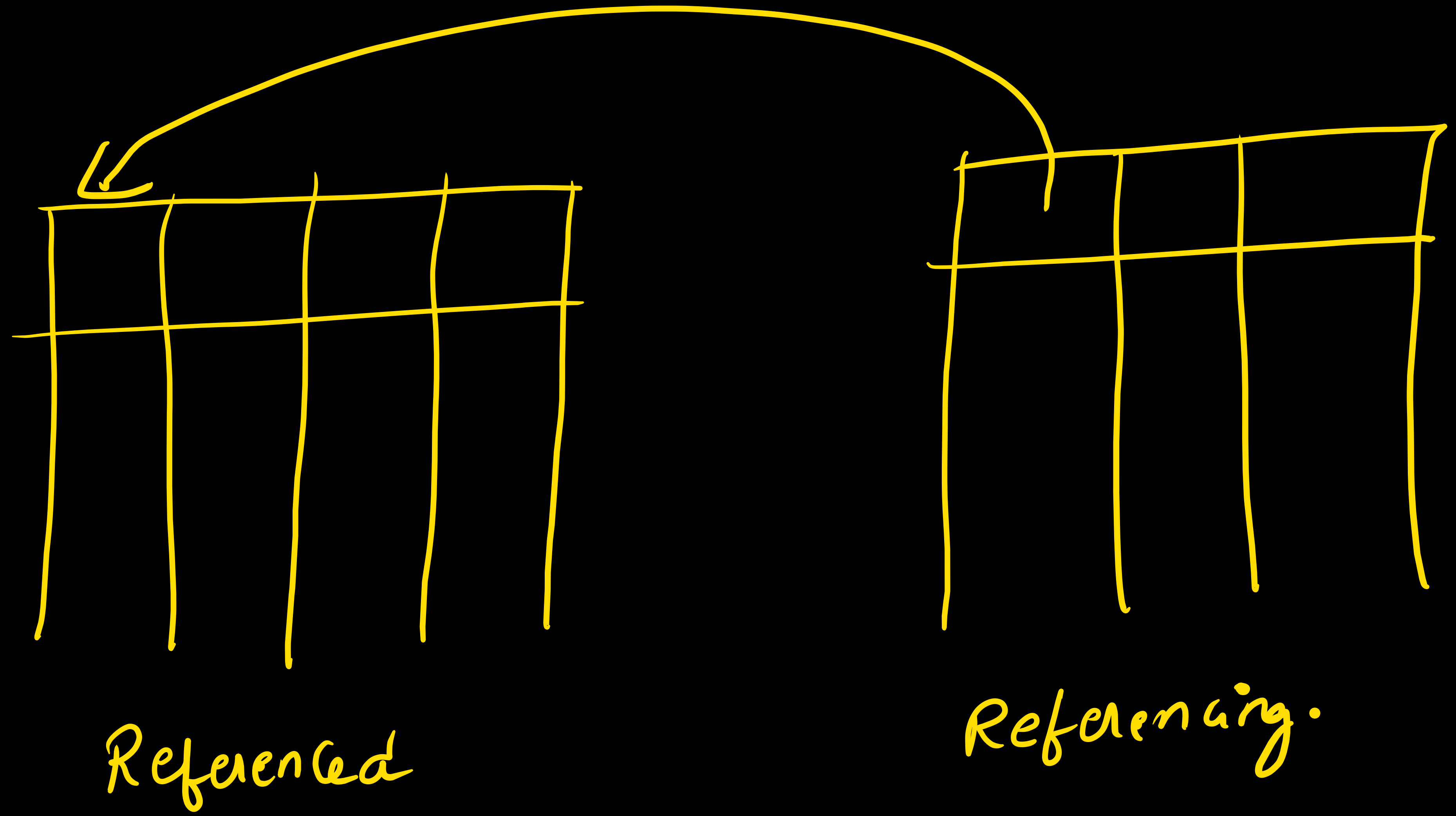
Referencial integrity Constraints. (FKIC)

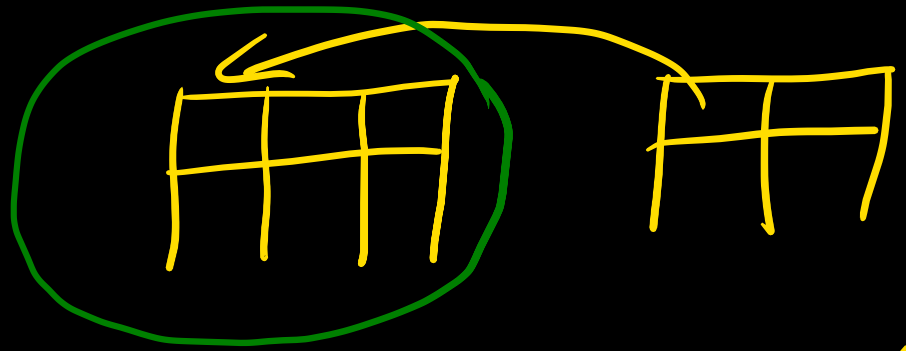
Referencial → FK

integrity → Correctness

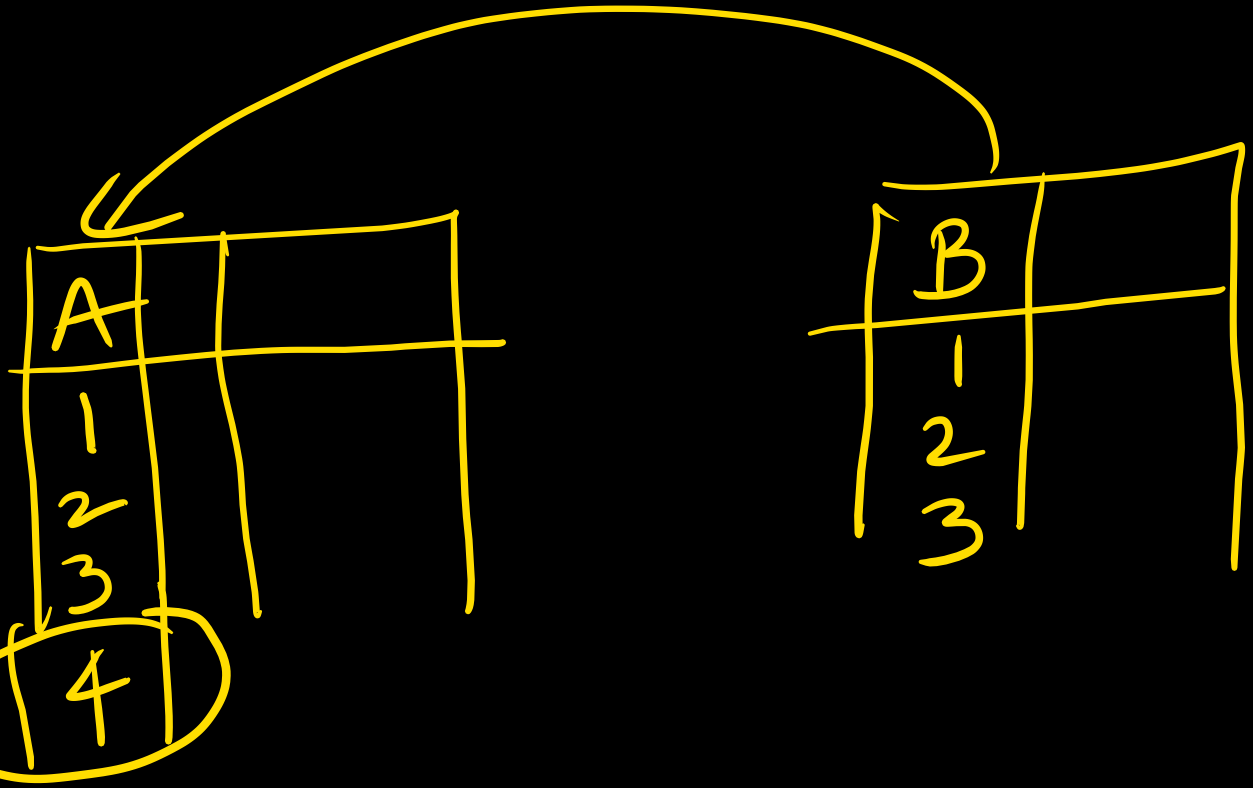
Constraints → Conditions



Referenced Relation:



Insertion → Does not cause any problem or any violation of FKIC.



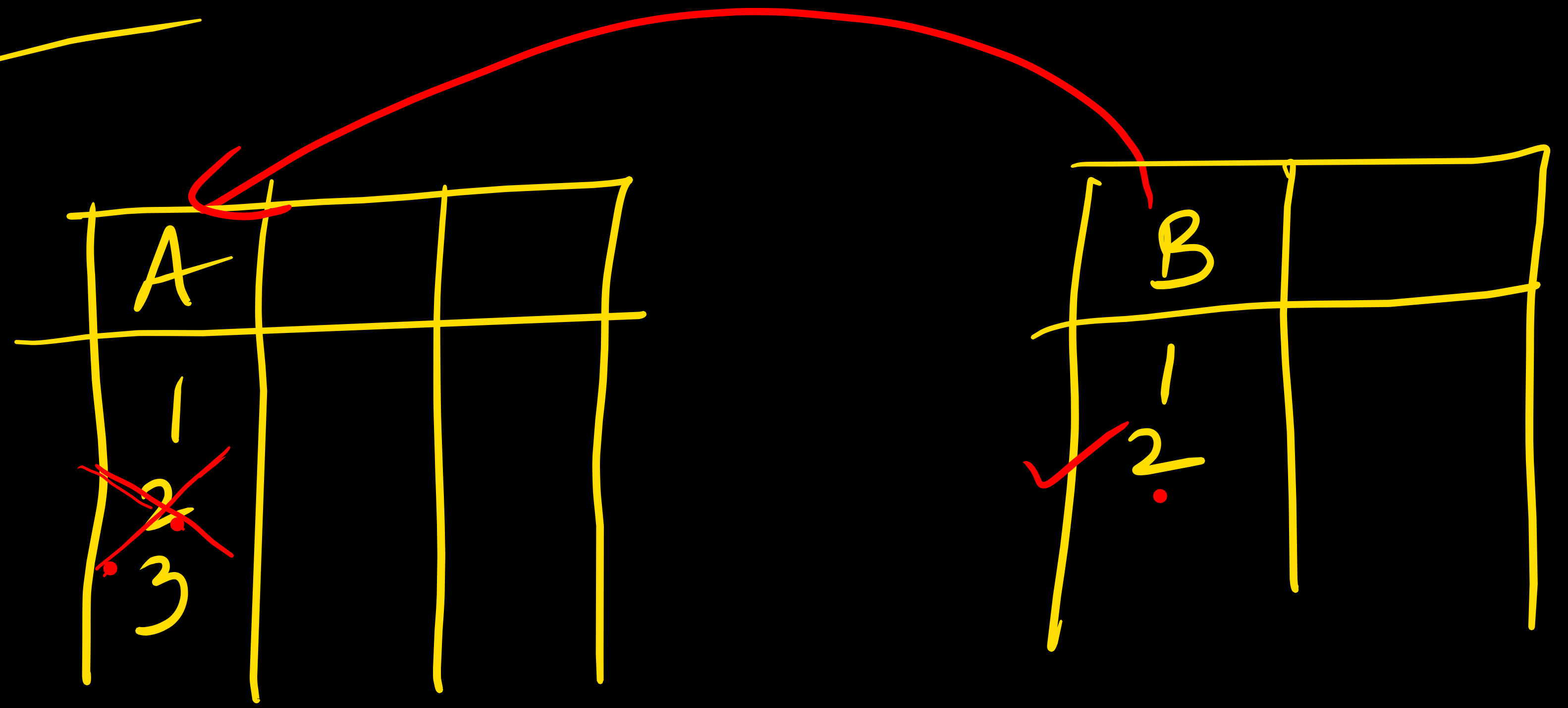
does not cause any problem.

Referenced Relation:

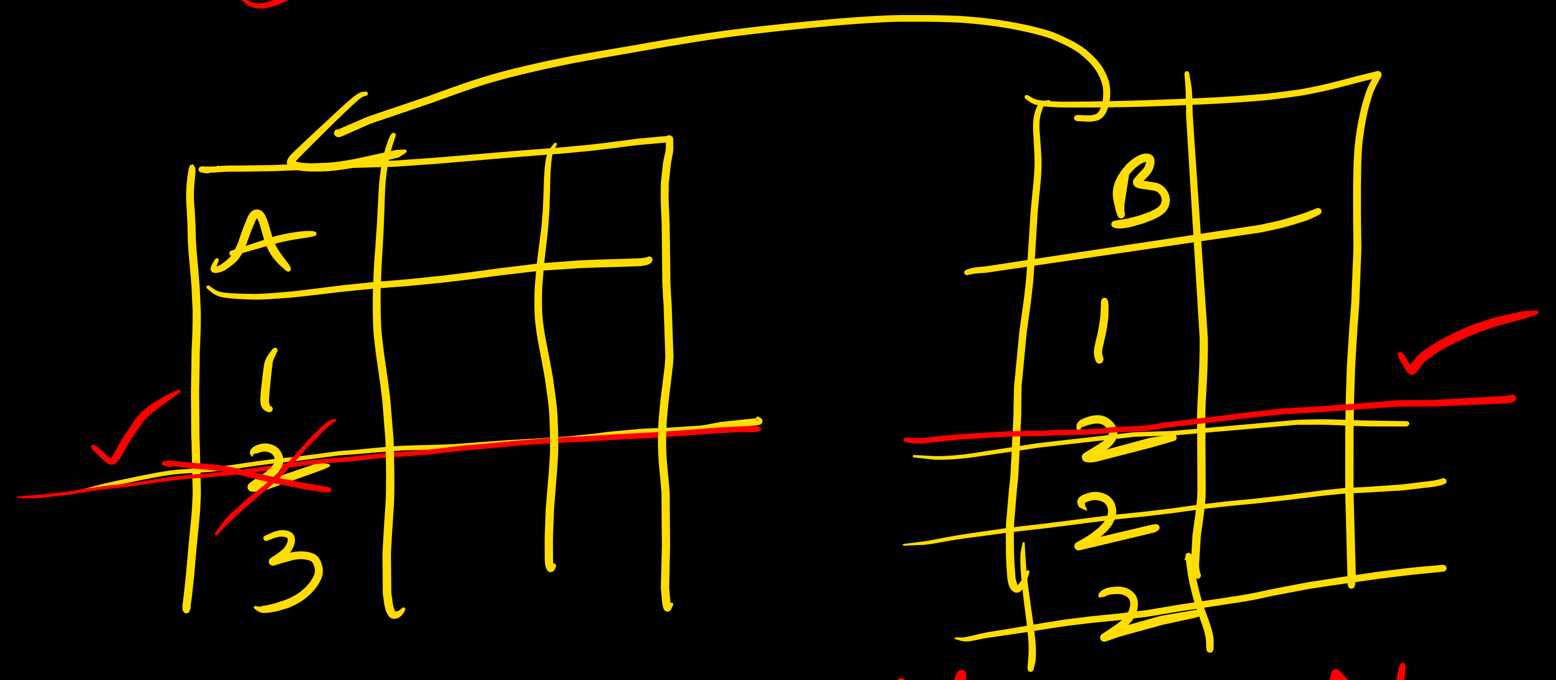
Delete: may cause voilation

Three solutions:

1) on delete no action (no deletion is allowed)

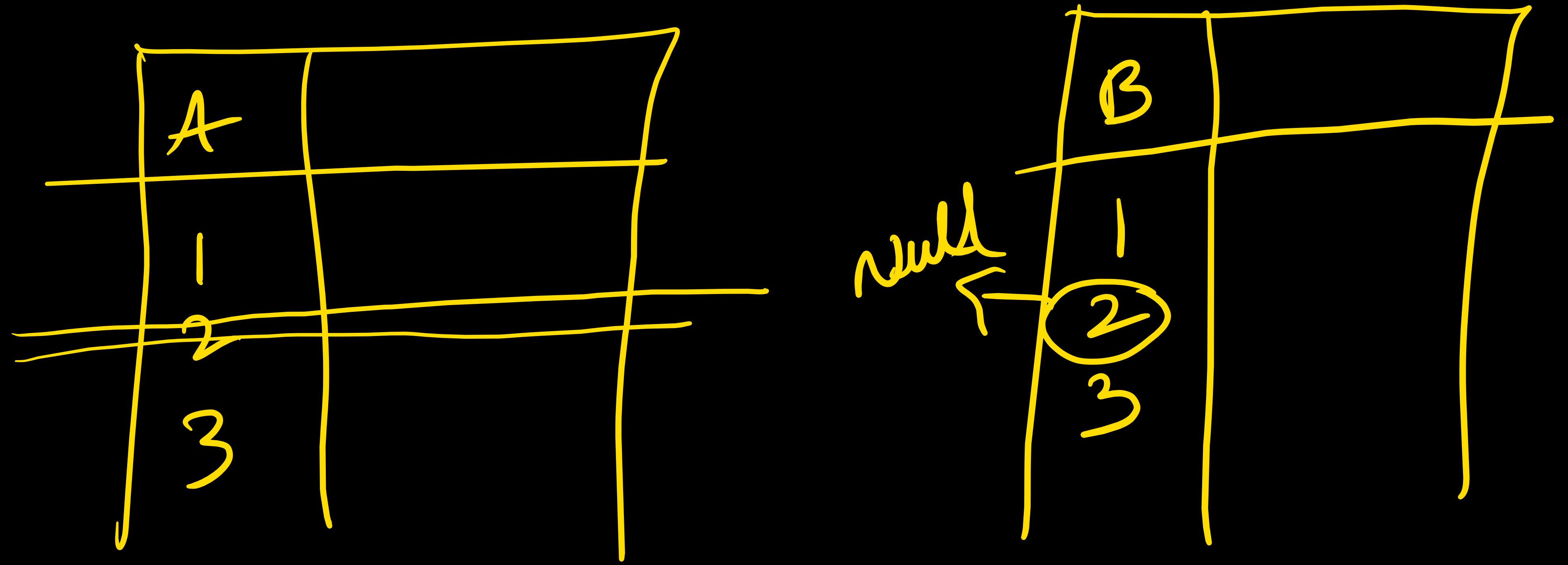


2) on delete cascade



on deletion of referenced relation record, we are forced to delete all related referencing record.

3) on delete set null :



~~Both~~ But which option to follow?

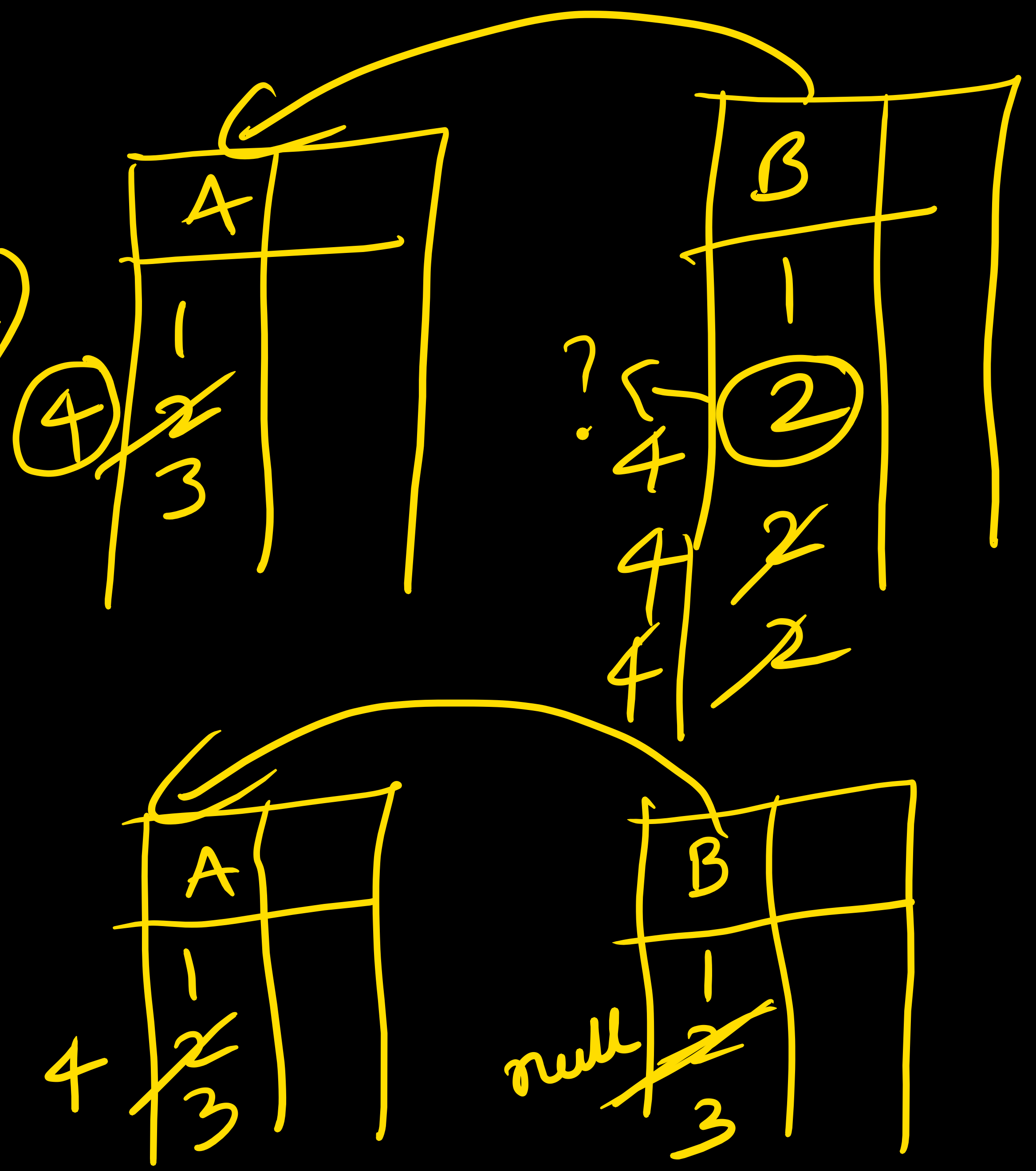
DBMS will give you 3 options. It is upto you.

Referenced relation:

updation:

- on update no action
  (no updation allowed)
- on update cascade
- on update set null.

choose



Normalization → 2 to 4 marks → easy to score.

1) Introduction to DB anamalies
2) Functional dependency
3) Attribute closure
   membership test
4) Finding candidate keys
5) Lossless join and dependency preserving
6) Normal forms
   1NF, 2NF, 3NF, BCNF

In dbms
→ no confusion

DBMS, Digital → easiest.
100%

7) Finding highest NF of Relational schema

8) Decomposition into higher NFS

9) MVD & 4NF → not important for gate.

10) Canonical cover of FD set.
(minimal cover)

Normalization:

→ It is used to eliminate / reduce redundancy in DB tables.

→ If two or more independent relations are stored in single relation, then it forms redundancy.

not Redundancy

Sid	Sname	DOB	CID	Cname	Ins	fee
S1.	A.	1990	C1	DB	Korth	100
S2.	A.	1991	C1	DB	Korth	100
S3	B	~~1992~~ 1990	C1	DB	Korth	100
S3	B	1992	C2	Algo	Cormen	200
S3	B	1992	C3	OS	galvin	300.

Redundancy

How do you eliminate Redn?

By splitting the table.

This is called normalization.

Problems because of Redundancy & DB anamolies:-

→ **Insertion anamoly:**
In order to insert a course, we have to insert a student also.
Same way for student.

→ **Deletion anamolies:-** If we delete student S_3, we will also lose the course info.

→ **Updation anamolies:-** If one row is updated, other rows will fail to udate. This causes inconsistency.

All these problems are because of redundancy.

If there is 0% redundancy in DB tables, there will be no anomalies.

How to avoid anomalies:-

Decompose relation R into sub rel $R_1, R_2, \ldots\ldots R_n$ to eliminate or reduce redundancy.

This is called normalized DB.

intuition → decompose.

There are rules we can follow

Normalization

PK

Sid	Sname	DB
1		
2		
3		
4		

Student

Sid (FK)	Cid (FK)
1	1
1	2
2	1
2	2
2	3

not redundant.
It is required.

PK

Cid	Cname	Ins	fee
1			
2			
3			
4			
5			

Courses.

A student can enroll in ~~[illegible]~~ 1 or more courses

A course may have 0 or more students.

Functional dependency (FD):

X, Y are some sets of attributes of Relation 'R', t_1, t_2 are any tuples of R.

$X \rightarrow Y$ is fd in Rel R, if $t_1 . X = t_2 . X$ then $t_1 . Y = t_2 . Y$.

X	Y
x_1	y_5
x_1	y_5
x_2	y_2

X	Y
x_1	y_5
x_1	y_5
x_2.	y_2.
x_3	y_2
x_4	y_4
x_2.	y_2.
x_1	y_5

$X \to Y.$

whenever x is same y should be same

whenever x is not same, then y can be same or not same

$Y \to X$

Stud

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S1	A	C3
S2	B	C4
S2	B	C1
S3	C	C2
S4	C	C3

Sid → Sname ✓

Sname → Sid ✗

Cid → Sid

$C_1 \to S_1$

$C_1 \to S_2$ ✗

Types of functional dependency:

- Trivial FD ✓
- non trivial FD ✓
- Semi non trivial FD ✓

Trivial FD:

Self FD:

$X \to Y$ is trivial if

$X \supseteq Y$

Sid $\to$ Sid

Sname $\to$ Sname

Sid Sname $\to$ Sname

Sid Sname $\to$ Sid Sname

Sid Sname $\supseteq$ Sname.

Non trivial FD :

$$\left.\begin{array}{l} \underline{\underline{Sid}} \rightarrow \underline{\underline{Sname}} \\ Sid\,Cid \rightarrow fee \end{array}\right\} \text{not trivial}$$

$X \rightarrow Y$ is non trivial FD if $\underline{\underline{X \cap Y = \phi}}$.

Semi non trivial:

$$\text{Sid} \rightarrow \text{Sid Sname} \quad \text{SNT}$$

$$\underbrace{\text{Sid} \rightarrow \text{Sid}}_{\text{T}}, \quad \underbrace{\text{Sid} \rightarrow \text{Sname}}_{\text{NT}}$$

$$\underline{\text{Sid Cid}} \rightarrow \text{Cid Sname}.$$

$$\text{Sid Cid} \rightarrow \text{Cid} \qquad \text{Sid Cid} \rightarrow \text{Sname}$$

$$\underline{X \rightarrow YZ} \Rightarrow \underline{X \rightarrow Y} \text{ and } \underline{X \rightarrow Z}$$

FD

$$X \rightarrow Y$$

$\rightarrow$ determines.

$X \rightarrow Y$ → not production.
→ FD.

Armstrong Rules over FD's :

let X, Y, Z be some attributes over 'R', then

(i) Reflexivity : (Trivial FD)

$X \rightarrow X$ always in R

(ii) Transitivity : If $(\underbrace{X \rightarrow Y} \ \&\& \ \underbrace{Y \rightarrow Z})$ then $(\underbrace{X \rightarrow Z})$

Augmentation:

If $X \to Y$ is in R, then $XZ \to YZ$ is in R.

Split rule:

If $X \to YZ$, then $X \to Y$, $X \to Z$

$\cancel{XY \to Z \quad \times \quad X \to Z \quad Y \to Z}$

You cannot split LHS.

You can only split RHS.

merge union:-

$X \to Y$, $X \to Z$ then $X \to YZ$.

**** Attribute closure:- (X^+)

X is some attribute set of Rel 'R'.

then X^+ = { Set of all attributes which can be determined by X }

FD Set = { $A \to B$, $C \to D$, $AB \to E$, $BE \to C$, $EF \to G$ }   R(ABCDEFG)

$A^+ = \{ A\ B\ E\ C\ D \}$

ABCDE ~~FG~~

No RHS Containing F.

G.   EF

AF is SK.

→ if AF is minimal

AF → Candidate key.

AF^+ { A F B E C G D }

$(AF)^+$ ABCDEFG ✓

Contains all Attri of R.

(AF) is Superkey of R.

AF is super key.

CK → <u>minimal</u> SK.

$(AF)^+ = (ABCDEFG)$.

$F^+ = (F)$

$A^+ = (ABECD)$

AF is minimal.

A^+ is not determining all attr.

F^+ is not determining all attr.

$\{A \to B,\ C \to D,\ AB \to E,\ BE \to C,\ EF \to G\}$

$(AF)^+ = \{A\,F\,B\,E\,C\,G\,D\}$

AFBECGD.

ABCDEFG.

CK $\leftarrow$ (AF) $\rightarrow$ superkey.

$(ABC)^+ = \{ABC$

Test CK.

$A^+ = \{ABECD\}$

$F^+ = \{F\}$

$(AD)^+ = \{AD$

$A \to B, \; C \to D, \; AB \to E, \; BE \to C, \; EF \to G$

$(BE)^+ = \{B\,E\,C\,D\}$

$A \to ABCDE.$

Superkey: X is some attribute Set of Relational schema 'R', X is a Superkey of 'R' if X^+ ~~must~~ determines all the attributes of R.

Ex: R(

Ex: $R(ABCDEF)$ $\{AB \to C, C \to D, D \to E, E \to F, F \to B\}$

$(AB)^+ = \{ABCDEF\}$

$\downarrow$

SK.

any superset of SK is a SK.

$AB + C = (ABC)^+ = \{ABCDEF\}$

$\therefore$

$(BC)^+ = \{BCDEF\}$

$\therefore$ BC is not SK.

Candidate key: (minimal SK)

X is CK of Rel R iff

① X must be superkey of Rel 'R'

ie $X^+ = \{$all attri of $R\}$

and

② no proper subset of X is a super key 'R'.

$\forall y \subset x$, Y^+ doesnt determine all attributes of R.

AB → Super key.

A^+ → not Super key

B^+ → not Super key.

AB → Candidate key.

$ABC \to SK.$

AB^+
BC^+
AC^+
A^+
B^+
C^+

$\left.\right\}$ not $\underline{\underline{SK}}$. then ABC is $\underline{\underline{CK}}$.

~~Ad~~ $R(ABCDE)$ $\quad \underbrace{AB \to C, \; C \to D, \; B \to E}$

Find CK of R?

Trick: ~~Find~~ Go to RHS of all FD's and find missing attributes.

These attributes should definitely be in CK.

more S.K? ✓

CK? ✓

$(AB)^+ = \{ABCDE\}$.

CK.

Test if CK.

$A^+ = \{A\}$ ✓

$B^+ = \{BE\}$. ✓

30th

2-hours

|

7 - 9

7 - 8 - DSA.

8 - 9 → DBMS.

weekends also we have dsa.

3 hours.

6 - 7 - CD - Sumith

2 hours mdg

6 - 8

6 - 8 → DSA projects

5 hours of dsa