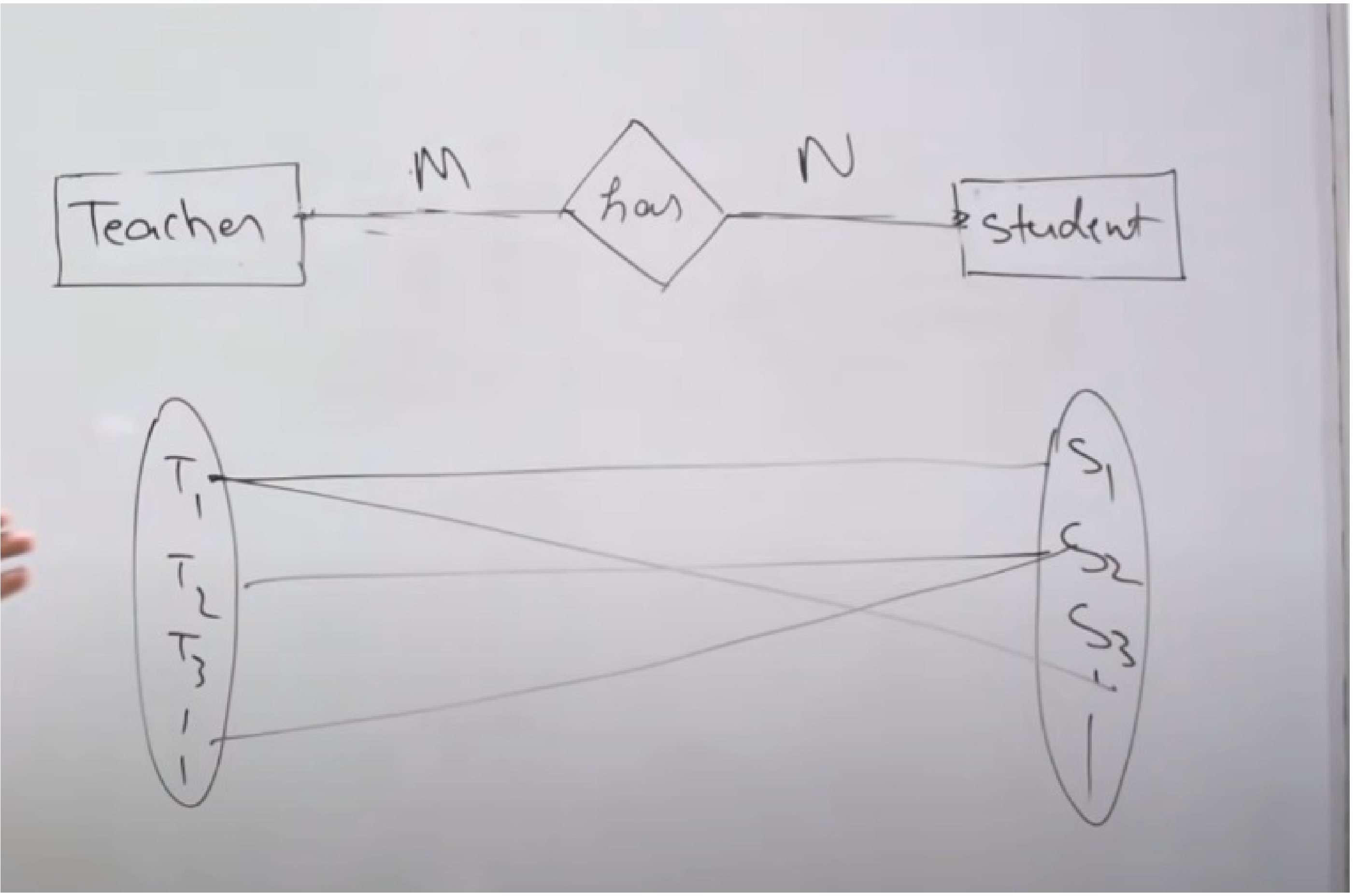
So, every entity of 1st entity set can relate to atmost N entities in the second entity set.

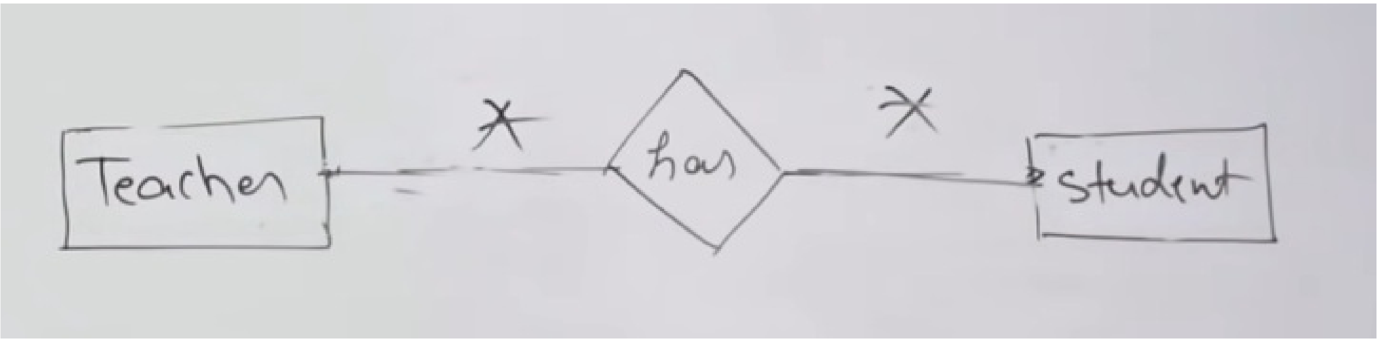
Again, ya to 1 & N likha hua hona chahiye like in prev page. Else jis side edge hai vo 1 & other side many.

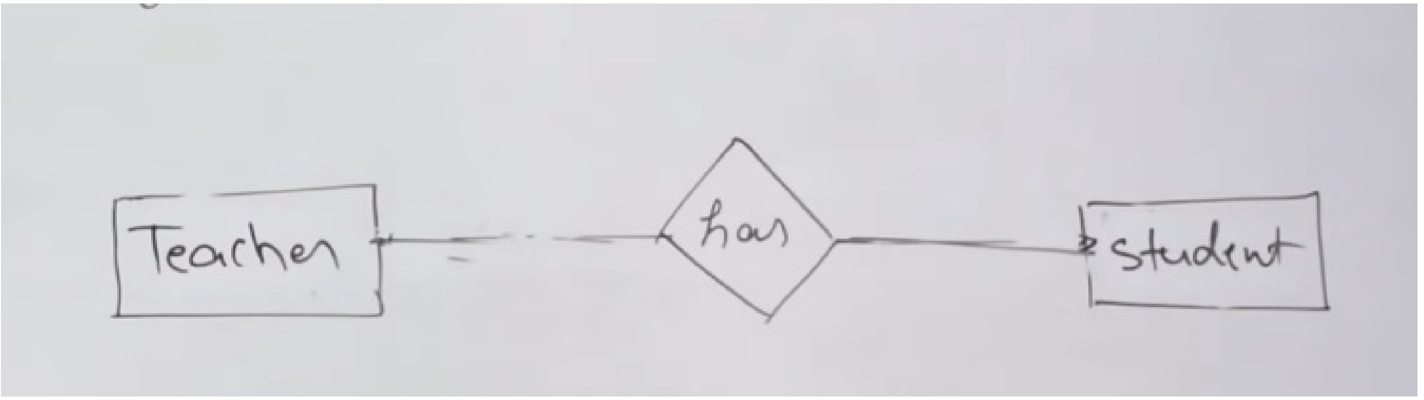


* Every 1 to 1 relationsip is a 1 to Many as well as Many to One Relationship

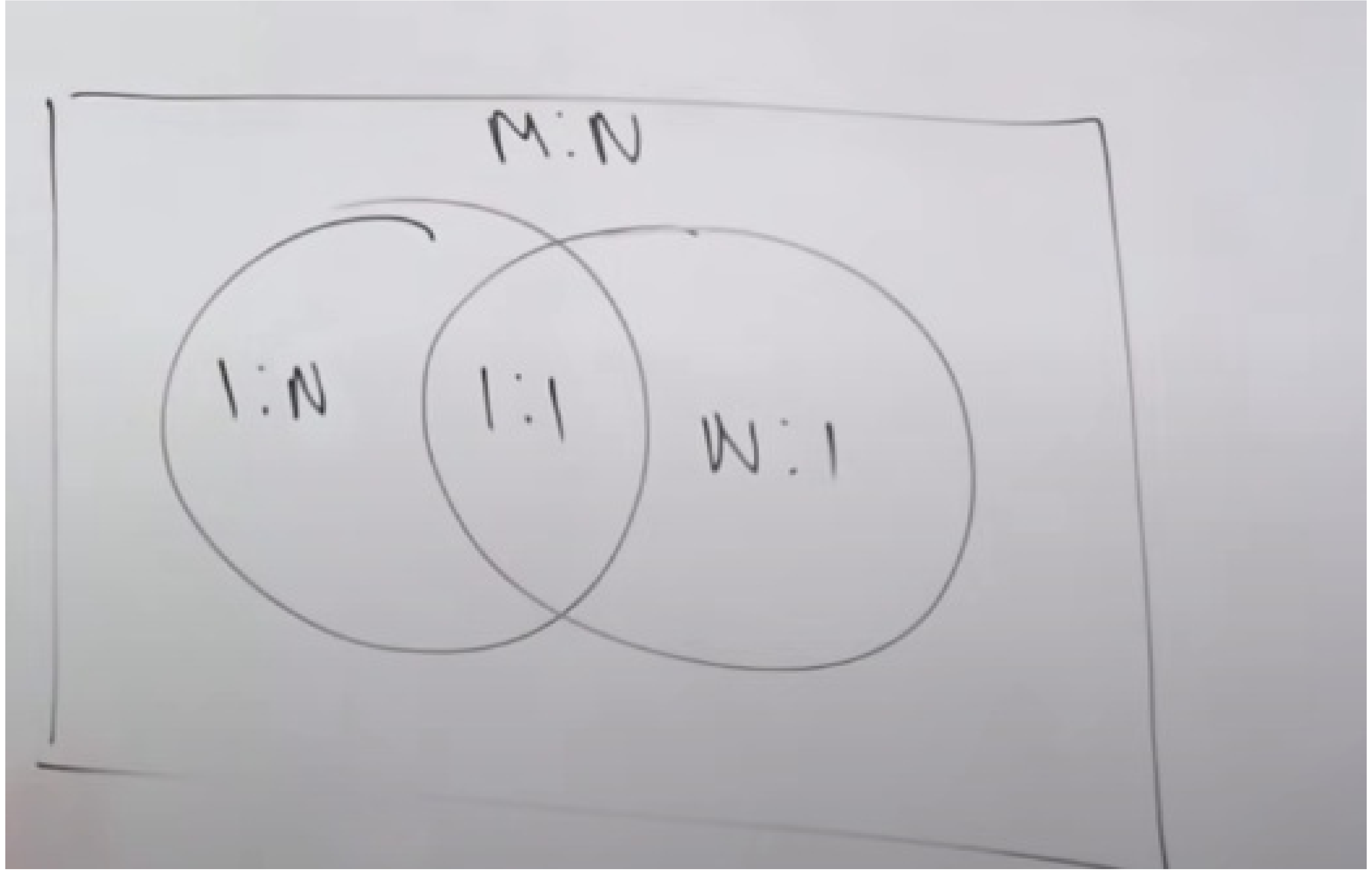
Many to Many







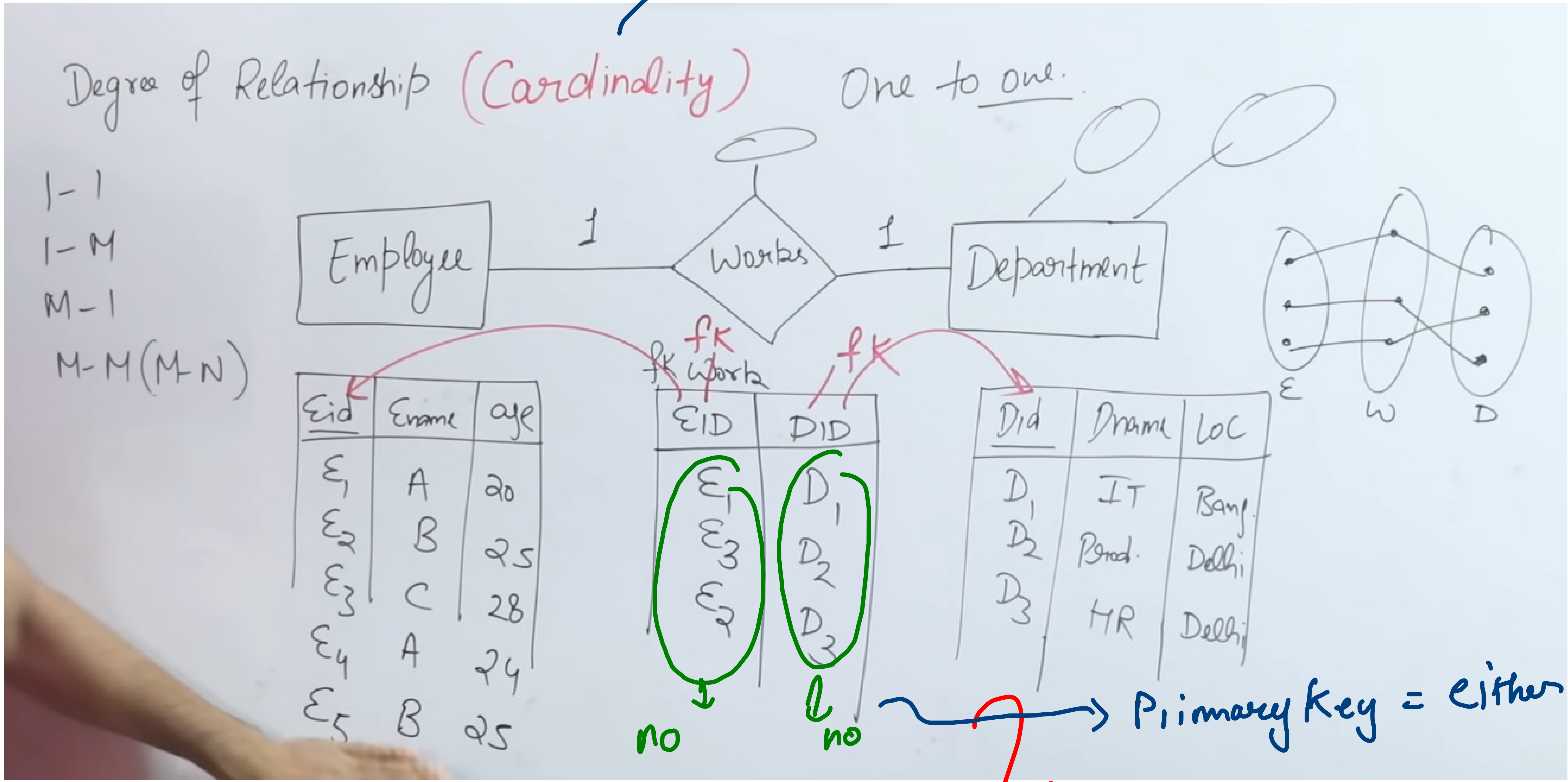
Representations.



Many to Many covers all other relationships.

One to One

This is written wrong as degree & cardinality are different.

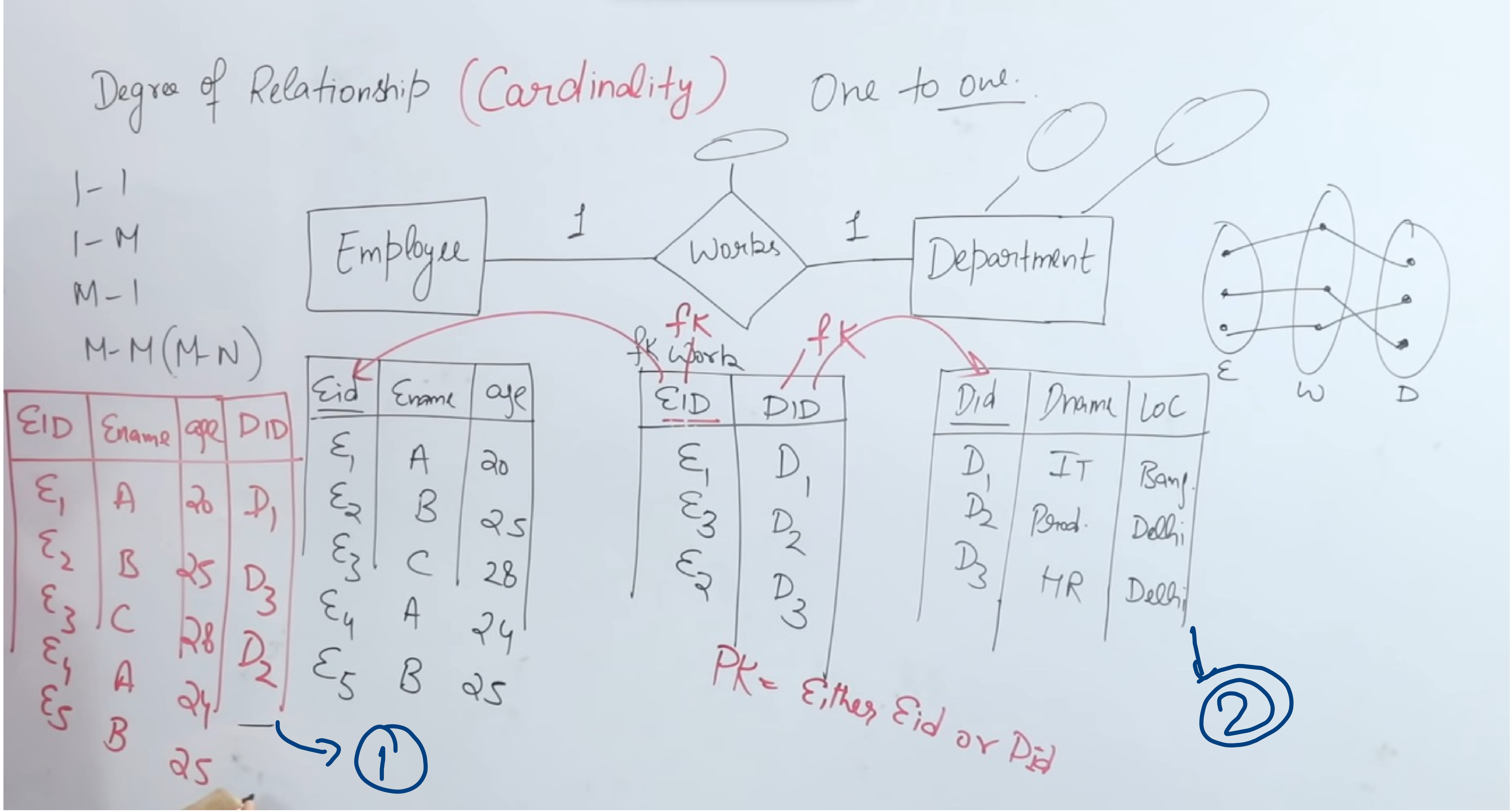


no no

Primary Key = either EID or DID

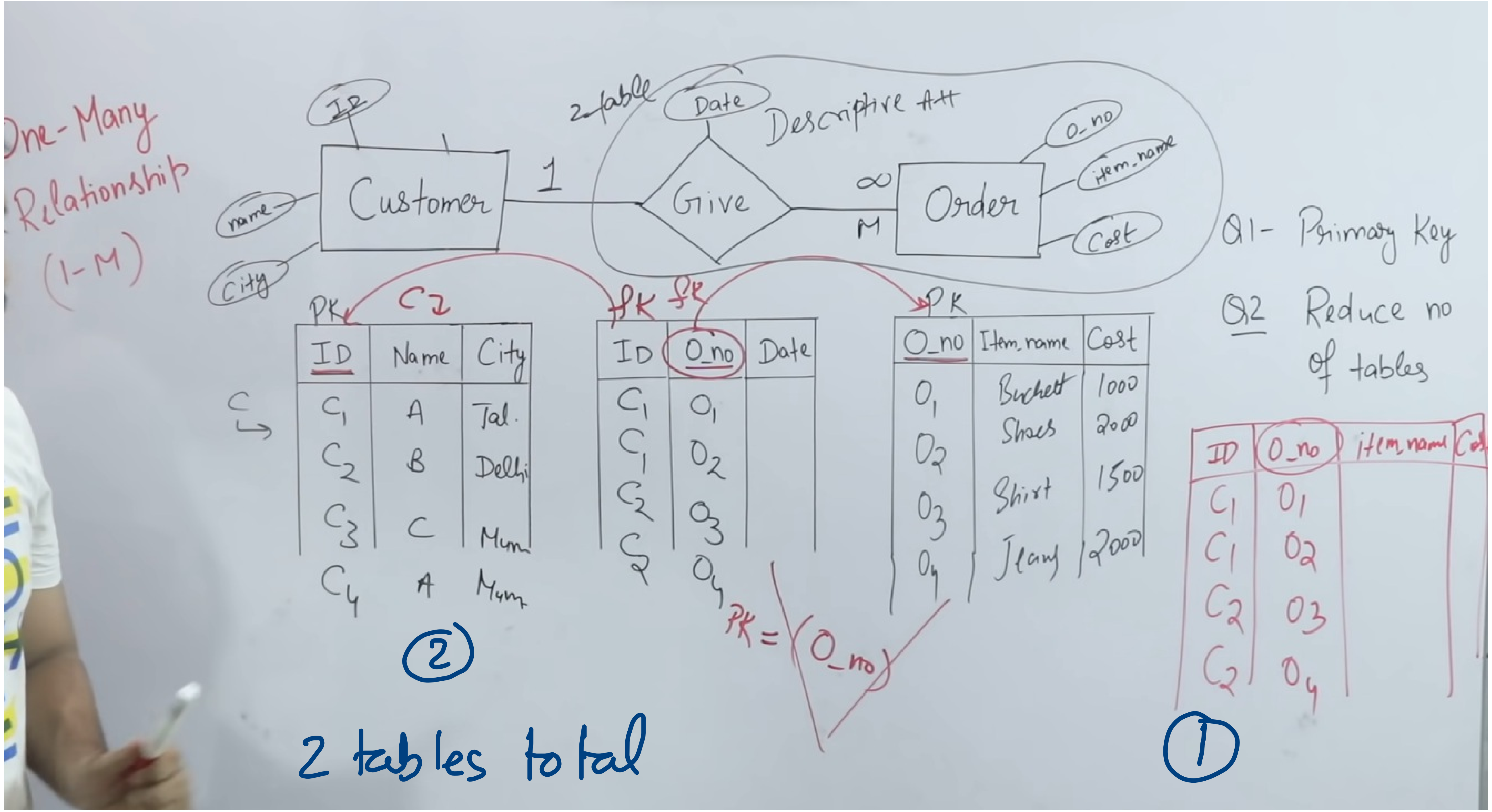
repetition repetition } because 1 to 1 relation

So, the tables can be merged. finally, we have 2 tables only -



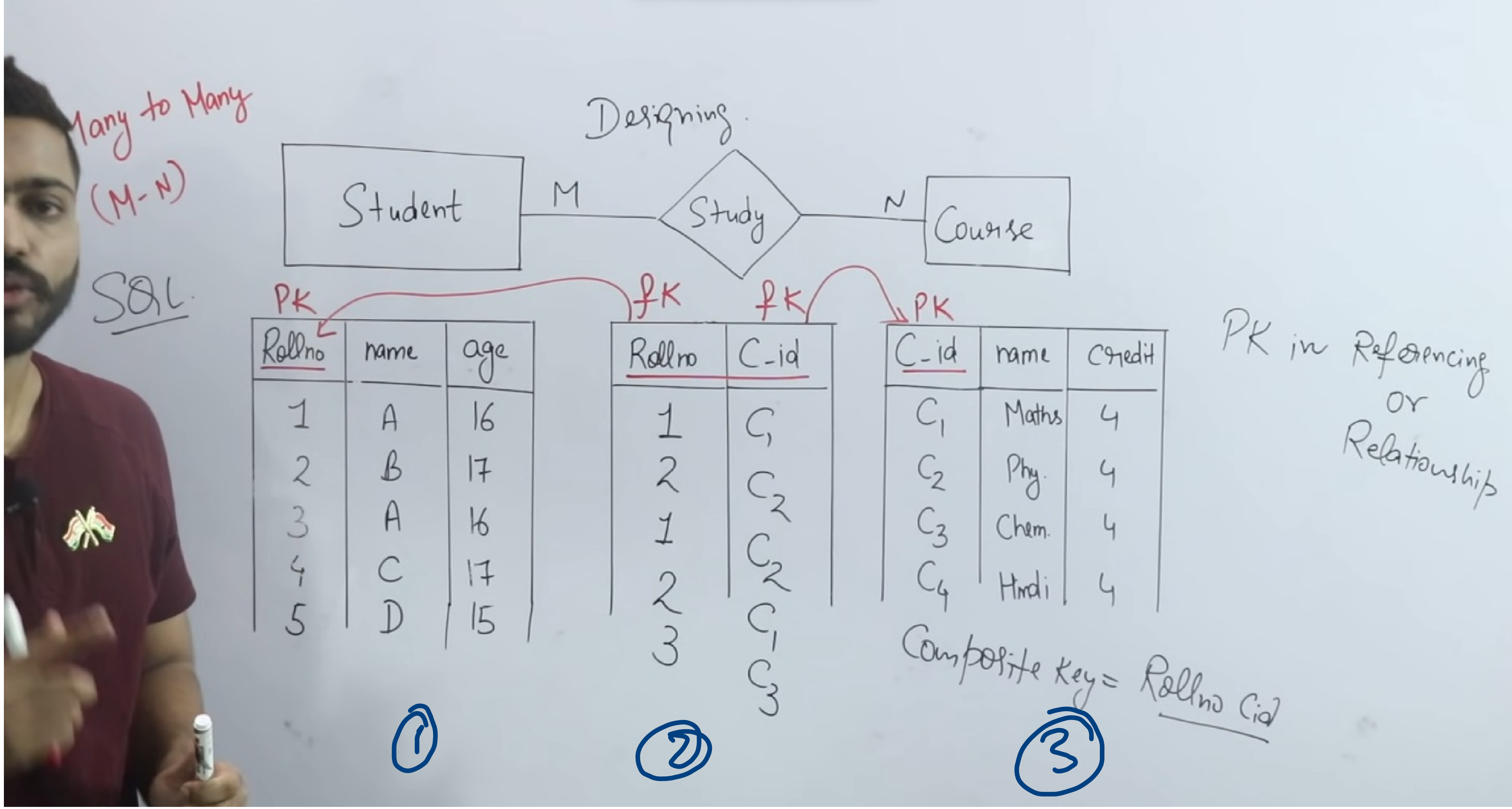
One to Many Relationship

→ Primary Key of the Relationship table will always be the primary key on the Many side.



Many to Many

→ Here, we have composite PK.



Hence, no of tables can't be reduced.

Participation Constraints.

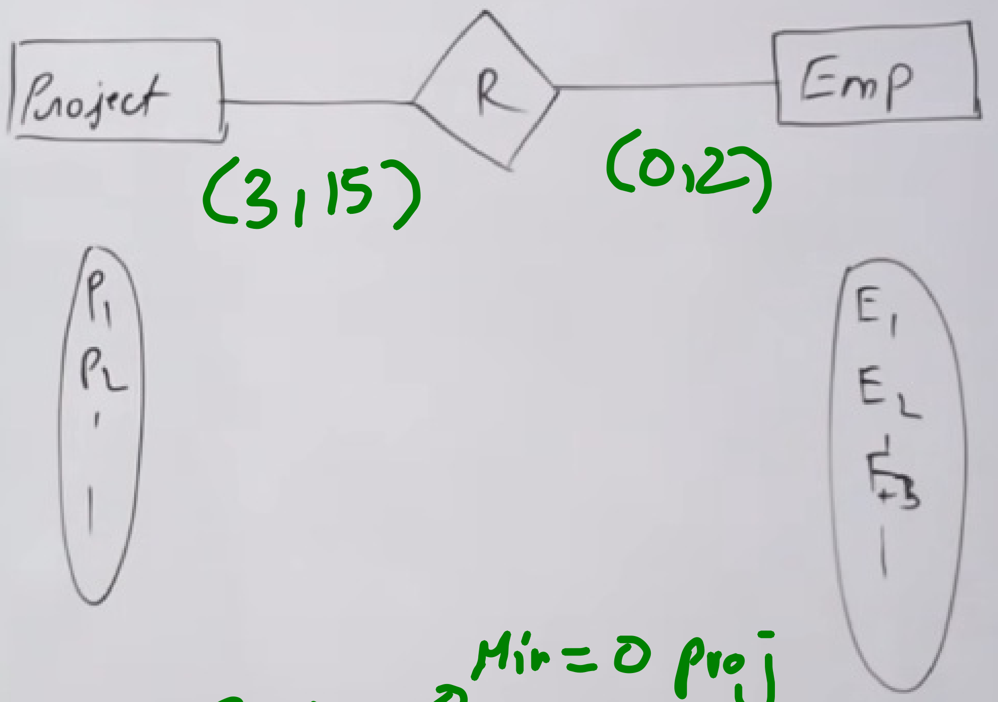
Project → Employee → Total Participation
as each proj has min 3emp

Participation Constraints: - Specifies weather the existance of an entity depends on its being related to another entity via a relationship type.

These constraints specify the minimum and maximum number of relationship instances that each entity can/must participate in.

Max cardinality: - it defines the maximum number of times an entity occurrance participating in a relationship.

Min cardinality: - it defines the minimum number of times an entity occurrance participating in a relationship



→ Partial participation

→ Total participation

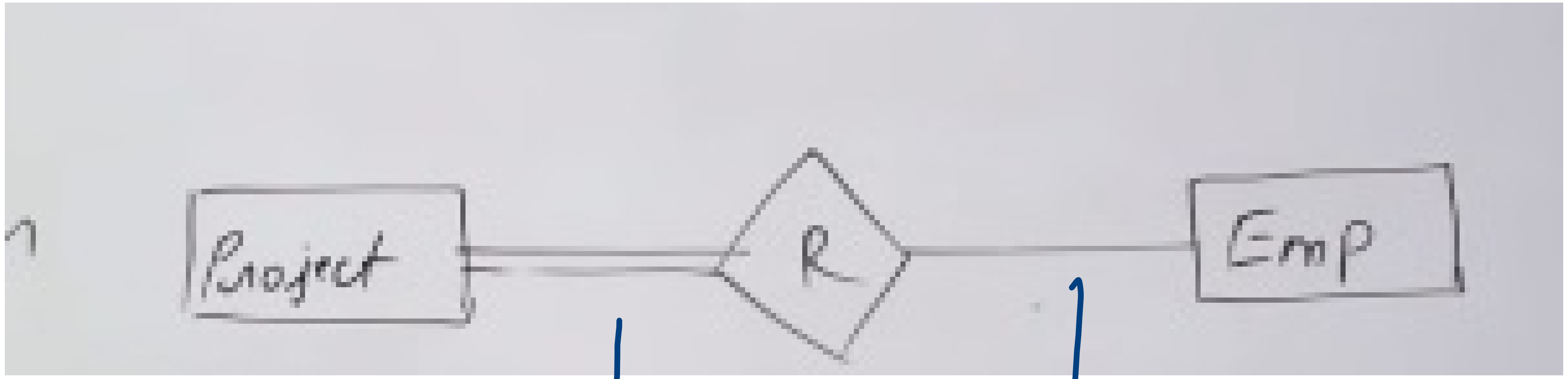
Project → Min 3 emps
Project → Max 15 emps

Emp → Min = 0 proj
Emp → Max = 2 proj

Employee → Project ⟶ Partial Participation has employee can have min 0 projects.

If Min cardinality is 0 $\Rightarrow$ partial participation

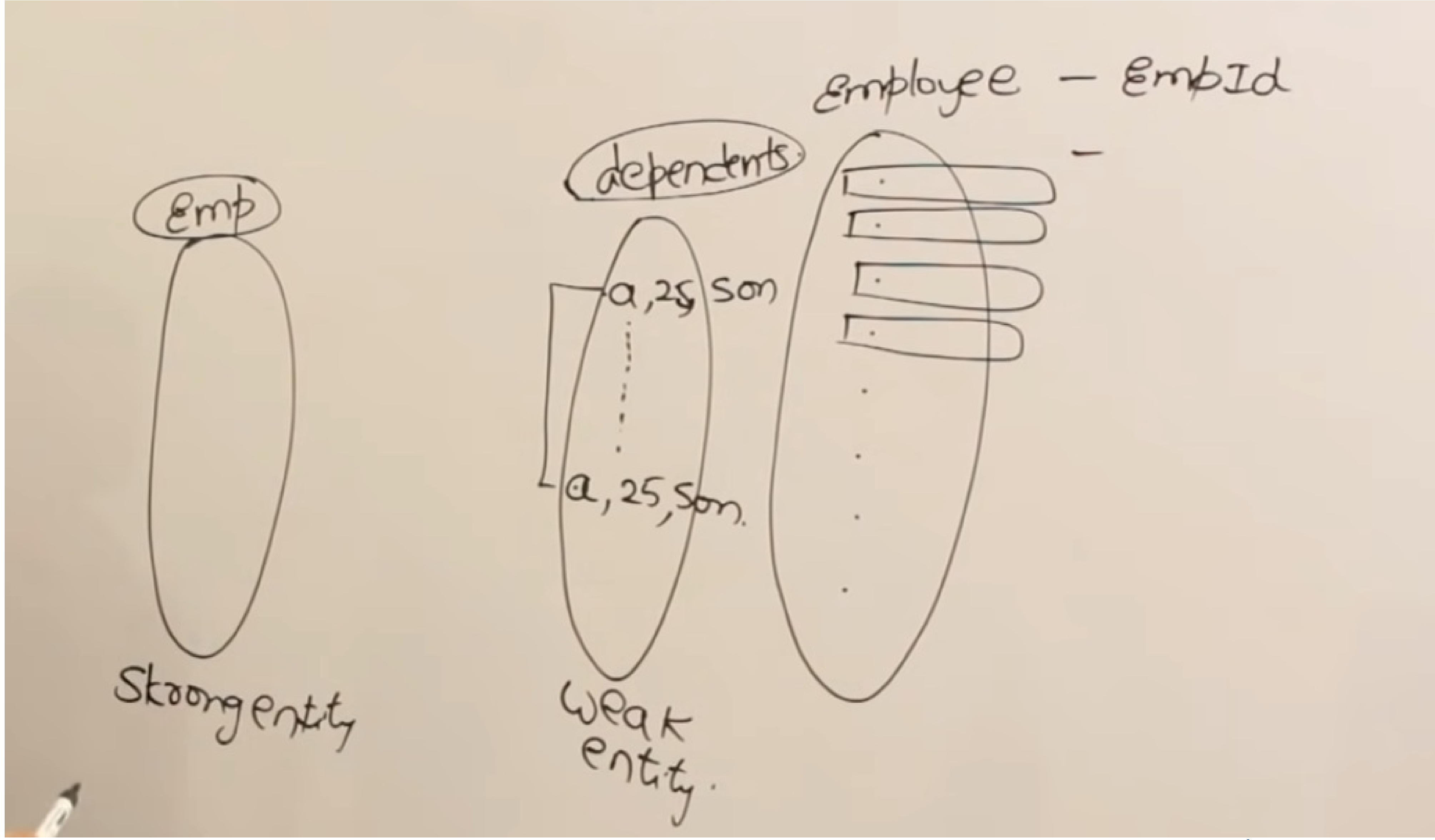
If Min cardinality ≥ 1 $\Rightarrow$ total participation.



total
Participation

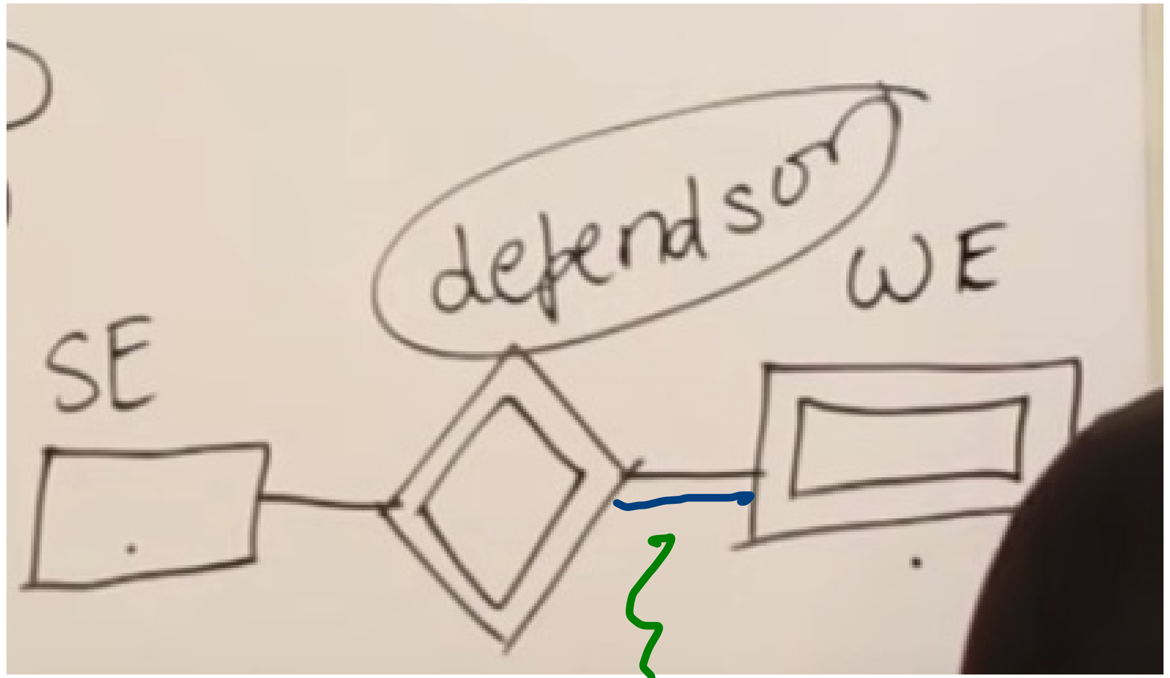
Partial
participation.

Weak Entity



Any entity that does not have a key is called a weak entity.

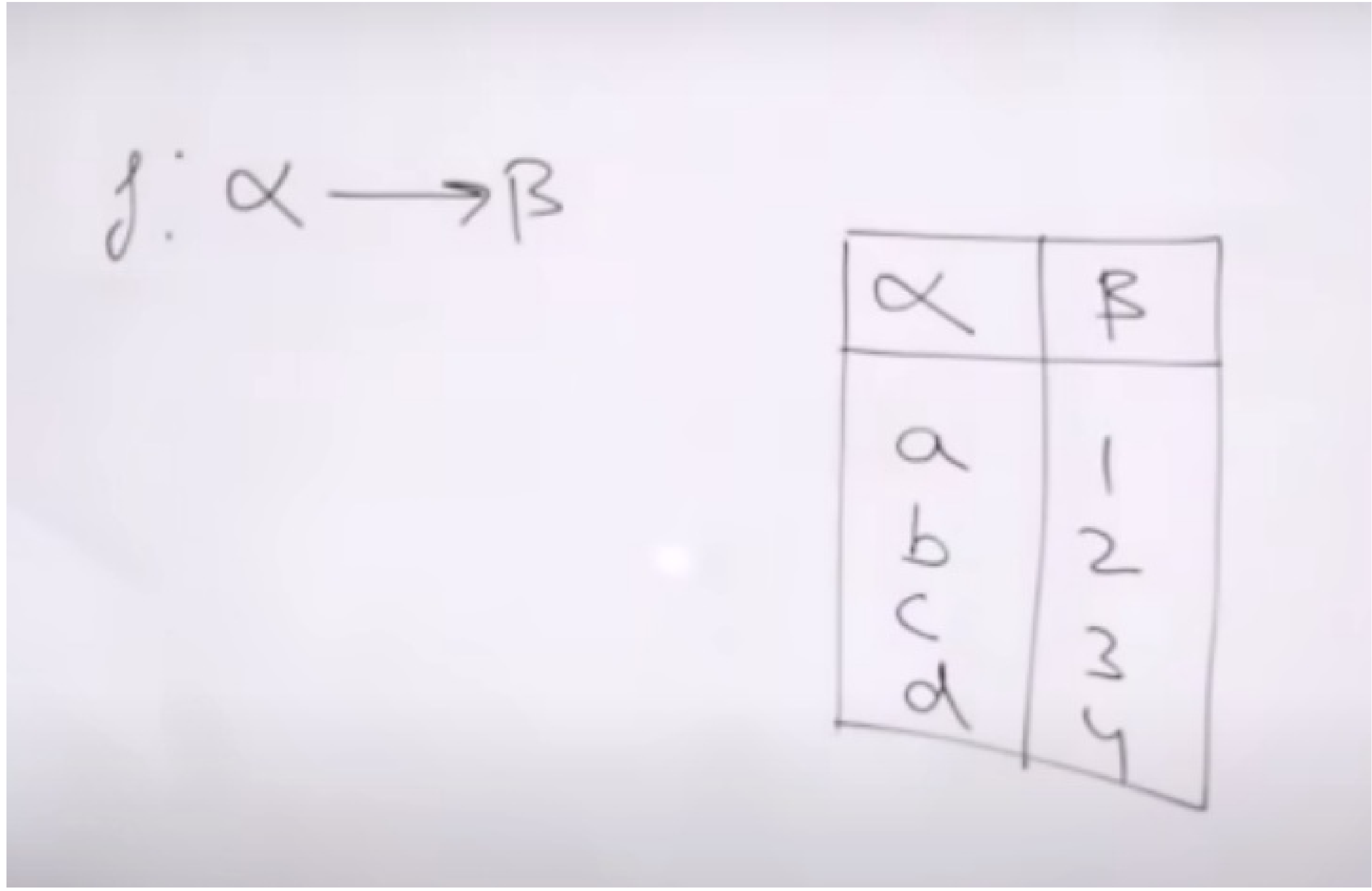
To overcome it, every weak entity must be related to a strong entity.



** However, every total participation might not be a weak entity.

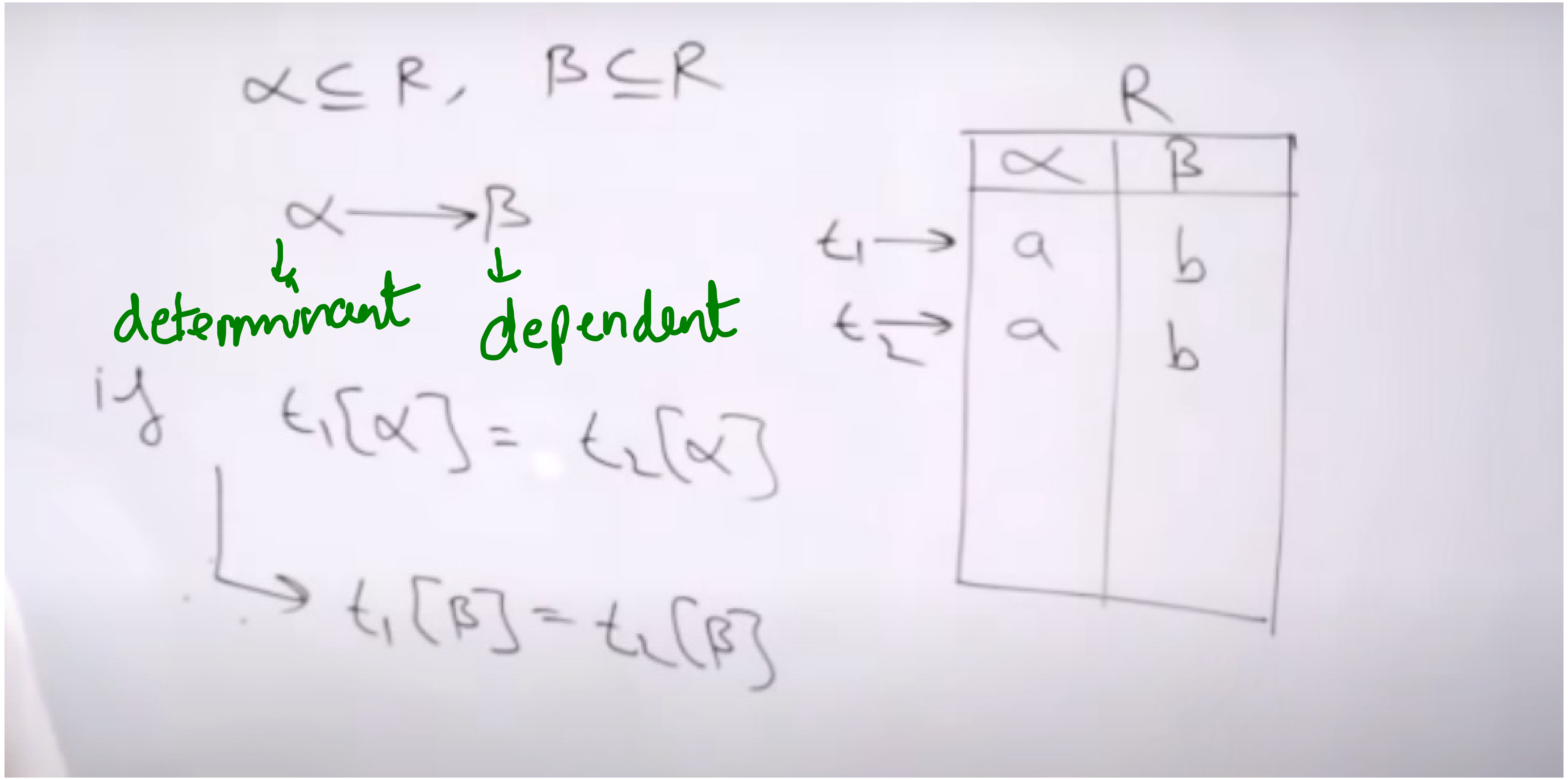
always total participation { A weak entity will always participate totally in a relationship }

Functional Dependency in DBMS



If there is a functional dependency from $\alpha \rightarrow \beta$ this means that we can search the value of β if we know the value of α.

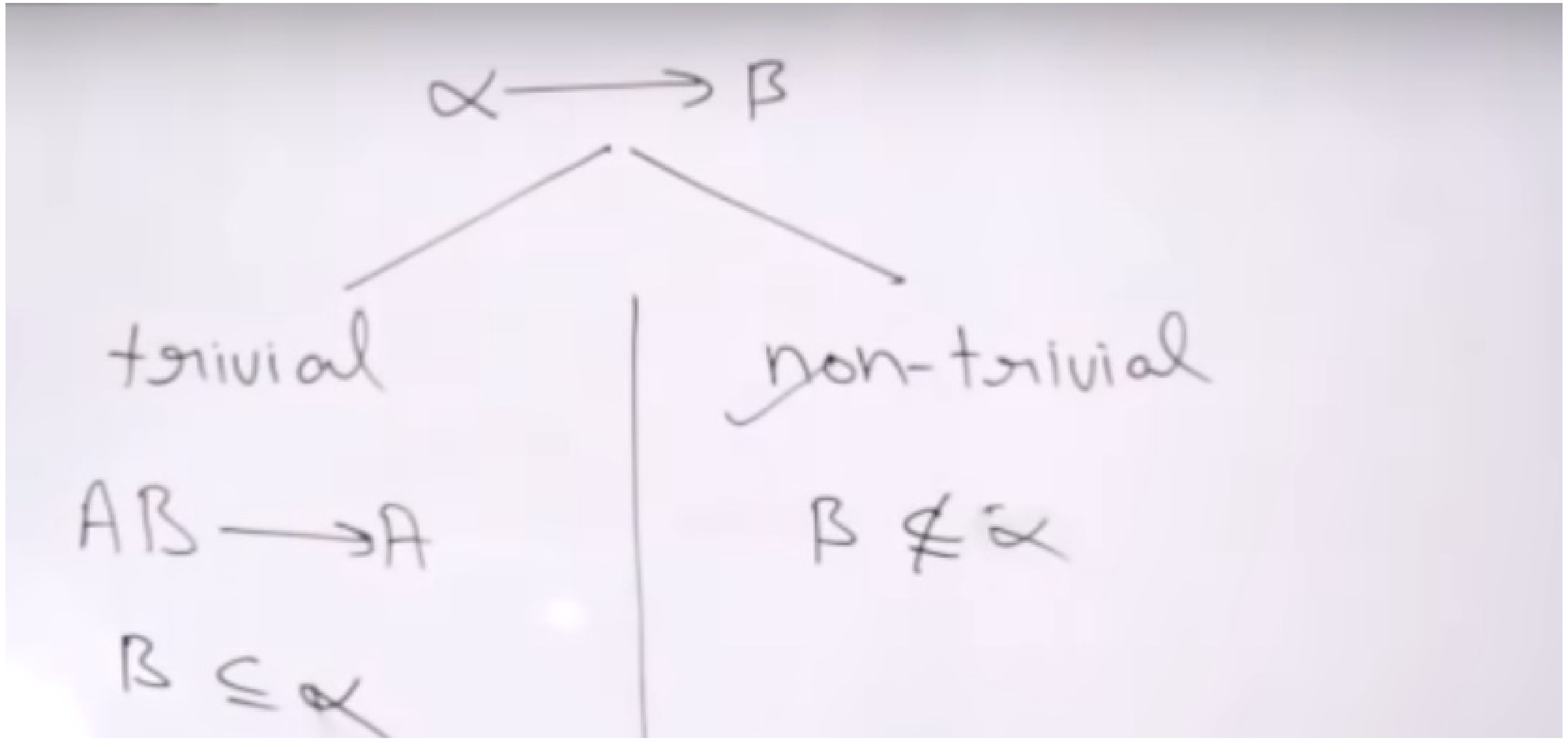
condition is α has only 1 occurance.



α	β
a	b
c	b

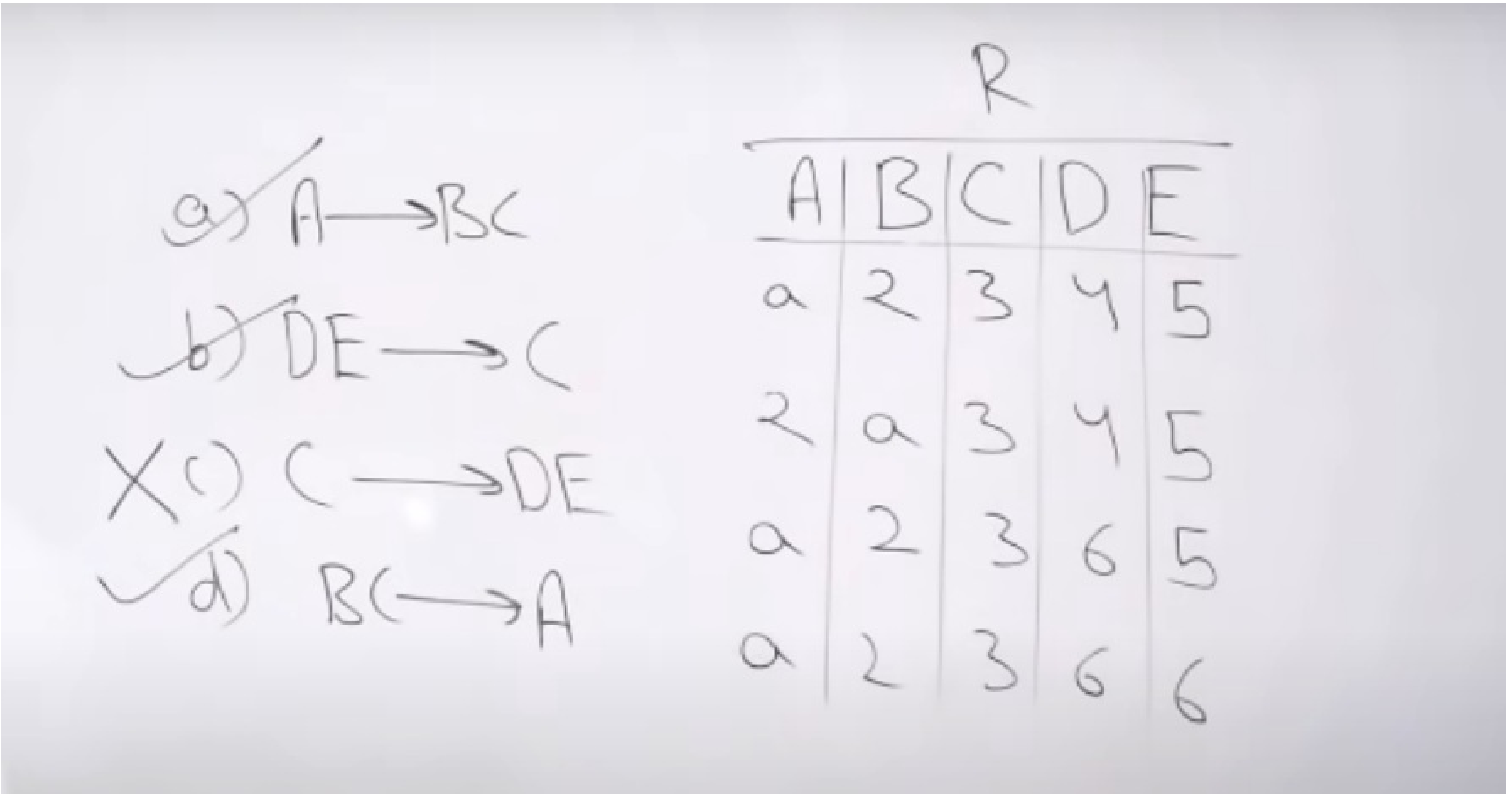
This is acceptable. No problem in this.

Ye bus kehne ka method hai ki α ki multiple occurrences pe kabhi alag values of β nahi milengi.



Data depend karta hai dependency pe, dependency data pe depend nahi karti. This means ek particular data ke liye shayad ek dependency valid ho, but data change hone pe shayad vo " valid na rahe

functional dependency {Questions}



a) $A \to BC$

b) $DE \to C$

c) $C \to DE$

d) $BC \to A$

R

A	B	C	D	E
a	2	3	4	5
2	a	3	4	5
a	2	3	6	5
a	2	3	6	6

\# Some imp points.

\# Dependency $\alpha \to \beta$

if all the values of α in the table are unique, ans is valid.

\# if all the values of β in the table are same, again the ans is valid.

Que 2)

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

a) $XY \rightarrow Z$ && $Z \rightarrow Y$

b) $YZ \rightarrow X$ && $Y \rightarrow Z$

c) $YZ \rightarrow X$ && $X \rightarrow Z$

d) $XZ \rightarrow Y$ && $Y \rightarrow Z$

Que 3

A	B	C
1	2	4
3	5	4
3	7	2
1	4	2

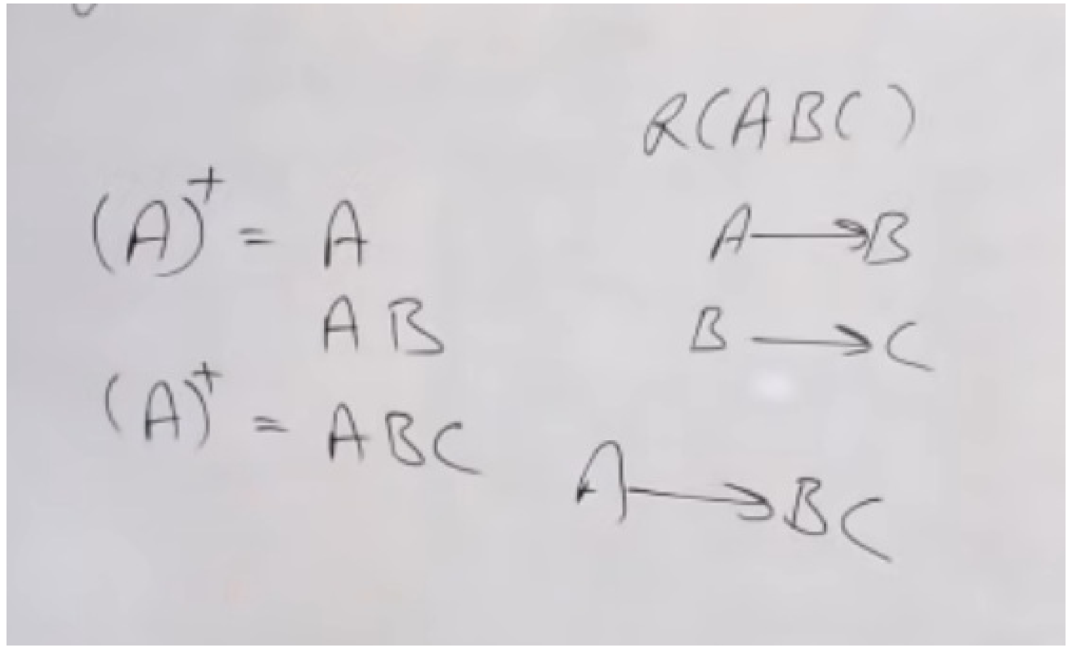
a) $A \rightarrow B$ && $BC \rightarrow A$

b) $C \rightarrow B$ && $CA \rightarrow B$

c) $B \rightarrow C$ && $AB \rightarrow C$

d) $A \rightarrow C$ && $BC \rightarrow A$

Armstrong's Axioms and Inference Rules in Functional Dependency



When we are finding the closure set of any attribute (say A is an attr then closure set is denoted as $(A)^+$) this means that we are trying to find all the attributes that can be searched directly or indirectly if we know the value of A.

Attribute closure/ closure on attribute set / closure of attribute set - Attribute closure of an attribute set 'A' can be defined as a set of attributes which can be functionally determined from it Denoted by F^+

$R(ABCDEF)$

$A \rightarrow B$

$C \rightarrow DE$

$AC \rightarrow F$

$D \rightarrow AF$

$E \rightarrow CF$

lets try to find $(D)^+$

$(D)^+ = DAFB$

lets find $(DE)^+$

$(DE)^+ = DEAFCB = (ABCDEF)$

Armstrong's axiom / rule:-

→ axiom is a statement that is taken to be true and serve as a premise or starting point for further arguments.

→ Armstrong axioms holds on very relational database can be used to generate closure set

Primary rules (RAT)

Reflexivity: if $Y \subseteq X$ then, $X \rightarrow Y$

$X = ABC$
$Y = AB$
$X \rightarrow Y$

⇓
same as trivial dependency.

Augmentation - if $X \rightarrow Y$ then, $XZ \rightarrow YZ$

Transitivity :- if $X \rightarrow Y$ && $Y \rightarrow Z$ then, $X \rightarrow Z$

Secondary rules

Union :- if $X \rightarrow Y$ && $X \rightarrow Z$
$X \rightarrow YZ$

Decomposition :- if $X \rightarrow YZ$
then, $X \rightarrow Y$ && $X \rightarrow Z$

Apply to the RHS of FD.

Pseudo transitivity :- if $X \rightarrow Y$ && $WY \rightarrow Z$
then, $WX \rightarrow Z$

Composition :- if $X \rightarrow Y$ && $Z \rightarrow W$
$XZ \rightarrow YW$

SUBSCRIBE [KG]

$AB \rightarrow C$

$A \rightarrow C$
$B \rightarrow C$ } Wrong ✗

$A \rightarrow C$

$B \rightarrow C$

$AB \rightarrow C$ } It is correct but this is making our FD weak as earlier

either A **or** B was sufficient to find C. Now, we are saying A **and** B both are needed

So, Union & decomposition are always applied to the RHS of FD.

Pseudo transitivity

If $X \rightarrow Y$ && $WY \rightarrow Z$

then, $WX \rightarrow Z$

① $X \rightarrow Y \xrightarrow{\text{augmentation}} WX \rightarrow WY$ ③

② $WY \rightarrow Z$

③ and ② are transitive.

$\Rightarrow WX \rightarrow Z$

Ques 1) $R(ABC)$

$A \rightarrow B$

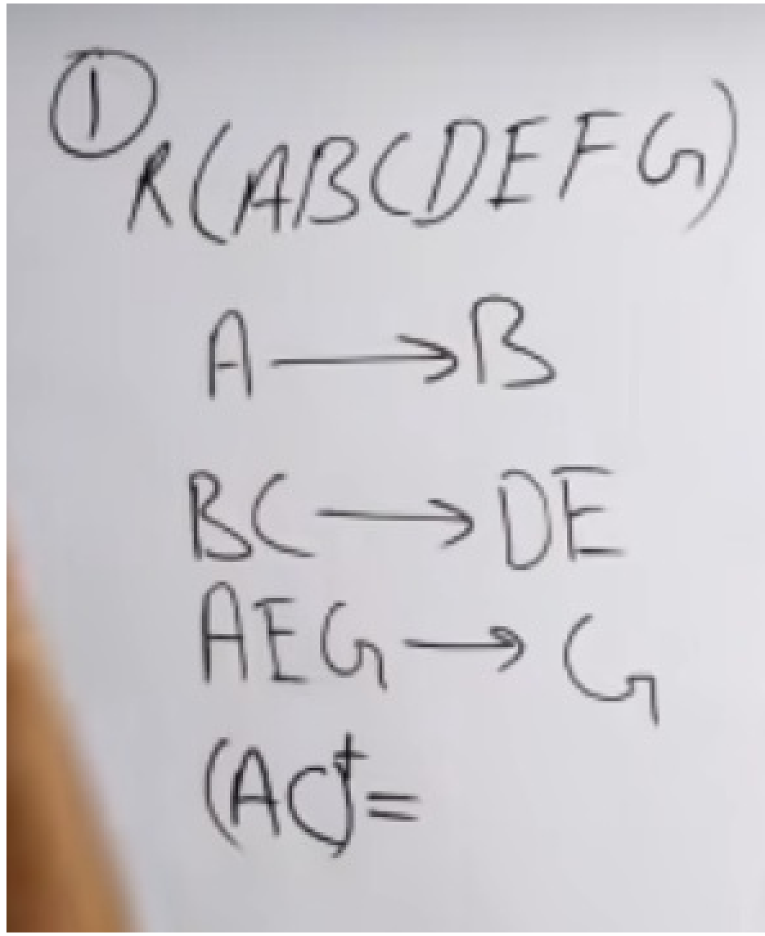
$B \rightarrow C$

$A^+ = ABC$

$B^+ = BC$

$C^+ = C$

Ques 2)



① $R(ABCDEFG)$

$A \rightarrow B$

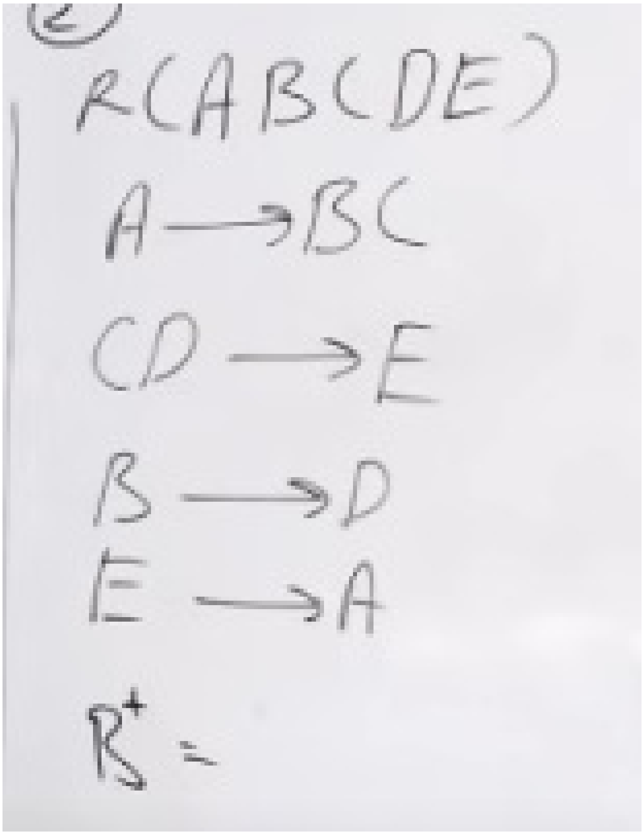
$BC \rightarrow DE$

$AEG \rightarrow G$

$(AC)^+ =$

$(AC)^+ = ACBDE = (ABCDE)$

Ques 3



② $R(ABCDE)$

$A \rightarrow BC$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$

$B^+ =$

$B^+ = BD$

Ques 4)

$R(ABCDEF)$

$AB \rightarrow C$

$BC \rightarrow AD$

$D \rightarrow E$

$CF \rightarrow B$

$(AB)^+ = ?$

$(AB)^+ = (ABCDE)$

Ques 5)

$R(ABCDEFGH)$

$A \rightarrow BC$

$CD \rightarrow E$ ✓

$E \rightarrow C$ ✓

$D \rightarrow AEH$ ✓

$ABH \rightarrow BD$ ✓

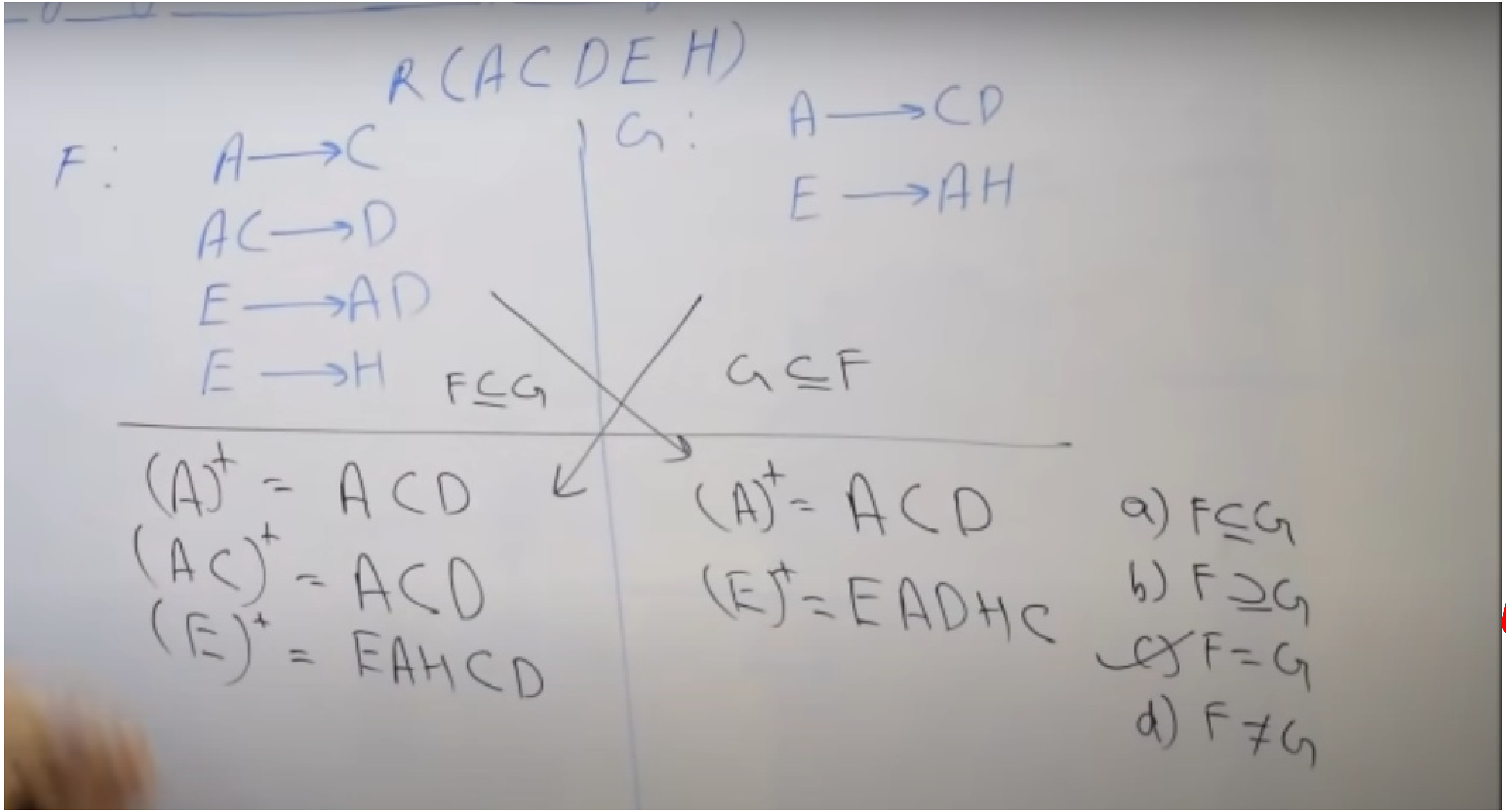
$DH \rightarrow BC$

$BCD \rightarrow H$?

$(BCD)^+ = BCDEHA$

True → valid FD

Equivalence of functional dependencies.



Compute closure of LHS of f using G. Then see if all FD of f hold true.

Now, compute closure of LHS of G using f & see if all FDs of G hold true.

Q

$R(PQRS)$

$X:$

$P \rightarrow Q$

$Q \rightarrow R$

$R \rightarrow S$

$Y:$

$P \rightarrow QR$

$R \rightarrow S$

$X \not\subseteq Y$

$(P)^+ = PQRS$

$(Q)^+ = Q$

$(R)^+ = RS$

$(P)^+ = PQRS$

$(R)^+ = RS$

a) $X \leq Y$

b) $Y \leq X$

c) $X = Y$

d) $X \neq Y$

Q

$R(ABC)$

$F: A \rightarrow B$

$B \rightarrow C$

$C \rightarrow A$

$G: A \rightarrow BC$

$B \rightarrow A$

$C \rightarrow A$

$(A)^+ = ABC$

$(B)^+ = ABC$

$(C)^+ = ABC$

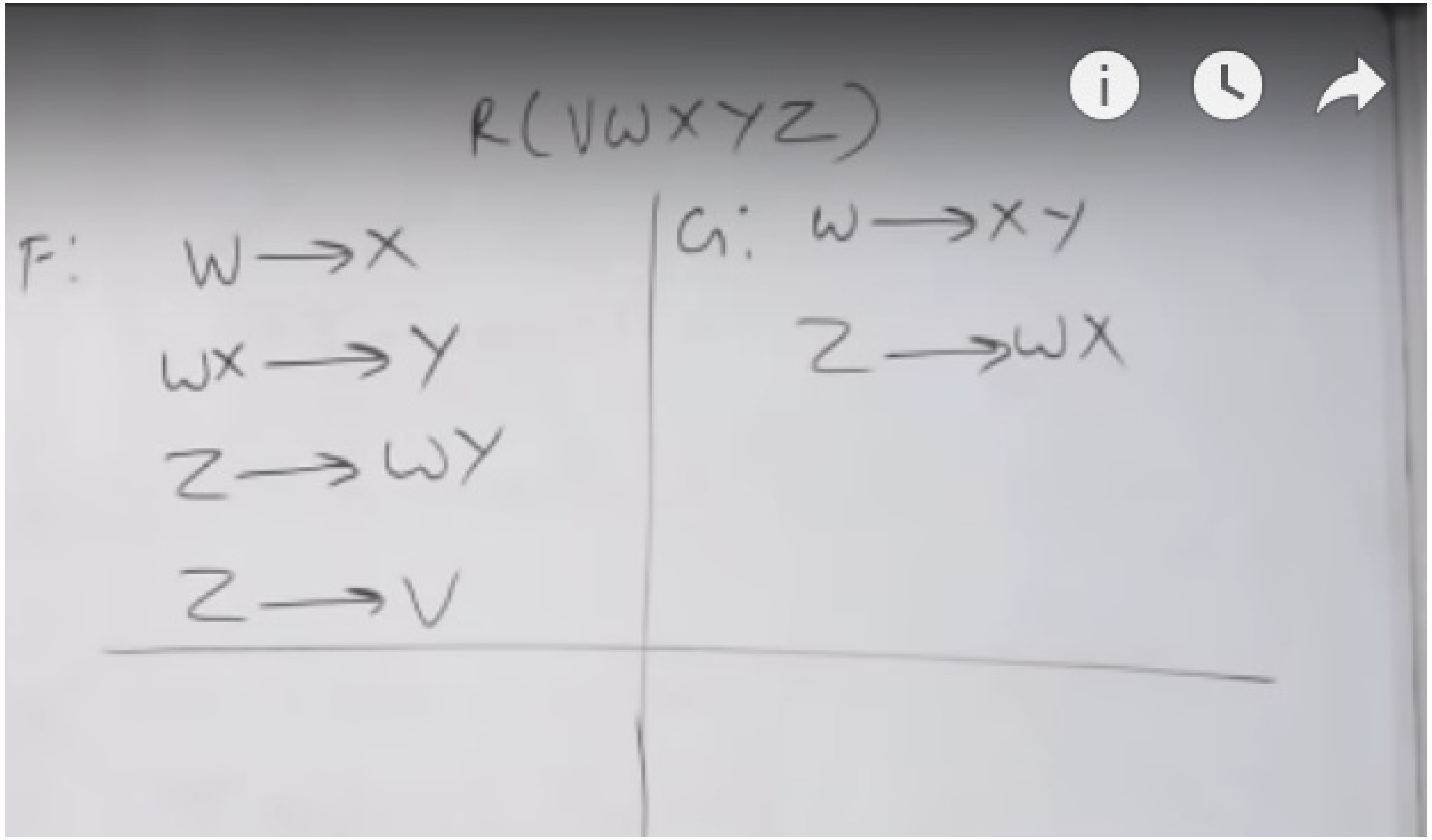
$(A)^+ = ABC$

$(B)^+ = BCA$

$(C)^+ = CAB$

$$\boxed{F = G}$$

0



$(W)^+ = WXY$

$(WX)^+ = WXY$

$(Z)^+ = ZWXY$

$(W)^+ = WXY$

$(Z)^+ = WZYX$

$\boxed{G \subseteq F}$

Minimal Cover or Canonical Cover or Irreducible set of Functional Dependencies

R(WXYZ)

X → W

WZ → XY

Y → WXZ

Step 1

decomposition

RHS redundancy removal

X → W ✓

~~WZ → X~~

WZ → Y ✓

~~Y → W~~

Y → X ✓

Y → Z ✓

check entire Rules

$(X)^+ = XW$

$(X)^+ = X$ (ignoring 1st Rule)

$(WZ)^+ = WZXY$

$(WZ)^+ = WZYX$

$(WZ)^+ = WZYX$

$(WZ)^+ = WZ$

$(Y)^+ = YXZW$

$(Y)^+ = YXZW$

$(Y)^+ = YXZW$

$(Y)^+ = YZ$

$(Y)^+ = YXZW$

$(Y)^+ = YXW$

X → W

WZ → Y

Y → X

Y → Z

Removing LHS Redundancy

$(WZ)^+ = WZYX$

$(W)^+ = W$

$(Z)^+ = Z$

X → W

WZ → Y

Y → XZ

Ans

One FD can have multiple canonical forms. Canonical form also depends on the order in which we process functional dependency.

$R(ABCD)$

$A \to B$

$C \to B$

$D \to ABC$

$AC \to D$

$A \to B$ ✓

$C \to B$ ✓

$D \to A$ ✓

$\boxed{\cancel{D \to B}}$

$D \to C$ ✓

$AC \to D$ ✓

$(A)^+ = AB$

$(C)^+ = BC$

$(D)^+ = DABC$

$(D)^+ = DBC$

$(D)^+ = DAC\,B$

$(D)^+ = DCAB$

$(D)^+ = DAB$

$(AC)^+ = ACDB$

$(AC)^+ = ACB$

$A \to B$

$C \to B$

$D \to A$

$D \to C$

$AC \to D$

$\Rightarrow$

$A \to B$

$C \to B$

$D \to AC$

$AC \to D$

$(AC)^+ = ACBD$

$(A)^+ = AB$

$(C)^+ = CB$

$R(VWXYZ)$

$V \to W$

$VW \to X$

$Y \to VXZ$

$V \to W$ ✓

$VW \to X$ ✓

$Y \to V$ ✓

$\boxed{\cancel{Y \to X}}$

$Y \to Z$ ✓

$(V)^+ = VWX$

$(VW)^+ = VWX$

$(VW)^+ = VW$

$(Y)^+ = YVXZW$

$(Y)^+ = YXZ$

$(Y)^+ = YVZWX$

$(Y)^+ = YVWX$

$V \to W$

$VW \to X$

$Y \to V$

$Y \to Z$

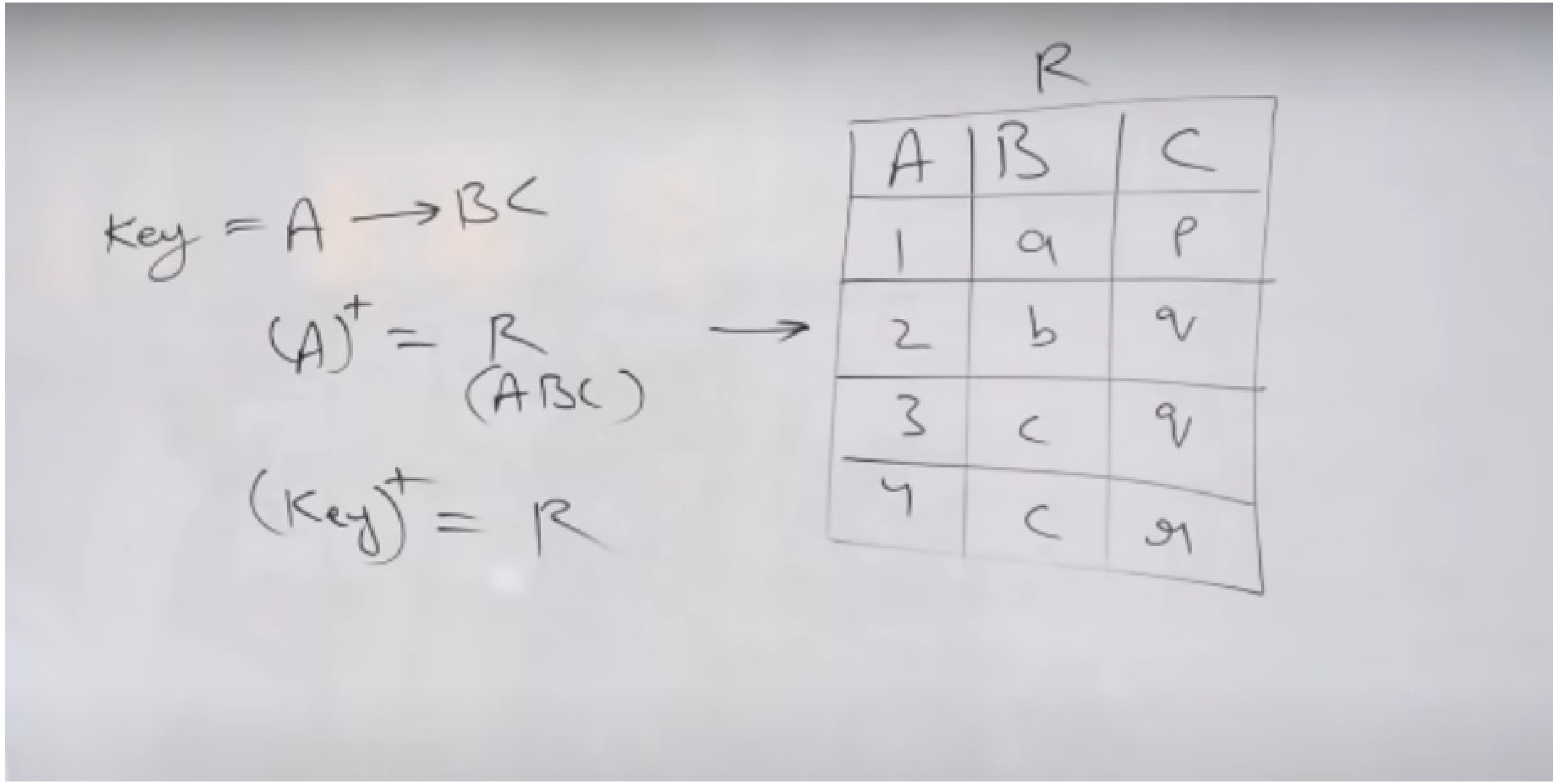
$\Rightarrow$ $\boxed{\begin{array}{l} V \to WX \\ Y \to VZ \end{array}}$

$(VW)^+ = VWX$

$(V)^+ = VWX$

$(W)^+ = W$

Keys in DBMS



Agar attr ka closure poora relation de hai, attr is a key.

A and BC, both can be keys here.

Key is a set of attributes that can uniquely identify a row or a tuple in a Relational DB.

Key Table ki hoti hai, complete DB ki nahi.

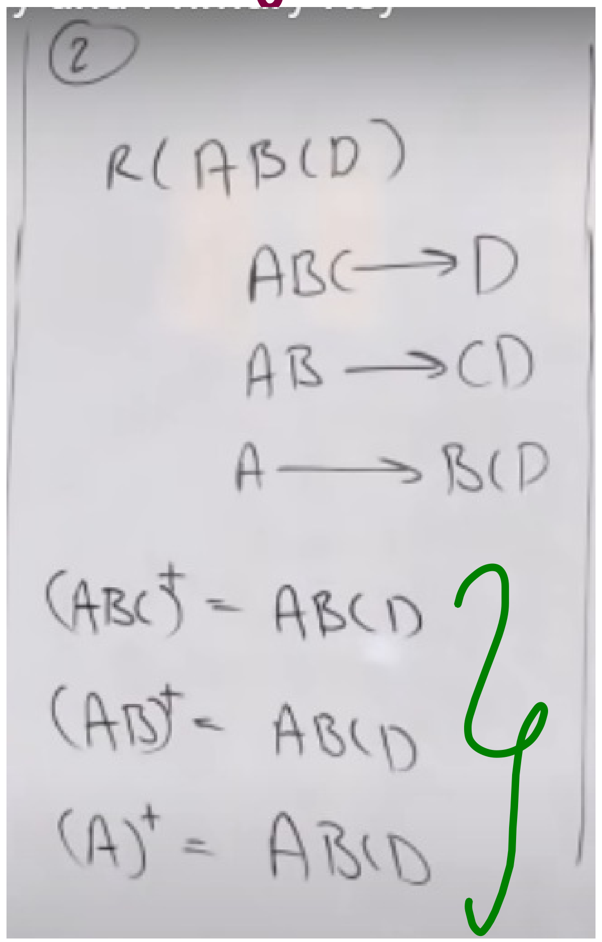
Super Key, Candidate Key & Primary Key

Super Key = key ko he super key kehte hai. DBMS me key keval nahi hoto. Key humne bus concept ke liye padna hai.

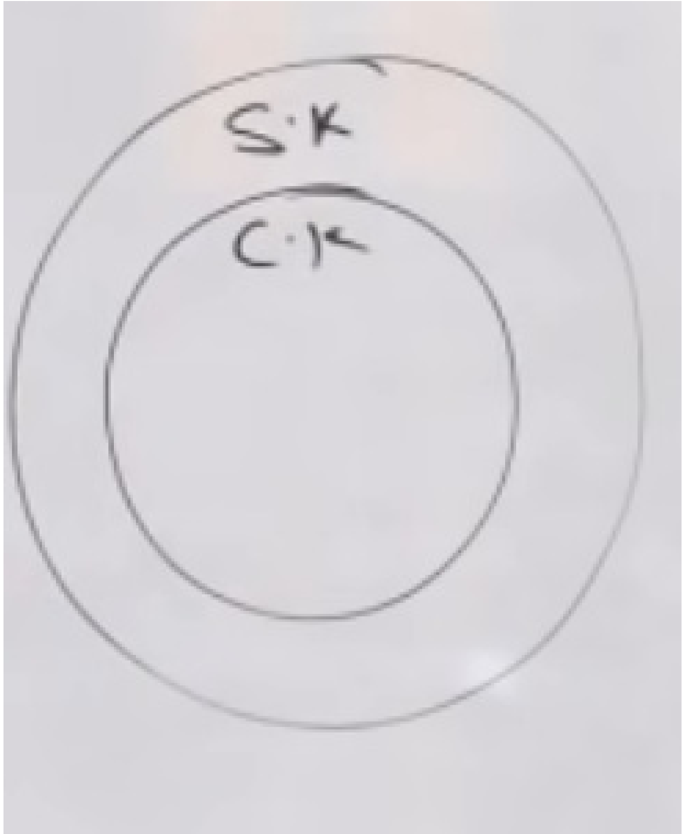
Candidate Key = Efficient/Minimal Super Key.

Us Super key ko candidate key kehte hai jiska koi proper subset super key na ho.

Candidate Key →



All are Super Keys



$R(ABCD)$

$B \rightarrow ACD$

$ACD \rightarrow B$

$(B)^+ = (BACD)$ → Super Key & Candidate Key

$(ACD)^+ = (ACDB)$ → Super key & candidate key

Now, DBA chooses one Candidate Key out of all to uniquely identify the tuples in a relationship. That CK is called Primary key.

R(ABCD)

$AB \rightarrow C$

$C \rightarrow BD$

$D \rightarrow A$

$(AB)^+ = ABCD$

$(C)^+ = CBDA$

$(D)^+ = DA$

Super Keys & candidate Keys

① R(ABCD)

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow A$

A B C D

essential

$AD = ADBC$

$BD = BDCA$

$CD = CDAB$

3 CKs

R(ABCD)

AB → CD
D → A

AB CD

B → essential

AB → ABCD → Candidate Key

BC → BC

BD → BDAC → candidate key

(3) R(ABCDEF)
AB → C
C → D
B → AE

⇒ BF → BF AECD

CK

(9) $R(ABCD)$

$AB \rightarrow CD$

$C \rightarrow A$

$D \rightarrow B$

$(A)^+ = A$

$(B)^+ = (B)$

$(C)^+ = CA$

$(D)^+ = DB$

$(AB)^+ = ABCD \Rightarrow CK$

$(AC)^+ = AC \times$

$(AD)^+ = ADBC \Rightarrow CK$

$(BC)^+ = BCAD \Rightarrow CK$

$(BD)^+ = BD \times$

$(CD)^+ = CDAB \Rightarrow CK$

An

AB, AD, BC, CD

(5) R(ABCDE)

AB → CD

D → A

BC → DE)

B → essential

AB → ABCDE ⟹ CK

BC → BCDEA ⟹ CK

BD → BDACE ⟹ CK

BE → BE

Ans

⑥ $R(WXYZ)$

$Z \to W$

$Y \to XZ$

$XW \to Y$

$(Y)^+ = YXZW \Rightarrow \underline{CK}$

$(Z)^+ = ZW$

(Y, XW, XZ)

$WX = WXYZ \Rightarrow CK$

$WZ = WZ$

$XZ = XZWY \Rightarrow CK$

⑦ R(ABCDEF)

AB → C

DC → AE

E → F

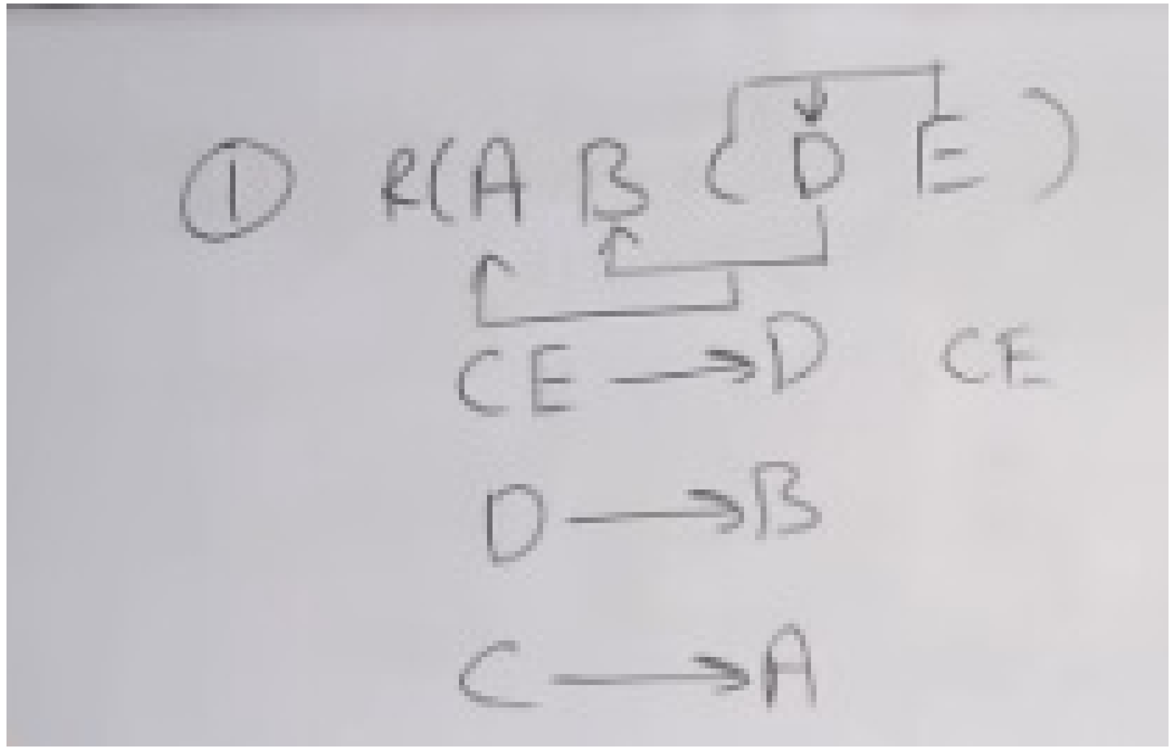
BD

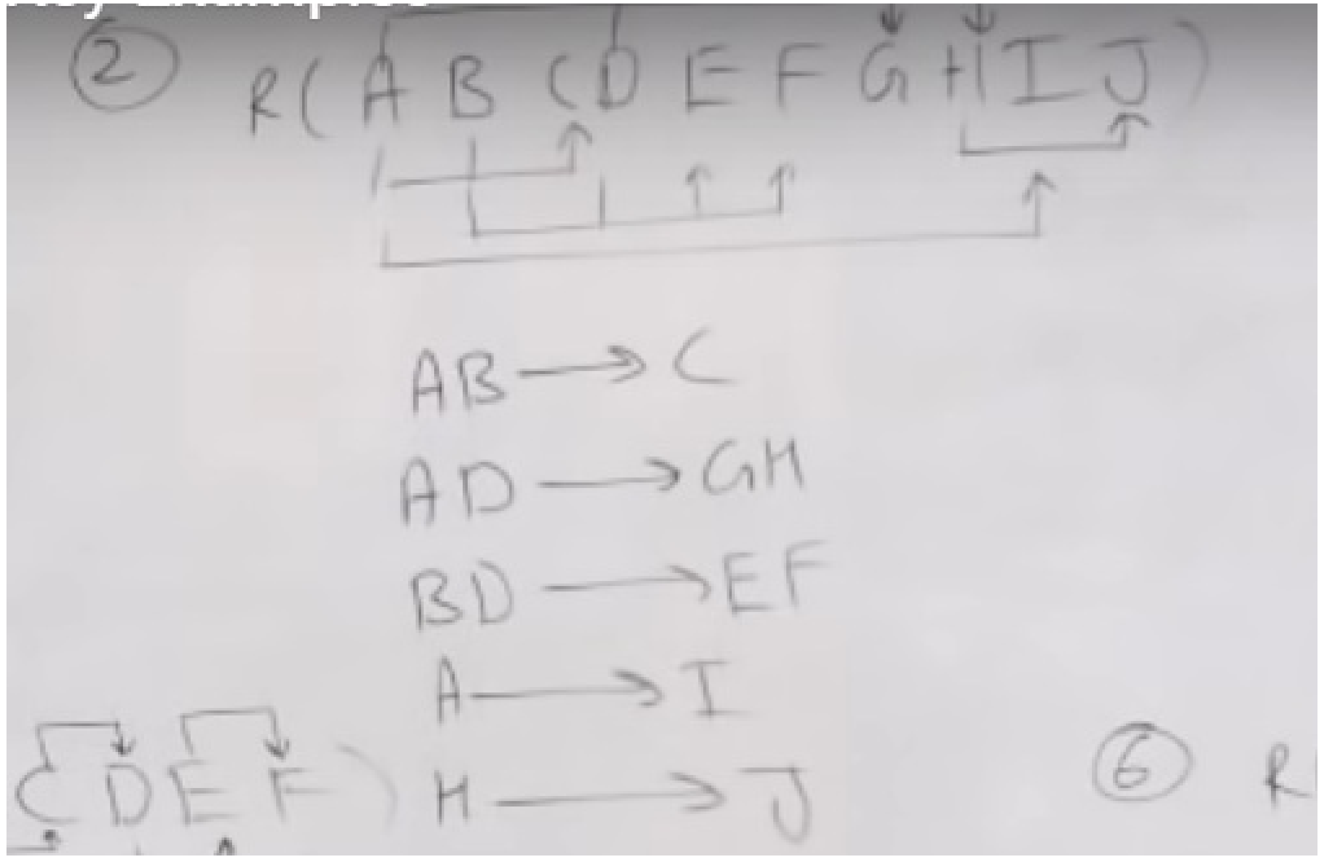
ABD = ABDCEF → CK

CBD = CBDAEF → CK

EBD = EBDF

FBD = FBD





$(CE)^+ = CEDBA \Rightarrow$ CK

Essential : ABD → CK

$(ABD)^+ = ABDCGHEFIJ$

③ R(A B C D E)

$A \rightarrow B$

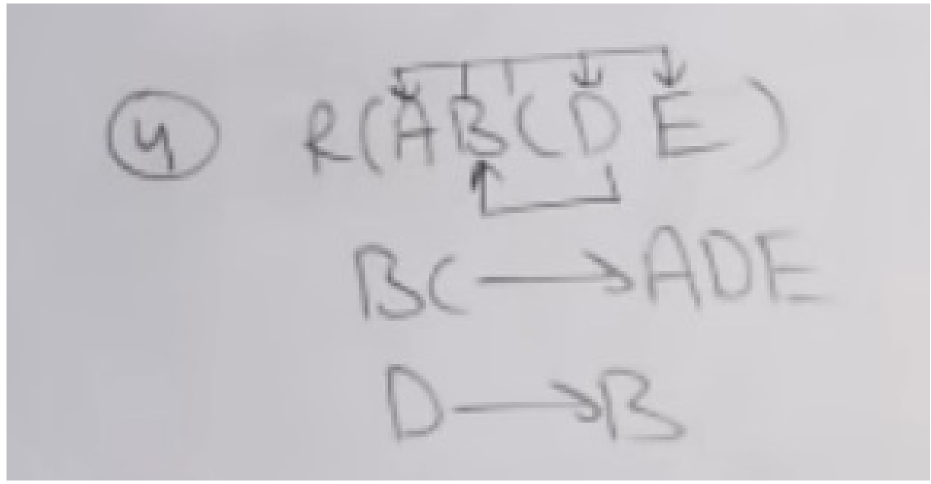
$BC \rightarrow E$

$DE \rightarrow A$

Essential: CD

$(CD)^+ = CD$

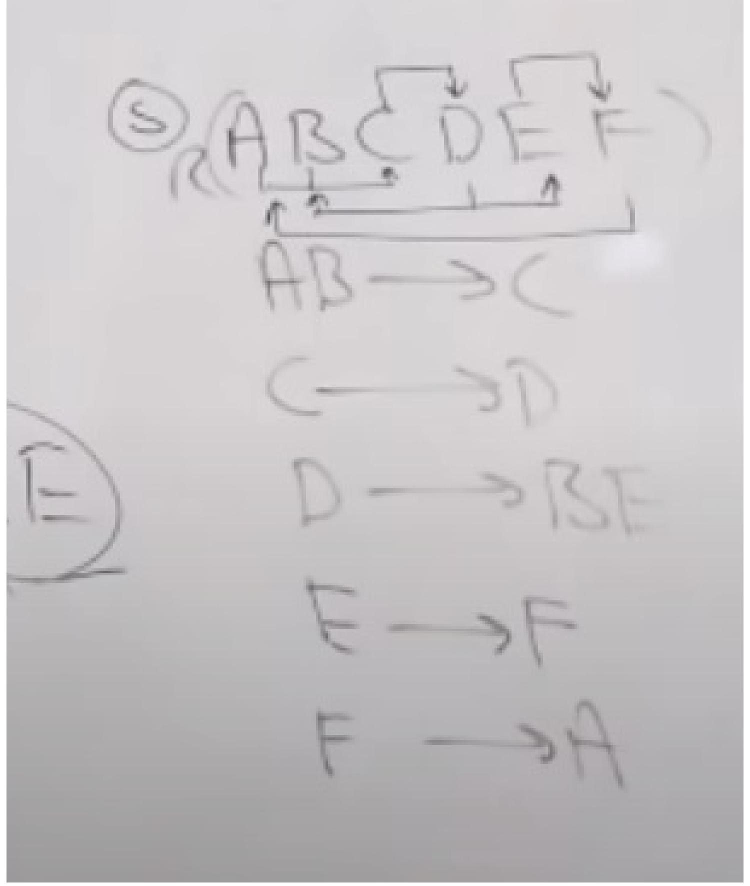
$$\left.\begin{aligned} (ACD)^+ &= ACDBE \\ (BCD)^+ &= BCDEA \\ (CDE)^+ &= CDEAB \end{aligned}\right\} CK$$



Essential: C

$(BC)^+ = ABCDE \Rightarrow CK$

$(DC)^+ = DCBAE \Rightarrow CK$



$(AB)^+ = ABCDEF \rightarrow CK$

$(AE)^+ = AEF$

$(AF) \propto$

$(BE)^+ = BEFACD \rightarrow CK$

$(BF)^+ = BFACDE \rightarrow CK$

$(EF)^+ = EFA \propto$

$(C)^+ = CDBEFA \rightarrow CK$

$(D)^+ = DBEFAC \rightarrow CK$

$(E)^+ = EFA$

$(F)^+ = FA$

⑥ $R(ABCDEFGH)$

$CH \to G$

$A \to BC$

$B \to CFH$

$E \to A$

$F \to EG$

Essential: D

$(AD)^+ = ABCDEFGH \to CK$

$(BD)^+ = ABCDEFGH \to CK$

$(CD)^+ = CD$

$(DE)^+ = DEABCFHG \to CK$

$(DF)^+ = DFEGABCH \to CK$

$(DG)^+ = DG$

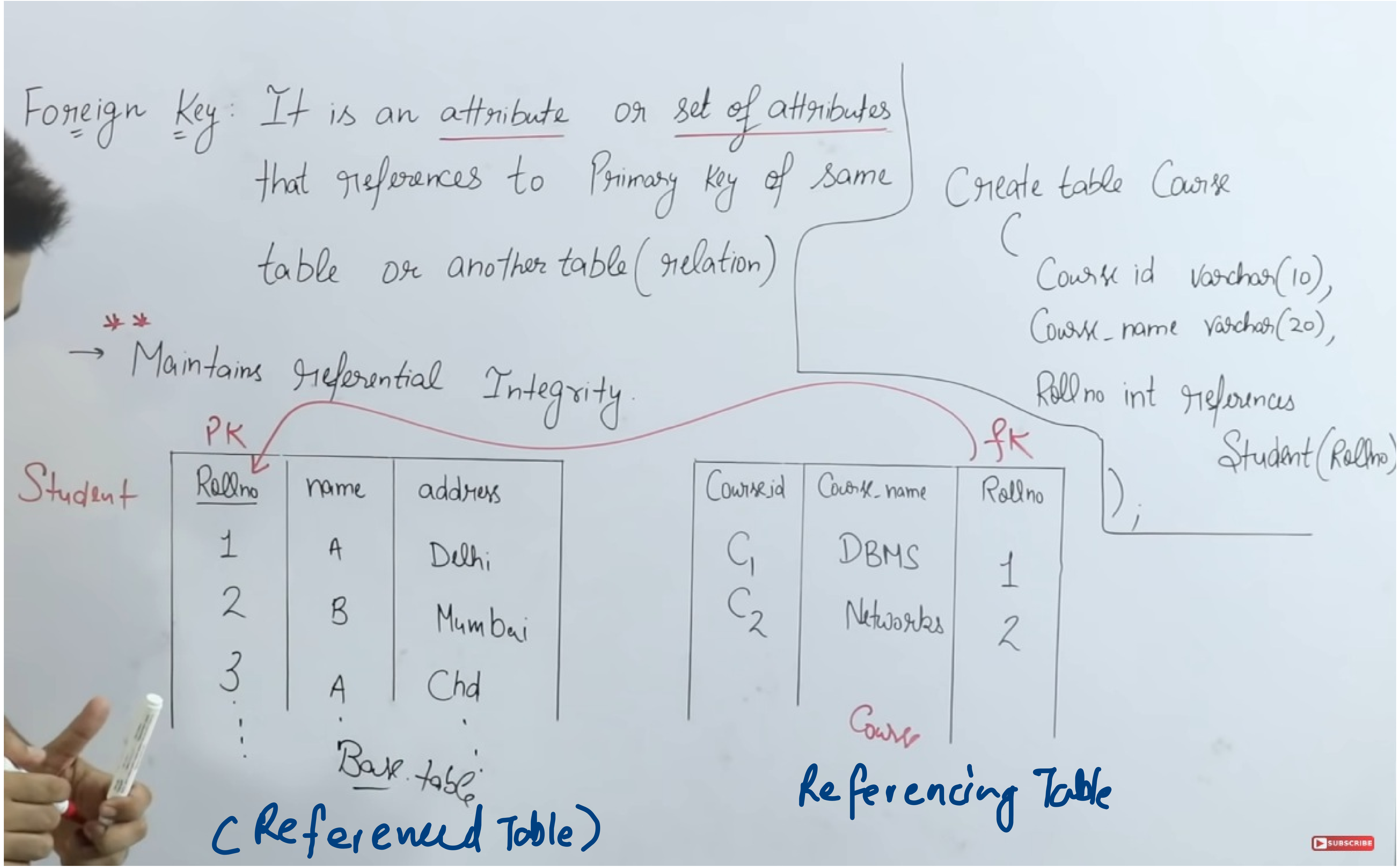
$(DH)^+ = DH$

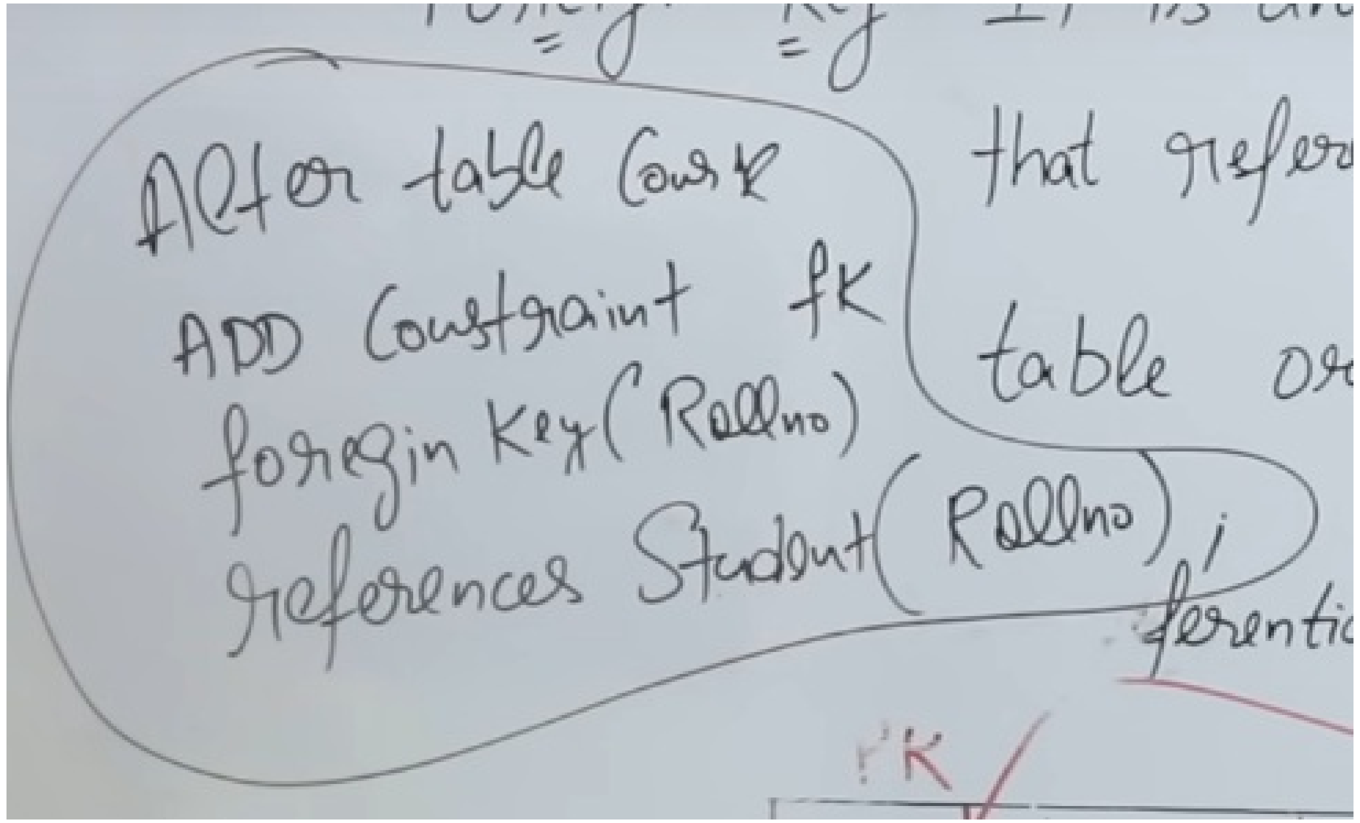
$CDG = CDG$

$CDH = CDHG$

$DGH = DGH$

Foreign Key


Foreign Key: It is an attribute or set of attributes
that references to Primary key of same
table or another table (relation)
** → Maintains referential Integrity.
PK
Student
Rollno name address
1 A Delhi
2 B Mumbai
3 A Chd
Base table
(Referenced Table)
FK
Courseid Course_name Rollno
C1 DBMS 1
C2 Networks 2
Course
Referencing Table
Create table Course
(
Course id varchar(10),
Course_name varchar(20),
Rollno int references
Student(Rollno)
);



One table can have more than one foreign keys but only 1 PK.

FK provides 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

[redacted]

3) Updation → May cause violation

All in bote table

Student
(Base table or Referenced table)

Rollno (PK)	name	address
~~1~~	~~A~~	~~Delhi~~
2	B	Mumbai
3	A	Chd
~~4~~	~~D~~	~~Chd~~

Course_id	Course_name	Rollno (FK)
C_1	DBMS	1
C_2	Networks	2

Course (Referencing table)

Referencing table

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

Data Redundancy, Updation, Deletion & Insertion Anomalies in DBMS

Idea → In the table studentinfo we have tried to store entire data about student.

Result → Entire branch data of a brach must be repeated for every student of the branch

Redundancy → when same data is stored multiple times unnecessarily in a database

Disadvantages → (i) Insertion, deletion and modification anomalies
(ii) Inconsistency (data)
(iii) Increase in database size and increase in time (slow)

Insertion anomalies: → when certain data (attribute) cannot be inserted into DataBase, without the persence of other data

Deletion anomalies: - if we delete some data (unwanted), it cause deletion of some other data (wanted)

Updation/Modification anomalies: when we want to update a single piece of data, but it must be done at all its copies.

Student_info

S_id	name	age	Br_code	Br_name	HOD_name
1	A	18	101	CS	XYZ
2	B	19	101	CS	XYZ
3	C	18	101	CS	XYZ
4	D	21	102	EC	PQR
5	E	20	102	EC	PQR
6	F	19	103	ME	KLM

SUBSCRIBE [KG]

Insertion Anomaly.

Agar Civil Branch bhi college me hai, I can't find the data of Civil Branch from my table unless a student enrolls in it. This is wrong as the existence of Branch should be independent of Student.

Student_info

S_id	name	age	Br_code	Br_name	HOD_name
1	A	18	101	CS	XYZ
2	B	19	101	CS	XYZ
3	C	18	101	CS	XYZ
4	D	21	102	E.C	PQR
5	E	20	102	E.C	PQR
6	F	19	103	M.E	KLM

Deletion Anomaly

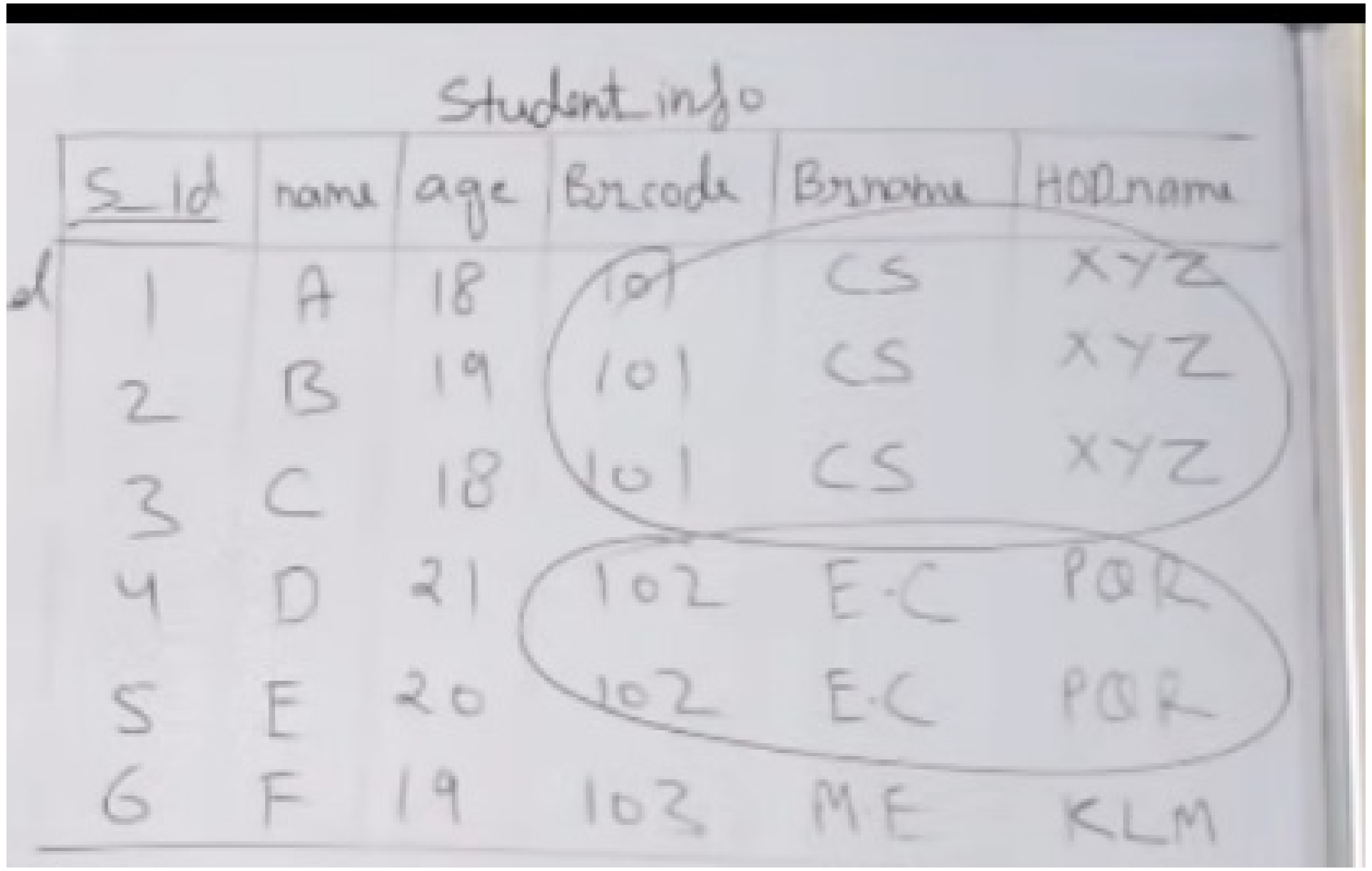
If from the table, we delete the Student of Mechanical Branch, we will not have any data related to Mech Branch in the table however the Branch exists.

Student_info

S_id	name	age	Br_code	Br_name	HOD_name
1	A	18	101	CS	XYZ
2	B	19	101	CS	XYZ
3	C	18	101	CS	XYZ
4	D	21	102	E·C	PQR
5	E	20	102	E·C	PQR
6	F	19	103	M·E	KLM

Updation Anomaly

If HoD of CSE changes, the single change would have to be made in all the rows where the Branch is CSE.



Student_info

S_id	name	age	Br_code	Br_name	HOD_name
1	A	18	101	CS	XYZ
2	B	19	101	CS	XYZ
3	C	18	101	CS	XYZ
4	D	21	102	E.C	PQR
5	E	20	102	E.C	PQR
6	F	19	103	ME	KLM

Normalisation

Student info

S_id	name	age	Br code	Br name	HoD_name
1	A	18	101	CS	XYZ
2	B	19	101	CS	XYZ
3	C	18	101	CS	XYZ
4	D	21	102	EC	POR
5	E	20	102	EC	POR
6	F	19	103	ME	KLM

student_info

S_id	name	age	Br code
1	A	18	101
2	B	19	101
3	C	18	101
4	D	21	102
5	E	20	102
6	F	19	103

Branch_info

Br code	Br name	HoD name
101	C.S	XYZ
102	E.C	POR
103	M.E	KLM

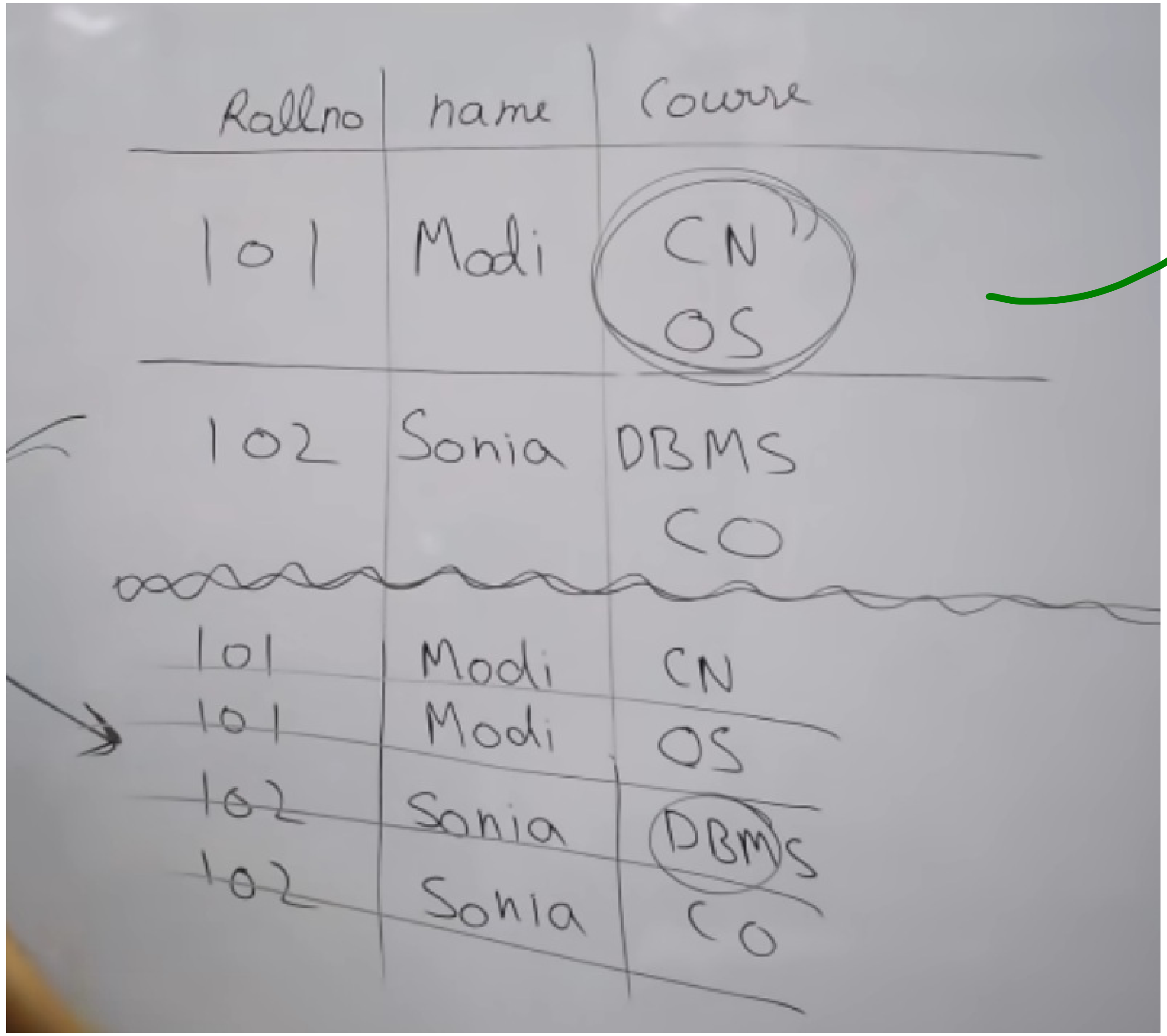
→ As one paragraph contains a single idea similary one table must contain direct & main data about an Entity

→ Normalization (Decomposition of tables) of table is done of the basic of functional dependencies

1NF

(1) Every col should have unique name.

(3) Order of Rs & Cls is insignifica

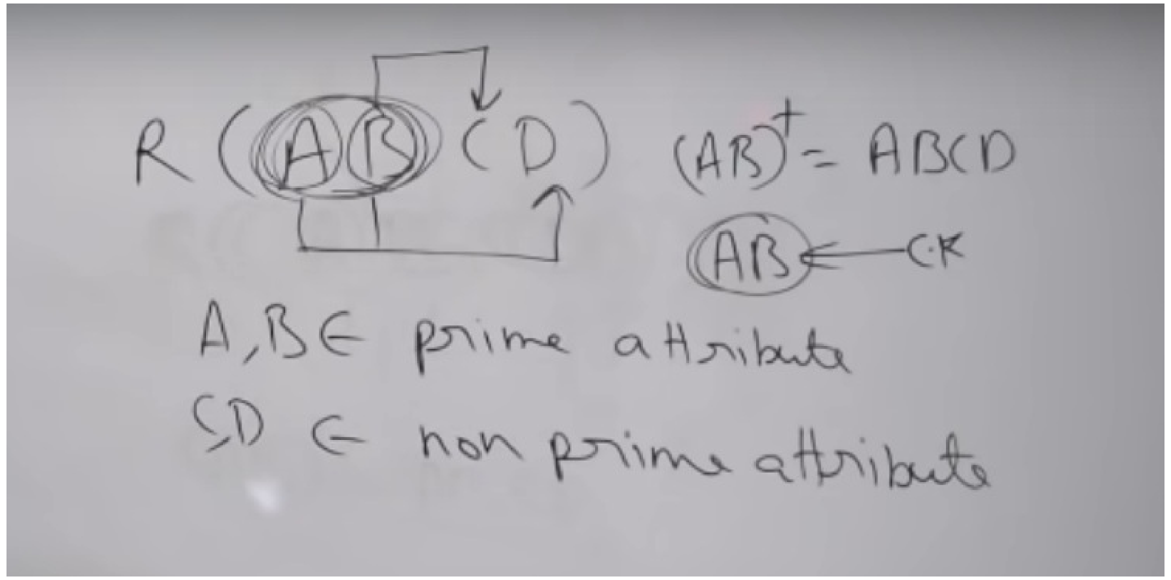


(2) Multivalued attributes should not be present in the table.

Usually, if we talk about any relational schema, it is already implied that it is in 1NF.

(4) In every col, values must belong to the same domain.

2NF



If an attr is a part of any one of the candidate keys, it is called a prime attribute.

When a non-prime attribute depends only on some (1 or more) prime attributes & not the entire candidate key, then it is called a partial dependency.

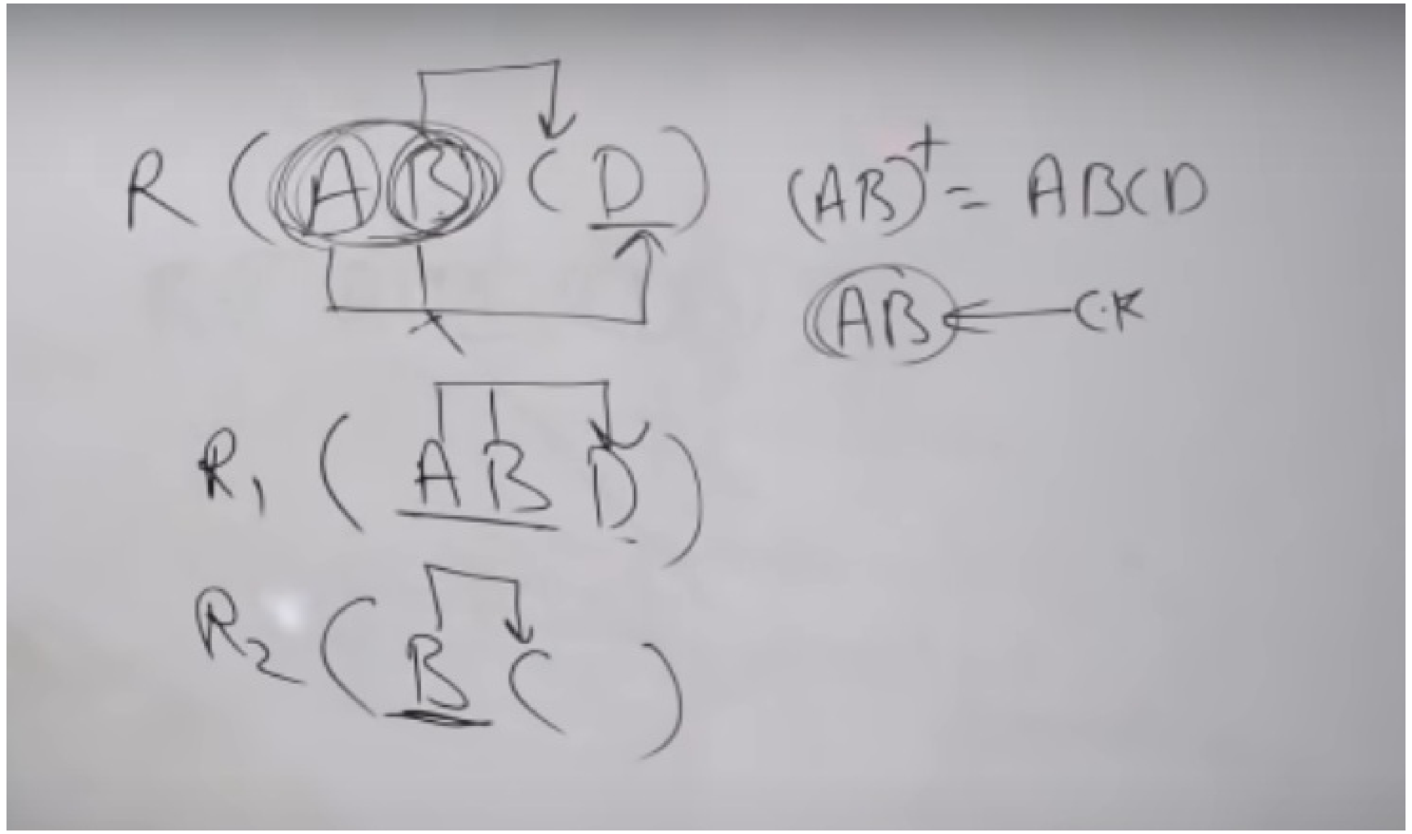
$B \rightarrow C$

↳ C instead of AB, depends only on B.

If a table is in 2NF, it should be in 1NF & independent of partial dependencies.

The problem with partial dependency is that if CK has multiple attributes, they can all not be null together as CK will become null but, some of them can be null at the same time. So, if in prev example

$B \rightarrow C$ } C.Key is AB. So, B can be null & when B = null, we won't be able to compute C.



Converting to 2NF.

$R(\underline{AB}C)$

$R_1(\underline{AB})$ $R_2(\underline{B}C)$

R

A	B	C
a	1	x
b	2	y
a	3	z
c	3	z
d	3	z
e	3	z

R_1

A	B
a	1
b	2
a	3
c	3
d	3
e	3

R_2

B	C
1	x
2	y
3	z

SUBSCRIBE KG

Identification of 2NF & decomposition to 2NF

① $R(ABCDE)$

$AB \rightarrow C$

$D \rightarrow E$

Essential : ABD

$(ABD)^{+} = ABCDE \Rightarrow$ Candidate Key.

$$\left.\begin{array}{l} AB \rightarrow C \\ D \rightarrow E \end{array}\right\} \text{Partial dependency.}$$

$R_1(ABC) \quad R_2(DE) \quad R_3(ABD)$

② $R(ABCDE)$

$A \rightarrow B$
$B \rightarrow E$
$C \rightarrow D$

Essential = AC

$(AC)^+ = ABCDE \Rightarrow$ <u>CK</u>

PA = AC

NPA = B, D, E

$$\left.\begin{array}{l} A \rightarrow B \\ C \rightarrow D \end{array}\right\} \text{Partial dependencies}$$

$R_1(ABE)$

$R_2(CD)$

$R_3(AC)$

(3) R(ABCDEFGHIJ)

$AB \to C$
$AD \to GH$
$BD \to EF$
$A \to I$
$H \to J$

Essential = ABD

$(ABD)^+ = ABCDEFGHIJ$
$\Downarrow$
candidate key.

PA = ABD
NPA = CEFGHIJ

R_1 ✓ (ABC)
R_2 ✓ (ADGHJ)
R_3 ✓ (BDEF)
R_4 ✓ (AI)
R_5 (ABD) ✓

3NF

Transitive dependency:- A F.D from $\alpha \rightarrow \beta$ is called transitive if $\alpha, \beta \in$ non-prime

R(A B C)

$A \rightarrow B$

$B \rightarrow C$

R

A	B	C
a	1	x
b	1	x
c	1	x
d	2	y
e	2	y
f	3	z
g	3	z

NF:- A relation R is in 3NF if

a) it is in 2NF

b) no transitive dependency

every dependency from $\alpha \rightarrow \beta$

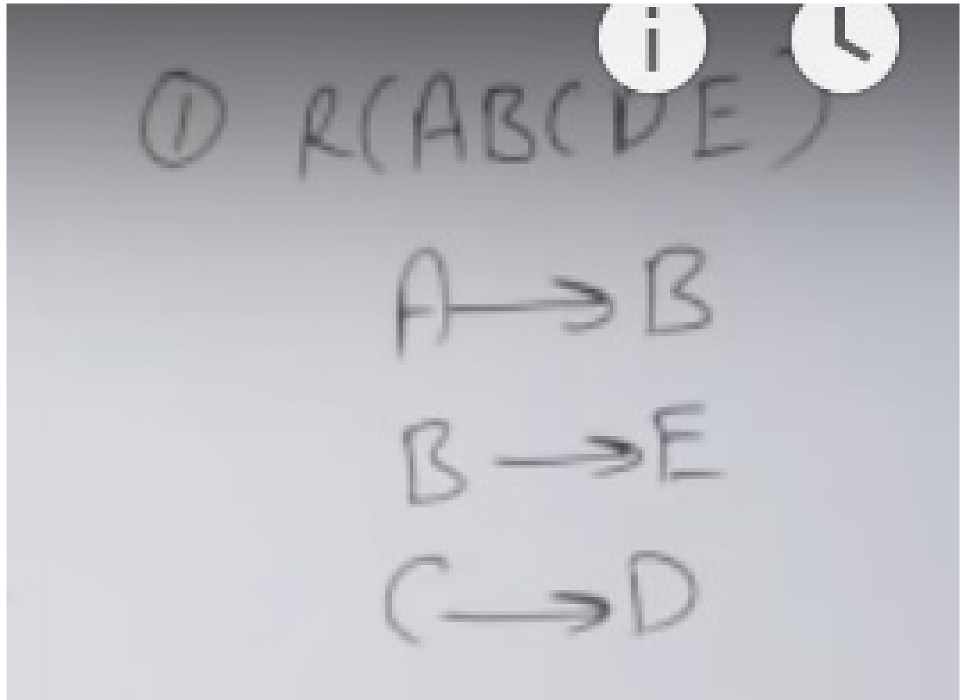
(i) either α is super key

(ii) or β is a prime attribute

A	B
a	1
b	1
c	1
d	2
e	2
f	3
g	3

B	C
1	x
2	y
3	z

SUBSCRIBE [KG]



$(AC)^{+} = ABCDE \Rightarrow$ candidate key

$\Downarrow$

PA

$R_1(A, B, E)$

$R_{11}(B, E)$

$R_{11}(A, B)$

$R_2(C, D)$

$R_3(A, C)$

R(A B C D E F G H I J)

$AB \rightarrow C$

$A \rightarrow DE$

$B \rightarrow F$

$F \rightarrow GH$

$D \rightarrow IJ$

AB

$(AB)^+ = ABCDEFGHIJ$

⇓

candidat key.

- $R_1(A, D, E, I, J)$
  - $R_{11}(A, D, E)$ ①
  - $R_{12}(D, I, J)$ ②
- $R_2(B, F, G, H)$
  - $R_{21}(B, F)$ ③
  - $R_{22}(F, G, H)$ ④
- $R_3(A, B)$ ⑤

4.10 Decomposition into 3NF T

(3) R(ABCDE)

$AB \to C$

$B \to D$

$D \to E$

$AB \to$ essential

$(AB)^+ = ABCDE$

$\Downarrow$

candidate key

- $\to R_1(B, D, E)$
  - $\to R_{11}(B, D)$
  - $\to R_{12}(D, E)$
- $\to R_2(ABC)$

④ $R(ABCDEFGHIJ)$

$AB \to C$

$AD \to GH$

$BD \to EF$

$A \to I$

$H \to J$

$(ABD)^+ = ABCDEFGHIJ$

$\Downarrow$ CK

$R_1(ABCI) \to R_{11}(ABC)$ 1

$ \to R_{12}(AI)$ 2

$R_2(ADGHJ)$

$\to R_{21}(ADGH)$ 3

$\to R_{22}(HJ)$ 4

$R_3(BDEF)$ 5

$R_4(ABD)$ 6

BCNF

$\alpha \rightarrow \beta$ 2NF (Partial dependency)

P NP

$\alpha \rightarrow \beta$ 3NF (Transitive dependency)

NP NP

$\alpha \rightarrow \beta$ We have not talked about these cases.

P/NP P BCNF deals with these cases.

$R(A\ B\ C)$

$AB \rightarrow C$ $\quad (AB)^+ = ABC \Rightarrow CK$

$C \rightarrow B$ $\quad (AC)^+ = ABC \Rightarrow CK$

This dependency should not be allowed in BCNF.

So, in BCNF, α for all FDs should be a super key.

$\Downarrow$

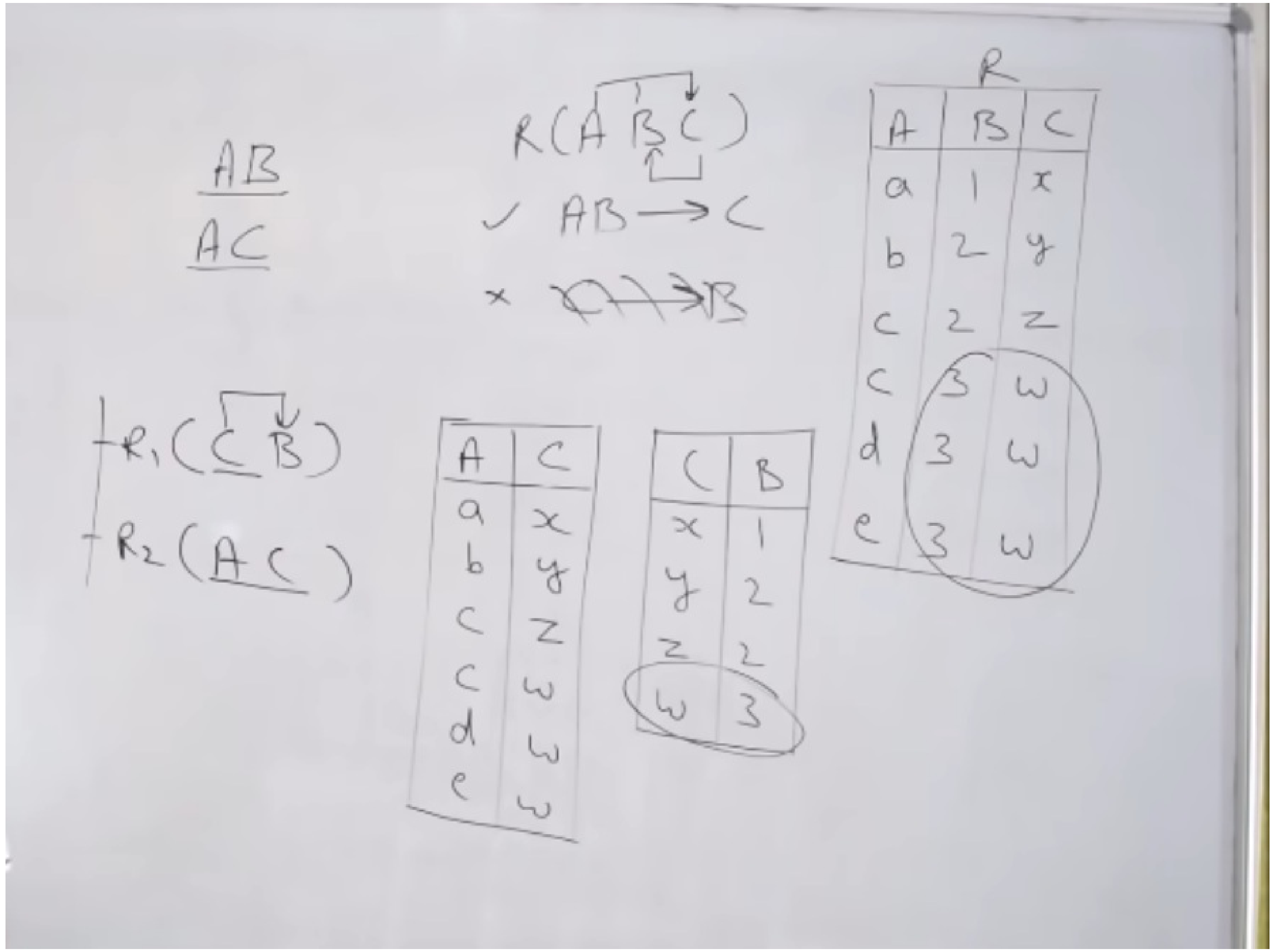
definition of BCNF

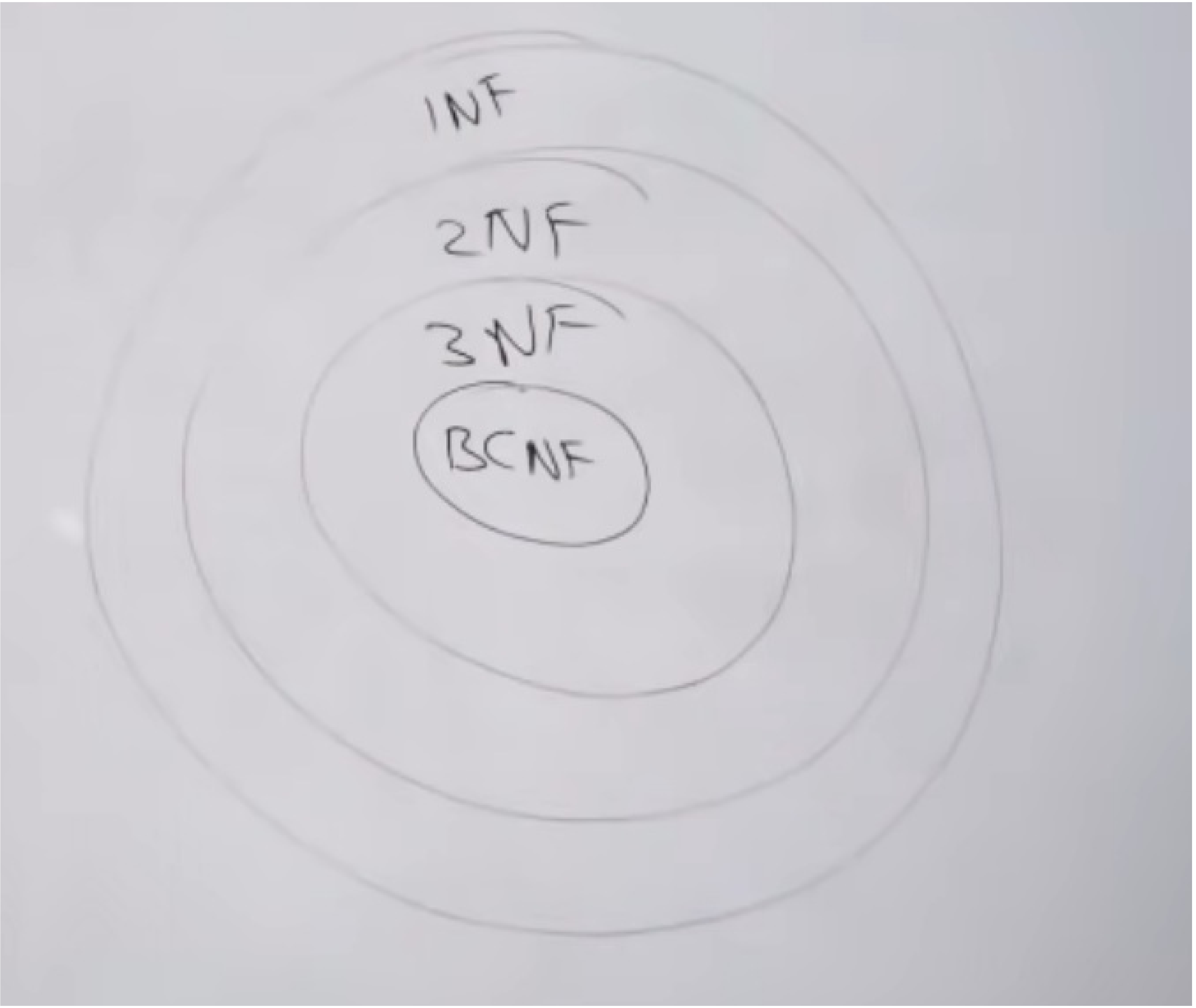
1NF → 2NF → 3NF

dependency preserving conversion

3NF → BCNF

dependency losses are possible




1NF
2NF
3NF
BCNF

Identify the Normal form

① R(ABCDEFGH)

$AB \rightarrow C$

$A \rightarrow DE$

$B \rightarrow F$

$F \rightarrow GH$

$(AB)^+ = ABCDEFGH$

CK

1NF

② R(ABCDE)

$CE \rightarrow D$

$D \rightarrow B$

$C \rightarrow A$

$CE \Rightarrow$ cand. Key

INF

R(ABCDEF)

$AB \rightarrow C$

$DC \rightarrow AE$

$E \rightarrow F$

$\left.\begin{array}{l}(ABD)^+ = ABCDEF \\ (BCD)^+ = ABCDEF\end{array}\right\}$ CK

INF

R(ABCDEGHI)

$AB \rightarrow C$

$BD \rightarrow EF$

$AD \rightarrow GH$

$A \rightarrow I$

ABD

$ABD \Rightarrow CK$

1NF

R(ABCDE)

$BC \rightarrow ADE$

$D \rightarrow B$

3NF

Essential = C

$(BC)^+ = ABCDE$

$(CD)^+ = ABCDE$

R(ABCDE)

$AB \rightarrow CD$

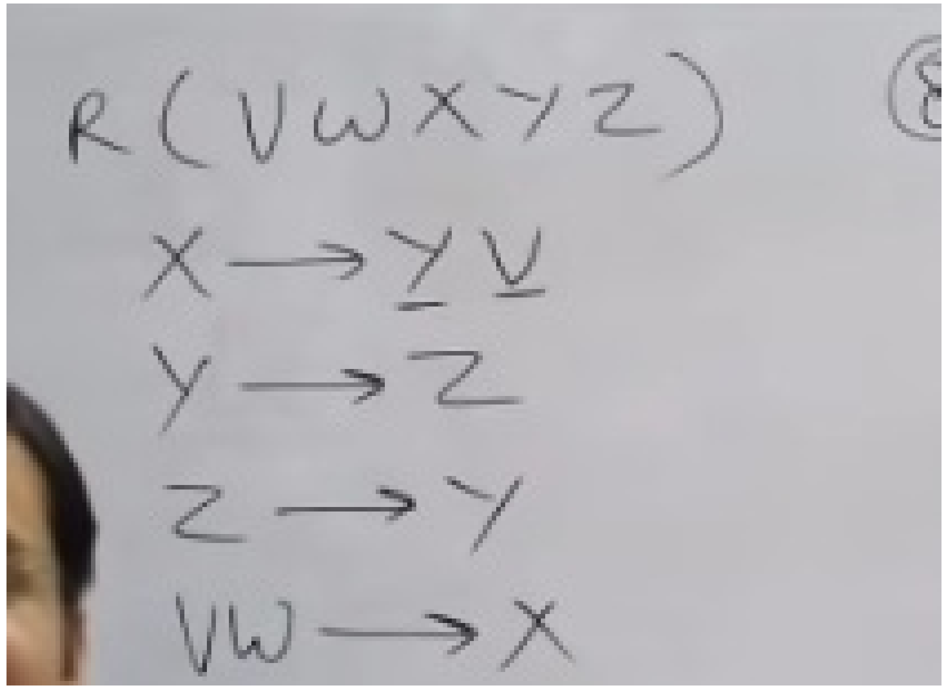
$D \rightarrow A$

$BC \rightarrow DE$

$$\left.\begin{aligned}(AB)^+ &= ABCDE\\ (BC)^+ &= ABCDE\\ (BD)^+ &= ABCDE\end{aligned}\right\} CK$$

Essential = B

3NF



Essentials = W

$(VW)^+ = VWXYZ \Rightarrow CK$

$(WX)^+ = VWXYZ \Rightarrow CK$

$(WY)^+ = WYZ$

$(WZ)^+ = WYZ$

INF

$R(ABCDEF)$

$AB C \rightarrow D$

$ABD \rightarrow E$

$CD \rightarrow F$

$CDF \rightarrow B$

$BF \rightarrow D$

$AC \rightarrow$ Essential

$(ABC)^+ = ABCDEF$

$(ACD)^+ = ABCDEF$

1NF

$R(ABC)$

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow A$

A,B,C

BCNF

① $R(ABCDEF)$

$A \rightarrow BCDEF$

$BC \rightarrow ADEF$

$DEF \rightarrow ABC$

$\left.A,\ BC,\ DEF\right\}$ CK

BCNF

Normal Form P

$R(ABC)$

$AB \rightarrow C$

$C \rightarrow A$

Essential = B

AB = ABC

BC = ABC

3NF

actice Question Part-2

R(ABCDE)

$A \to B$

$BC \to E$

$DE \to A$

ACD, BCD, CDE

3NF

④ R(ABCDE)

$AB \to CD$

$D \to A$

$BC \to DE$

AB, BC, BD

3NF

⑦ R(ABCDEF)

$AB \to C$

$DC \to AE$

$E \to F$

ABD, BCD

1NF

⑤ R(WXYZ)

$Z \to W$

$Y \to XZ$

$XW \to Y$

Y, XW, XZ

3NF

⑥ R(ABCDE)

$A \to B$

$B \to E$

$C \to D$

AC

1NF

⑧ R(VWXYZ)

$Z \to Y$

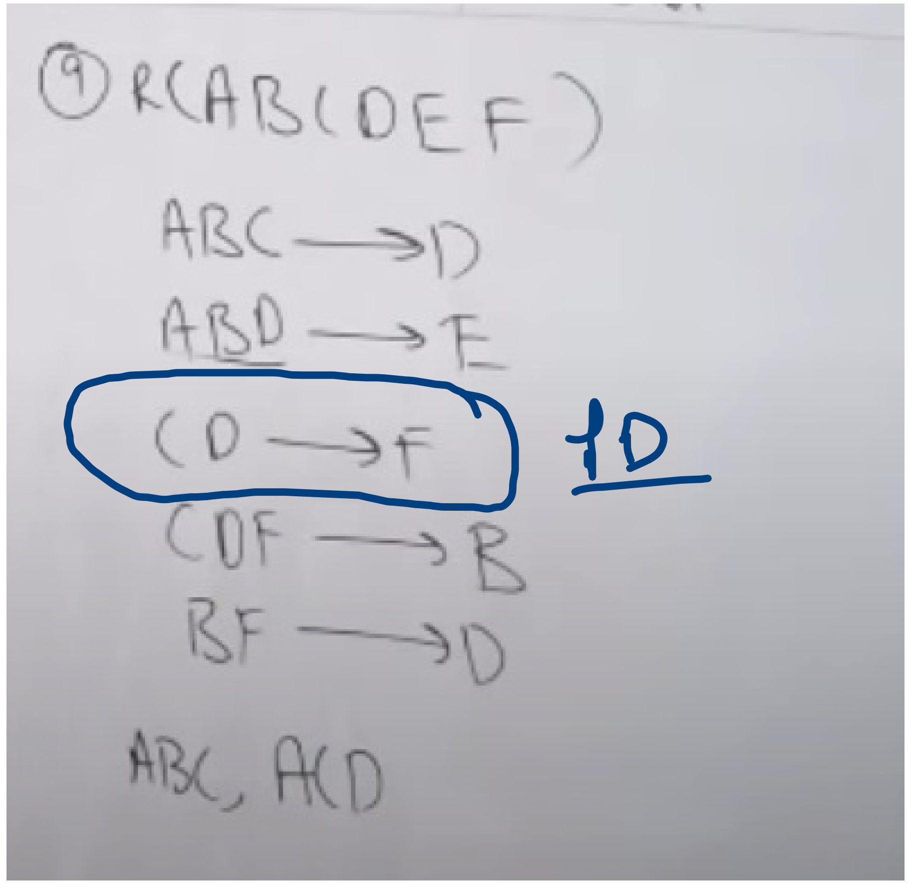
$Y \to Z$

$X \to YV$

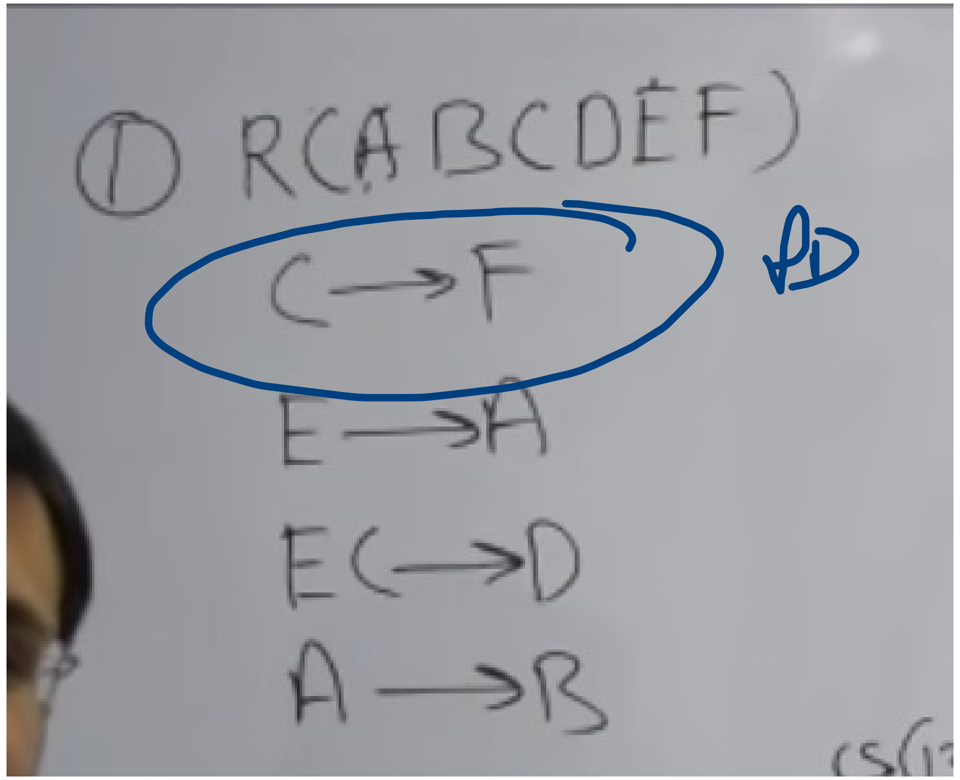
$VW \to X$

VW, XW

1NF



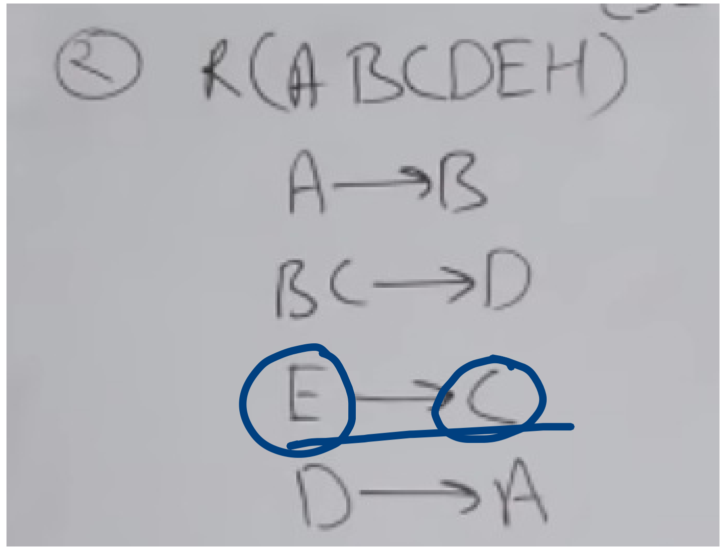
1NF



PD

CE ⇒ Cand key

1NF



1NF (EH) → Essential

$(AEH)^+ = ABEHCD$ } CK

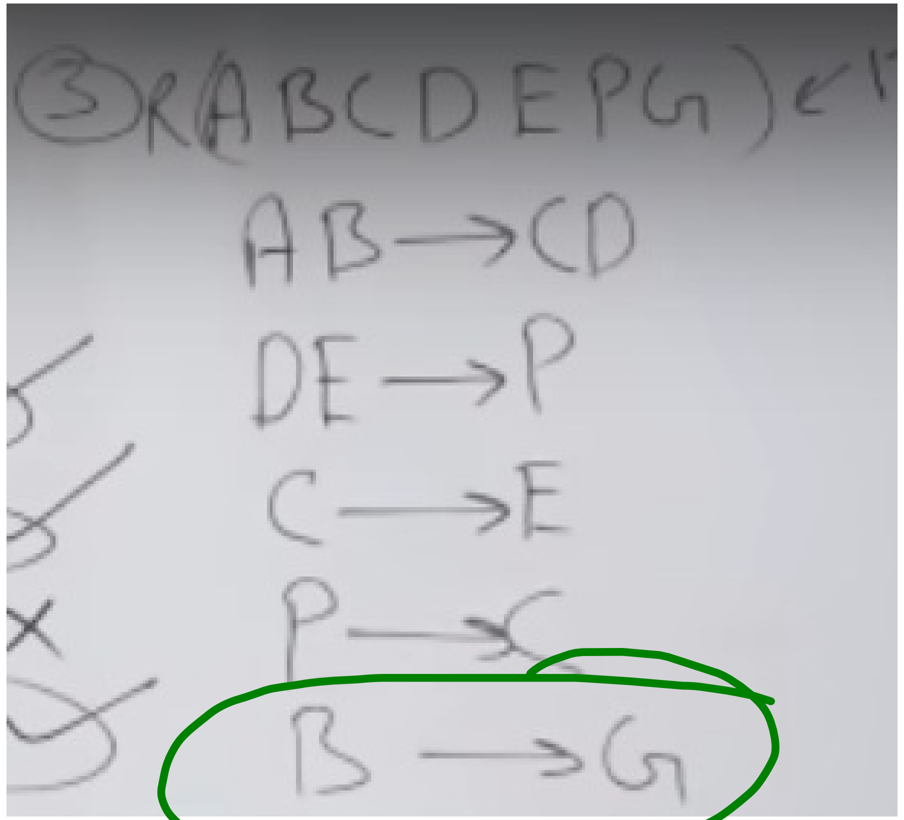
$(BEH)^+ = BEHCDA$

$(EH)^+ = EHC$

$(CEH)^+ = CEH$

ABDEH ⇒ PA NP C

$(DEH)^+ = DEHABC$ } CK



$(AB)^+ = ABCDEPG \Rightarrow CK$

1NF

Lossless Join decomposition.

Lossless join decomposition / non-additive:- this property guarantees that the extra or less tuple generation problem does not occur after decomposition

→ It is a mandatory property must always holds good.

→ if a relation R is decomposed into two relations R_1 & R_2, then it will be loss-less iff

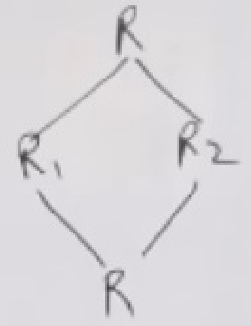
(i) $attr(R_1) \cup attr(R_2) = attr(R)$

(ii) $attr(R_1) \cap attr(R_2) \neq \phi$

(iii) $attr(R_1) \cap attr(R_2) \rightarrow attr(R_1)$

or

$attr(R_1) \cap attr(R_2) \rightarrow attr(R_2)$



R

A	B	C	D
1	a	p	x
2	b	q	y

R_1

A	B
1	a
2	b

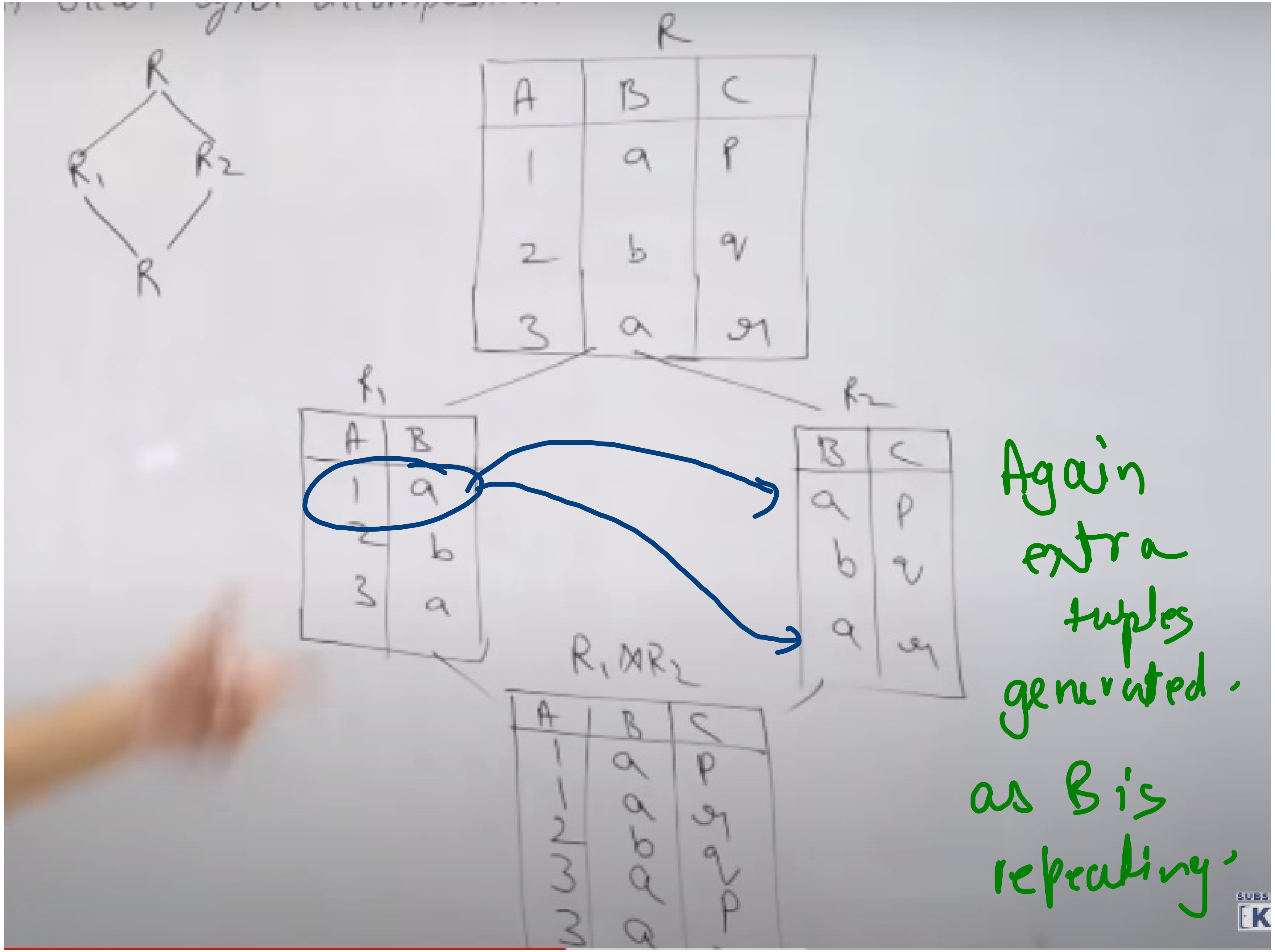
R_2

C	D
p	x
q	y

A	B	C	D
1	a	p	x
1	a	q	y
2	b	p	x
2	b	q	y

} This is data loss as per Rule 2.

KG



Again extra tuples generated. as B is repeating.

So, follow Rule 3.

Common Attr should be key.

/ Example Practice Problem of Lossless Join Decomposition Part-2

①

A	B	C	D	E
a	122	1	p	w
b	234	2	q	x
a	568	1	q	y
c	347	3	s	z

① $R_1(AB), R_2(CD)$ Lossy

② $R_1(ABC), R_2(DE)$ Lossy

③ $R_1(ABC), R_2(CDE)$ Lossy

④ $R_1(ABCD), R_2(ACDE)$

Lossless

⑤ $R_1(ABCD), R_2(DE)$

Lossless

⑥ $R_1(ABC)\, R_2(BCD)\, R_2(DE)$

Lossless

$R(VWXYZ)$

$Z \rightarrow Y$

$Y \rightarrow Z$

$X \rightarrow YV$

$VW \rightarrow X$

① $R_1(VWX)\ R_2(XYZ)$

② $R_1(VW)\ R_2(YZ)$

③ $R_1(VWX)\ R_2(YZ)$

④ $R_1(VW)\ R_2(WXYZ)$

$R(VWXYZ)$

$Z \to Y$

$Y \to Z$

$X \to YV$

$VW \to X$

find CK & then study

① $R_1(VWX)$ $R_2(XYZ)$

② $R_1(VW)$ $R_2(YZ)$

③ $R_1(VWX)$ $R_2(YZ)$

④ $R_1(VW)$ $R_2(WXYZ)$

May or may not be lossy.

$(f_1 \cup f_2)^+ = f^+$

Dependency Preserving $\Rightarrow$ optional property however lossless join is mandatory