Aditi Pandey

UNIT-3
(Database Design & Normalization)

Functional dependency :- (FD)

* FD play a key role in differentiating good d/b design from bad d/b design.

A FD is a constraint (means rules) that specifies the relationship b/w two set of attributes where one set accurately determine the values of other set.

→ It is represented by an arrow sign (→) that is, $X \rightarrow Y$, where X functionally determines Y.

Determinant (Attribute of L.H.S) — X

Dependent (Attribute on R.H.S) — Y

[It means X determines Y or Y is determined by X.]

FD : $X \rightarrow Y$

[if $t_1 \cdot x = t_2 \cdot y$
then $t_1 \cdot y = t_2 \cdot y$]

x	Y
1	1
2	1
3	2
4	3
2	5

$\boxed{X \rightarrow Y \text{ is not FD}}$

1- Types of FD :-

(i) Trivial :
(ii) Non-Trivial
(iii) Multi-valued
(iv) Transitive

(1) Trivial FD:- In Trivial FD, a dependent is always a subset of the determinant.
i.e. If $X \to Y$ then Y is subset of X. ($Y \subseteq X$)

e.g:

roll_no	name	age
42	abc	17
43	pqr	18
44	xyz	18

Here, {roll_no, name} → name is a trivial FD.

or e.g: $AB \to A$, $AB \to B$, $AB \to B$ are trivial functional dependency.

(2) Non-Trivial FD:- A FD $X \to Y$ is non-trivial if and only if $Y \not\subset X$ (the dependent is strictly not the subset of the determinant)

e.g: $AB \to BC$, $AB \to CD$ are non-trivial FD.

(3) Multi-valued FD:- In MVD entities of the dependent set are not dependent on each other.
i.e. If $a \to \{b, c\}$ & there exists no FD b/w b&c then it is called as MVD.

(4) Transitive FD:- In transitive FD, dependent is indirectly dependent on determinant
i.e. If $a \to b$ & $b \to c$, then $a \to c$. This is Transitive FD.

Scanned with OKEN Scanner

Canonical Cover:-

A canonical cover is a simplified and reduced version of the given set of functional dependencies. It is also called as <u>Irreducible set</u>.

Steps:- (i) Write the given set FD in such a way that each FD contains exactly one attribute on R.H.S.

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

(ii) Consider each FD one by one from the set in step 1. Determine whether it is essential or not (To find this compute closure of its L.H.S).

Case 1: Result comes out to be same. It means it is not essential. Eliminate that FD

Case 2: Result is Different: Thus, it is Essential.

(iii) Consider the newly obtained set of FD. Check if there is any FD that contains more than 1 att. of L.H.S.

Case 1: No- Then the step 2 result set is canonical cover.

Case 2: Yes: Reduce the L.H.S of all FD one by one.

Scanned with OKEN Scanner

Q. R(W, X, Y, Z) and FD:-
$X \to W$, $WZ \to XY$, $Y \to WXZ$.
Find canonical cover?

Soln: (i)
$X \to W$
$WZ \to X$
$WZ \to Y$
$Y \to W$
$Y \to X$
$Y \to Z$

Write FD as exactly one att. on R.H.S.

Step ii) For $X \to W$:-
$X^+ = \{X, W\}$
Ignore $X \to W$, $X^+ = \{X\}$
not same. So, $X \to W$ is essential

For $WZ \to X$
$\{W.Z\}^+ = \{W, Z, X, Y\}$
Ignoring $WZ \to X$, $(WZ)^+ = \{W, Z, X, Y\}$
same.
So, $WZ \to X$ not essential.

For $WZ \to Y$: $\{WZ\}^+ = \{W, X, Y, Z\}$
Ignore $WZ \to Y$, $\{WZ\}^+ = \{W, Z\}$
So it is essential.

$X \to W$
$WZ \to Y$
$Y \to W$
$Y \to X$
$Y \to Z$

For $Y \to W$: $Y^+ = \{W, X, Z, Y\}$
Ignore $Y \to W$, $Y^+ = \{Y, X, Z, W\}$
same
eliminate it,

For: $Y \to X$, $Y^+ = \{Y, X, Z, W\}$
Ignor $Y \to X$, $Y^+ = \{Y, Z\}$
Similar $Y \to Z$ is also essential.

Now:
$X \to W$
$WZ \to Y$
$Y \to X$
$Y \to Z$

Step3: F.D having more than one att. on L.H.S

$WZ \rightarrow Y$

$WZ^+ = \{W, X, Y, Z\}$

$W^+ = \{W\}$, $Z^+ = \{Z\}$

None of the subset have same result.

So, Canonical cover is →

$$\boxed{\begin{array}{c} X \rightarrow W \\ WZ \rightarrow Y \\ Y \rightarrow X \\ Y \rightarrow Z \end{array}}$$ Ans.

Q.2. $F = \{AB \rightarrow C, C \rightarrow AB, B \rightarrow C, ABC \rightarrow AC, A \rightarrow C, AC \rightarrow B\}$

Step1: $= \{\cancel{AB \rightarrow C}, C \rightarrow A, C \rightarrow B, B \rightarrow C, \cancel{ABC \rightarrow A}, \cancel{ABC \rightarrow C}, A \rightarrow C, AC \rightarrow B\}$

Step2: $= \{ C \rightarrow A, C \rightarrow B, B \rightarrow C, A \rightarrow C\}$

Q3 $F = \{A \rightarrow B, C \rightarrow B, D \rightarrow ABC, AC \rightarrow D$

Soln: (i) $\{A \rightarrow B, C \rightarrow B, D \rightarrow A, \cancel{D \rightarrow B}, D \rightarrow C, AC \rightarrow D\}$

(ii) Remove redundant FD

$\{A \rightarrow B, C \rightarrow B, D \rightarrow A, D \rightarrow C, AC \rightarrow D\}$ $AC^+ = \{A, C, B\}$

D nahi aa raha.

$A^+ \{ B, C, D\}$

(iii) L.H.S have 1 att.

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

Ans

$AC \rightarrow D$

Remove A

$C^+ = \{C, B\}$ A nahi aa raha

Remove C

$A^+ = \{A, B\}$ C nahi aa raha

So, we can't remove.

Rules/ Axioms (Armstrong's axioms) : ~~Rules~~:-

(1) Reflexivity:- If ~~B~~ Y is a subset of X then $X \to Y$.

(2) Augumentation:- If $X \to Y$ then $XY \to YZ$.

(3) Transitive : If $X \to Y$ and $Y \to Z$ then $X \to Z$.

(4) Union : If $X \to Y$ & $X \to Z$ then $X \to YZ$.

(5) Decomposition: If $X \to YZ$ then $X \to Y$ & $X \to Z$

(6) Pseudotransitivity: If $X \to Y$ & $WY \to Z$ then $XW \to Z$.

(7) Composition: If $X \to Y$ & $Z \to W$ then $XZ \to YW$

* **Attribute Closure**:- Att. closure of an attribute set can be defined as set of all that attributes which can be functionally determined from it.

→ The closure of an attribute set {X} is denoted as $\{X\}^+$.

Steps to find closure of an Att. set:-

(1) Add the attributes contained in the attribute set for which closure is being calculated ~~it~~ to the result set.

(2) Recursively add the attributes to the result set which can be functionally determined from the attributes already contained in the result set

② Second NF (2NF) :-

→ It is 1NF

→ No Partial dependency in the relⁿ.

$\Downarrow$

$\boxed{\text{Proper subset of C.K} \rightarrow \text{Non-Prime attribute}}$

How to find candidate key.

Q. R(ABCD) F: $BC \rightarrow A,\ AD \rightarrow B,\ CD \rightarrow B,\ AC \rightarrow D$

Find all candidate keys of Relation R.

→ Look for the att. which is not present in R.H.S

→ find closure of the att. / set of att. which are not present in R.H.S.

↳ if closure contain all att. then it will be the only key.

↳ find combination of remaining att. with att. not present in R.H.S.

$C^+ = \{C\}$

$(AC)^+ = \{A, C, D, B\}$ C.K

$(BC)^+ = \{B, C, A, D\}$ C.K

$(CD)^+ = \{C, D, B, A\}$ C.K

So there are three candidate keys.

AC, BC, CD.

Q. R(A, B, C, D, E) FD: $\{A \rightarrow B,\ D \rightarrow E\}$. Find C.K.

$(ACD)^+ = \{A, C, D, B, E\}$ // Only 1 candidate key.

Prime att. = $\{A, C, D\}$.

Non. Prime att. = $\{B, E\}$.

Scanned with OKEN Scanner

e.g: Consider a relⁿ R(A, B, C, D, E, F, G) with FD – A→BC, BC→DE, D→F, CF→G. Find the closure of attribute A, D, {BC}.

Soln:
$A^+ = \{A\}$
$= \{A, B, C\}$ Using A→BC
$= \{A, B, C, D, E\}$ BC→DE
$= \{A, B, C, D, E, F\}$ D→F
$\boxed{A^+ = \{A, B, C, D, E, F, G\}}$ CF→G.

$D^+ = \{D, F\}$
$\{B, C\}^+ = \{B, C, D, E, F, G\}$.

Note: If the closure att. set contain all the att. of the relⁿ, then that att. set is called as Super Key of that relⁿ.

Candidate Key:- If there exists no subset of an attribute set whose closure contains all the attributes of the relⁿ, then that att. set is called as a Candidate Key of that relⁿ.

Q. Find closure of an attribute : $(CF)^+$ & $(BG)^+$.
AB→CD, AF→D, DE→F, C→G, F→E, G→A.

Ans: $\{CF\}^+ = \{A, C, D, E, F, G\}$
$\{BG\}^+ = \{A, B, C, D, G\}$.

Normalization:-

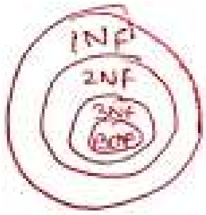
Normalization is a database design technique that reduces data redundancy and ensuring the integrity of data.

→ Normalization divides the big table into small table.

→ It is the process of minimizing redundancy from a relation. Normal forms are used to eliminate redundancy in d/b tables.

Types of Normal Form:-

1NF 2NF 3NF BCNF



① **First Normal Form (1NF):-** A relation is in 1NF if every attribute in that relation is single valued or atomic.

ID	Name	Course
1	A	C1, C2
2	E	C3
3	M	C2 C3

NOT in 1NF.

→ 1NF →

ID	Name	Cours
1	A	C1
1	A	C2
2	E	C3
3	M	C2
3	M	C3

1NF

② **Second (2NF):-** A rel^n must be in 1f and relation must not contain any partial dependency i.e., no non-prime attribute is dependent on any proper subset of any CK of the table.

→ Non-Prime att. (att. which are not part of any CK).

First Normal Form :-

A relation is in 1NF if domain of each attribute contain only atomic values.

e.g:

Sid	Sname	Address	P.No.
1	Jenny	Haryana, India	P1, P2
2	Jiya	Punjab, India	P3
3	Payal	Raj, India	P4, P5
4	Shanvi	Haryana, India	P7

This is not in 1NF.
~~b/z each attri~~

⇓ 1NF

Sid	Name	State	Country	Phone No.
1	Jenny	HR	India	P1
1	Jenny	"	"	P2
2	Jiya	PUNJAB	"	P3
3	Payal	RAJ	"	P4
3	Payal	RAJ	"	P5
4	Shanvi	HR	"	P7

(iii) 3NF :- No transitive dependency for non-prime att. as well as it is in 2NF.

A relⁿ is in 3NF if at least any one condⁿ holds:-

$X \rightarrow Y$

(i) X is super key.

(ii) Y is a prime attribute.

(iii) Transitive dependency. If $A \rightarrow B$ & $B \rightarrow C$ then $A \rightarrow C$.

(iv) BCNF :- Strict version of 3NF.

(Boyce-Codd) A relⁿ is in BCNF iff in every non-trivial FD $X \rightarrow Y$, X is super key.

Note: $\boxed{\text{* If all attributes of a rel}^n \text{ are prime att. then it is always in 3NF. * Every rel}^n \text{ at least in 1NF.}}$

Identify highest Normal Form:-

Q. R(ABCDEFGH)

FD: (ABC → DE, E → GH, H → G, G → H, ABCD → EF).

Soln: 1. Find out all C.K
2. Find out Prime & non-Prime att.
3. Then apply rules of NF.

① C.K $(ABC)^+ = \{A, B, C, D, E, F, G, H\}$.

② P.A = {A, B, C}
N.P.A = {D, E, F, G, H}

③ BCNF: L.H.S must be super key.

ABC → DE ✓, E → GH ✗, H → G ✗
G → H ✗, ABCD → EF ✓
This is not in BCNF

3NF: No Transitive dependency.
either L.H.S is S.K or R.H.S P.A

E → GH ✗ G → H ✗
H → G ✗
∴ Not in 3NF

2NF: No Partial dependency.
Proper subset C.K → Non P.A.

E → GH ✓ H → G ✓, G → H ✓

∴ This is in 2NF.

Ques: Relation schema $R(A,B,C,D,E,F,G,H,I,J)$

$F = \{AB \to C,\ A \to DE,\ B \to F,\ F \to GH,\ D \to IJ\}$.

Find normal form of R?

Soln: C.K:-

$(AB)^+ = \{A, B, C, D, E, F, G, H, I, J\}$

P.A = $\{A, B\}$

N.P.A = $\{C, D, E, F, G, H, I, J\}$

Check BCNF :- L.H.S must be super key.

$AB \to C$ ✓ $\quad B \to F$ ✗ $\quad D \to IJ$ ✗

$A \to DE$ ✗ $\quad F \to GH$ ✗

Not in BCNF

3NF:- L.H.S is S.K or R.H.S P.A

$B \to F$ ✗ , $A \to DE$ ✗ , $F \to GH$ ✗,

$D \to IJ$ ✗

Not in 3NF.

2NF:- ~~L.H.S~~ $\boxed{\text{Proper subset of C.K} \to \text{R.H.S N.P.A}}$ ✗

$B \to F$ ✓, $A \to DE$ ✓, $F \to GH$ ✗

$D \to IJ$ ✓

Not in 2NF.

∴ This in 1NF

Q. R(A, B, C, D)

F :- {AB → CD, AC → BD, BC → D}

Soln: C.K :-

$A^+ = \{A\}$ ×

$AB^+ = \{A, B, C, D\}$ ✓ C.K

$AC^+ = \{A, C, B, D\}$ ✓ C.K

$AD^+ = \{A, D\}$ ×

C.K = AB, AC

P.A = {A, B, C}

N.P.A = {D}

BCNF :- L.H.S S.K.

AB → CD ✓, AC → BD ✓, BC → D ×.

3NF :- L.H.S S.K or R.H.S P.A.

BC → D ×

2NF : $\boxed{\text{Proper subset of C.K → N.P.A}}$ ×

BC → D ✓ b/z BC is proper subset

~~But proper~~

(AB ∪ AC) (A, B, C) ✓

So this is Partial dependency.

So this is not in 2NF.

∴ $\boxed{\text{This is 1NF.}}$

Note: (1) If all C.K are single att. then it would be in 2NF.

(2) If all att. of a relⁿ are P.A then it would be in 3NF.

(3) If relⁿ is in 3NF & all C.K are single then it is in BCNF.

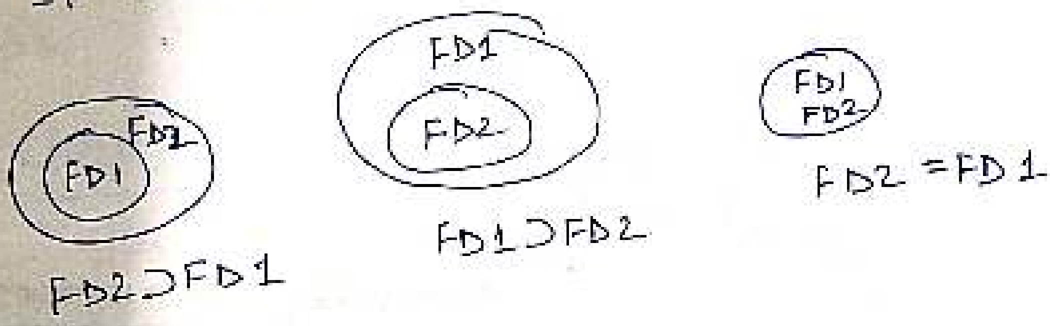
Equivalence of FD:-

Let FD1 and FD2 are two sets for rel^n R.

(1) If all FDs of FD1 can be derived from FDs present in FD2, $\boxed{FD2 \supset FD1}$

(2) If all FDs of FD2 can be derived from FDs present in FD1, $\boxed{FD1 \supset FD2}$ superset

(3) If 1 and 2 both are true, $\boxed{FD1 = FD2}$



Q. $R(A, B, C, D)$; $FD1 : \{A \to B, B \to C, AB \to D\}$

$FD2 : \{A \to B, B \to C, A \to C, A \to D\}$

Soln:

Step1: Checking $FD2 \supset FD1$:-

$A \to B$, $B \to C$ are present in FD2

$AB \to D$ is not directly present in FD2.

So, by using FD2,

$(AB)^+ = \{A, B, C, D\}$, So, $AB \to D$ present

$\therefore$ $\boxed{FD2 \supset FD1}$

step2: Checking $FD1 \supset FD2$:-

$A \to B$, $B \to C$ are present in FD1

$\left.\begin{matrix} A \to C \\ A \to D \end{matrix}\right\}$ is not directly present in FD1.

So, by using FD1

$A^+ = \{A, B, C, D\}$ So, $A \to C$ is present

$A \to D$ is present

$\therefore$ $\boxed{FD1 \supset FD2}$ So, $\boxed{FD1 = FD2}$ Ans.

Q: $R(A, C, D, E, H)$.

$F: \{A \to C, AC \to D, E \to AD, E \to H\}$ Check $F = G$ or not?

$G: \{A \to CD, E \to AH\}$

Soln:- Check $F \supset G$:-

$A \to CD$ is not directly present in F.

So, by using F:-

$A^+ = \{A, C, D\}$

$E^+ = \{E, A, D, H, C\}$

So, $\boxed{F \supset G}$

Check $G \supset F$:-

$A \to C$ is not directly present by using G

$A^+ = \{A, C, D\}$, $AC^+ = \{A, C, D\}$.

~~AC → D is present.~~

$E^+ = \{E, A, C, D, H\}$ $E \to AD, E \to H$ using set G.

So, $G \supset F$

Therefore; $\boxed{F = G}$

Scanned with OKEN Scanner

Decomposition :-

It is a process of dividing a single relation into two or more sub-relations.

There are two types :-

① Lossless Join Decomposition

② Lossy Join Decomposition.

*Why ? ⇒ The decomposition is required when the relational model is not in appropriate normal form. It is used to eliminate the problems ~~like~~ of bad design like redundancy :- inconsistencies, anomalies & redundancy.

① Lossless Join decomposition :-

→ If the infoⁿ. is not lost from the relation that is decomposed, then the decomposition will be lossless.

→ This decomposition gurantees that the join of relations will result in the same relation as it was decomposed.

$$\boxed{R_1 \bowtie R_2 \bowtie R_3 \cdots \bowtie R_n = R}$$

$\bowtie$ → natural join operator.

→ Also known as non-additive Join decomposition.

Contd's:-

Dependency Preservation:-

It ensures that :-

- None of the FD that holds on the original relation are lost.
- The sub-relations still hold or satisfy the functional dependencies of the original relation.

② Lossy Join decomposition:-

→ When the join of the sub relations does not result in the same relation R that was decomposed. Then, it is known as lossy Join decomposition.

→ It always found some extraneous tuples.

For lossy:- $\boxed{R_1 \bowtie R_2 \bowtie R_3 - - - \bowtie R_n \supset R}$

e.g:

R (A, B, C)

A	B	C
1	2	1
2	5	3
3	3	3

R1 (A, C)

A	C
1	1
2	3
3	3

R2 (B, C)

B	C
2	1
5	3
3	3

$R_1 \bowtie R_2$ =

A	B	C
1	2	1
2	5	3
2	3	3
3	5	3
3	3	3

$R_1 \bowtie R_2 \supset R$

∴ It is lossy.

Determine Wheathere Decomposition is Lossless or Lossy:-

Condition-1: $R_1 \cup R_2 = R$

Condition-2: $R_1 \cap R_2 \neq \phi$ ← null

Condition-3:- $R_1 \cap R_2 =$ superkey of R_1 or R_2

→ If all three condition satisfies, then the decomposition is lossless.

→ If any of condition fail, then the decomposition is lossy.

Ques:- Consider a relⁿ schema R(A, B, C, D) with FD $A \to B$ & $C \to D$. Determine whether the decomposition R into $R_1(A,B)$ & $R_2(C,D)$ is lossless or lossy.

Soln:- **Condition 1:-** $R_1 \cup R_2 = R$

$R_1(A,B) \cup R_2(C,D) = R(A,B,C,D)$

This cond 1 satisfies.

Condⁿ-2:- $R_1 \cap R_2 \neq \phi$

$R_1(A,B) \cap R_2(C,D) = \phi$

~~Cond~~ condⁿ. Fail.

∴ The decomposition is Lossy.

Ques: $R(A, B, C, D)$ with FD: $A \to B$, $B \to C$, $C \to D$, $D \to B$.
Determine the decomposition of R into $R_1(A, B)$, $R_2(B, C)$ & $R_3(B, D)$ is lossless or lossy?

Solⁿ:-

$R(A, B, C, D)$ → $R'(A, B, C)$ and $R_3(B, D)$

$R'(A, B, C)$ → $R1(A, B)$ and $R_2(B, C)$

Condⁿ 1: $R'(A, B, C) \cup R_3(B, D) = R(A, B, C, D)$ True.

Condⁿ 2: $R'(A, B, C) \cap R_3(B, D) = B$ True

Condⁿ 3: $R'(A, B, C) \cap R_3(B, D) = B$

$B^+ = \{B, C, D\}$

B can determine all the attributes of R_3.

∴ it is a superkey of R_3.

∴ The decomposition is lossless.

Now, decomposition of $R'(A, B, C)$ into R1 & R2:-

Cond 1:- $R_1(A, B) \cup R_2(B, C) = R'(A, B, C)$ True.

Cond 2:- $R_1(A, B) \cap R_2(B, C) = B$ True.

Cond 3:- $B^+ = \{B, C, D\}$.

B is a super key for R2

∴ The decomposition is lossless.

Hence, Overall decomposition is Lossless.

Multi-Valued Dependency (MVD) :- ($\rightarrow\rightarrow$)

→ For a dependency $x \rightarrow y$, if for single value of x, multiple value of y exists, then the relation have MVD.

→ It is a dependency where one attribute value is potentially a 'multi-valued fact' about another.

Imp. Points for MVD

→ It means that the relⁿ should have atleast three attributes ($x \rightarrow\rightarrow y$, $x \rightarrow\rightarrow z$).

→ The attributes y & z should be independent for $x \rightarrow\rightarrow y$ (or independent to each other).

x — y — z E.g. Employee Table

Example :-

Person(P)	Mobile(M)	Food like(F)
P_1	M_1 M_2	F1 F2
P_2	M_3	F3

Person(P)	Mobile(M)	Food like(F)	
P_1	M_1	F_1	($P \rightarrow\rightarrow M$)
P_1	M_2	F_2	($P \rightarrow\rightarrow F$)
P_2	M_3	F3	

Scanned with OKEN Scanner

M.V.D :- If $\alpha \twoheadrightarrow \beta$ is exists if in any relation r(R) for all pairs of tuples t_1 and t_2 in r such that $t_1(\alpha) = t_2[\alpha]$, then there exists tuples t_3 & t_4 in r such that

$$\boxed{\begin{array}{l} t_3[\alpha] = t_4[\alpha] = t_1[\alpha] = t_2[\alpha] \\ t_3[\beta] = t_1[\beta] \\ t_4[\beta] = t_2[\beta] \end{array}}$$

*4NF :- A relⁿ 'R' is in 4NF if and only if the foll. conditions are satisfied :-

(i) 'R' is already in 3NF or BCNF.

(ii) if it contains no MVDs.

Ques: Consider the relⁿ Student (name, computer, language).

Name	Computer	Lang.
Aman	Windows/ Apple	English Hindi
Mohan	Linux	English Spanish

Normalize the table. Is this table in 4NF? If no, decompose it to 4NF.

$R_1 \bowtie R_2$

Agent	Company	Product
Aman	C1	PD
Aman	C1	MIC
Aman	C1	speaker
Aman	C2	PD
Aman	C2	MIC
Aman	C2	speaker
Mohan	C1	speaker

$R_1 \bowtie R_2 \supset R$

$\boxed{R_1 \bowtie R_2 \bowtie R_3 \neq R}$

∴ There is no join dependency.

*. **5NF** :- (i) R is already in 4NF.
(ii) It cannot be further non-loss decomposed. (~~Join dependency~~) (No JD).

Inclusion dependence :-

⇒ Inclusion dependency is a statement in which some columns of a relation are contained in other columns.

e.g: Foreign key: In one relation, the referring relation is contained in the primary key column of the referenced relation.

Dependency Preservation:

Join dependency: (JD):-

let 'R' be a relation schema & $R_1, R_2 \cdots R_n$ be the decomposition of R, then R is said to satisfy the join dependency $(R_1, R_2 \cdots R_n)$ if and only if:

$$\boxed{\Pi_{R_1}(R) \bowtie \Pi_{R_2}(R) \bowtie \cdots \bowtie \Pi_{R_n}(R) = R}$$

(order of join doesn't matter)

Or: Ever relation R is equal of join of its projections on $R_1, R_2, \cdots R_n$.

Example:-

Agent	Company	Product
Aman	C1	PD
Aman	C1	MIC
Aman	C2	Speaker
Mohan	C1	Speaker

R1 (Agent, Company), R2 (Agent, Product), R3 (Company, Product)

R1 ~~R1 ⋈ R2 = R~~

Agent	Company
Aman	C1
Aman	C2
~~Aman~~ Mohan	C1

R2

Agent	Product
Aman	PD
Aman	MIC
Aman	speaker
Mohan	Speaker

R3

Company	Product
C1	PD
C1	MIC
C1	Speaker
C2	Speaker

Name	Computer	Lang.
Aman	Windows	English
Aman	Windows	Hindi
Aman	Apple	English
Aman	Apple	Hindi
Mohan	Linux	English
Mohan	linux	Spanish

$$\left.\begin{array}{l} Name \twoheadrightarrow Computer \\ Name \twoheadrightarrow language \end{array}\right] \text{M.V.D}$$

∴ This is not in 4NF.

Now, decompose :-

R1 (name , computer)

R2 (name, language)

R1

name	computer
Aman	Windows
Aman	Apple
Mohan	linux

R2

name	language
Aman	English
Aman	Hindi
Mohan	English
Mohan	Spanish

Only 2 attributes in both R1 & R2. So, no MVD.

$$\left.\begin{array}{l} \text{R1: Key (name, computer)} \\ \text{R2: Key (name, language)} \end{array}\right\} \text{Super key}$$

Ans.