29/8/11

NORMALIZATION:

{Schema Refinement}

⇒ Elimination / minimization of Redundancy.

Redundancy :- Duplicate copy of data stored in multiple location.

Sid	Sname	Cid	Cname	Fee
S1	A	C1	C	5K
S1	A	C2	C	10K
S2	B	C1	C	5K
S3	B	C1	C	5K
S2	B	C3	DB	8K

Redundant data

(Sid, Cid) : Candidate Key

⇒ Few attributes of the relation may suffer from redundancy.

⇒ **Anomalies** : (Problems becoz of redundancy)

- Updation anomaly
- Deletion anomaly
- Insertion anomaly.

29th August 2011

* **Update Anomalies** :- Update all redundant copies of the data, which is too costly.

Eg $C_1 \Rightarrow$ 5K to 8K.

* **Insertion Anomaly** :- Course ⇒ (C4 Java 12K)

Sid	Sname	Cid	Cname	Fee
NULL	NULL	C4	Java	12K
XX	XX	C4	Java	12K

✗ not possible ∵ sid, cid = P.K

So

Dummy data.

⇒ Bcoz of insertion of some data forced to insert — other dummy data.

Dummy data creates incorrect results.

Eg Count student, DB may include dummy data (xx).

* Deletion Anomaly :- Bcoz of deletion of some data, may force to loose other useful data.

Eg Deletion of S_3, may force to loose course info.

PROCESS of Elimination of Redundancy:

① DECOMPOSITION :- Split the relation into two or more sub-relations.

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_1
S_3	B	C_3

(Sid, cid) = P.K

Cid	Cname	fee
C_1	C	5K
C_2	C	7K
C_3	DB	10K

(Cid = P.K)

C4	Java	12K

→ inserted
→ No Insertion anomaly.

∴ All anomalies removed.

But Still

All three anomalies present in (Sid, sname) first table. Hence Further Decomposition

↳ Delete C_3 — Deletion anom.

↳ insert $\boxed{S_4 \mid C}$ — no course the inserting

↳ Updation $\boxed{S_1 \mid A} \rightarrow \boxed{S_1 \mid A1}$

Sid	Sname

Sid = P.K

Sid	cid

(Sid, cid) = P.K

* Redundancy is not because of single attribute.

How To Determine Redundancy:

Cid	Cname
C_1	C
C_1	C
C_1	C
C_1	~~C+~~

$\boxed{\text{Cid determine cname}}$

$\boxed{\text{Cname depend on cid}}$

for one cid = cname always same

Sid	Sname
S_1	A
S_1	A
S_1	A
S_1	~~B~~

$\boxed{\text{Sid determine Sname}}$

$\boxed{\text{Sname depend on Sid}}$

\# $\boxed{X \longrightarrow Y}$

not a candidate key

means x can repeat

i.e whenever x repeats, corresponding to all x, same value is there.

$\Rightarrow$ Then X and Y suffer from redundancy.

PROPERTIES OF DECOMPOSITION:-

① Loss less join Decomposition:

E.g To get the total fee paid by S_1.

(a) In original table, single query.

(b) In splitted tables, two tables are there.

Natural Join b/w sub relations should be equal to original relation.

$\boxed{R = R_1 \bowtie R_2}$ —— loss-less decomposition

$\boxed{R \subset R_1 \bowtie R_2}$ —— * lossy decomposition

↳ extra data : original relation

Lossy Decomposition :- Join b/w sub-relations may lead to a super set of original relation.

② **Dependency Preserving Decomposition :-** Becoz of decomposition, relation should not loose any dependency.

Decomposition is correct only when it is lossless and preserve dependency.

Dependency ? → Redundancy? → Decomposition] Normalization.
check 4 dependency — check for redundancy

DB DESIGN GOALS :

① 0% Redundancy
② Lossless Join Decomposition
③ Dependency preserving decomposition.

Single Valued Functional Dependency (FD) :-

Let R be the relation schema

X, Y be the attribute set over R

t_1, t_2 be any two tuples of R

$\boxed{X \rightarrow Y}$ exists in R

if $t_1.X = t_2.X$ then $t_1.Y = t_2.Y$

$\boxed{\text{means}}$, whereever the X value same, corresponding Y values must also be same, reverse may not be same.

$\boxed{X \to Y \text{ ; x functionaly determine Y}}$

X	Y	Z
x_1	y_1	z_1
x_1	y_1	z_2
x_1	y_1	z_3
x_2	y_1	z_3

$\boxed{X \to Y}$

X	Y
x_1	y_1
x_2	y_1
x_3	y_2

x: cand. key

$\boxed{X \to Y}$

X	Y
x_1	y_1
x_1	y_2

$(X \to Y)$ (invalid) ∴ for same value of x there exist multiple values of y.

Trivial Functional Dependency:

Let R be the Relational schema

X, Y be the attribute sets over R

if $\boxed{X \supseteq Y}$ then $\boxed{X \to Y}$ is trivial → always exist in a relation.

(X=Y) Sid → Sid

(X=Y) Sname → Sname

(X⊃Y) Sid Sname → Sname

(superset)

} Trivial

#

B	C
1	1
2	2
1	1
3	2

B → C = ?

Trival / Non-trivial

$\left.\begin{matrix} B[1,2,3] \\ C[1,2] \end{matrix}\right] X$

Still Non-trival B'coz we dont consider data. B → C are attribute & B is not the superset of C. So Non-trivial

$B \rightarrow B$, $C \rightarrow C$, $BC \rightarrow B$, $AB \rightarrow BC$, $BC \rightarrow C$ — TRIVIAL FD

$B \rightarrow C$ — Non-trivial FD

Non-Trivial Functional Dependency:-

Let R = Relation schema

$(A \rightarrow C)$ — A determining some other attribute.

$X \rightarrow Y$ — is non-trivial over R only if

$\boxed{X \cap Y = \emptyset \ \&\& \ X \rightarrow Y \text{ should satisfy FD definition}}$

$X \cap Y = 0$ → no common attribute in X, Y attribute sets.

$\boxed{X \cap Y = 0}$

B	C
1	1
2	2
1	1
3	2

⇒ $B \cap C = 0$ ✓ — Condition 1

$B \rightarrow C$ =

B	C
1	1
1	1

ok.

$C \rightarrow B$

C	B
1	1
1	1

✓

C	B
2	2
2	3

✗ not ok → Thus $C \rightarrow B$ does not hold.

Condition 2

$X_1 \rightarrow X_1$

Thus $\boxed{B \rightarrow C = \text{Non-trivial}}$

Semi Non-trivial Functional Dependency

combination of Trivial & Non trivial

$\boxed{Sid \rightarrow Sid\ Sname}$

- $Sid \rightarrow Sid$ — Trivial
- $Sid \rightarrow Sname$ — Non trivial

ARMSTRONG AXIOMS

① **Reflexivity** :-

If $\boxed{X \supseteq Y}$ then $\boxed{X \rightarrow Y}$ is trivial which is reflexive dependency (always exist in the relation)

② **Transitive** :-

If $X \rightarrow Y$ & $Y \rightarrow Z$ then $\boxed{X \rightarrow Z}$

③ **Argumentation** :- (Augmentation)

If $X \rightarrow Y$ then $\boxed{XZ \rightarrow YZ}$

E.g. Sid → Sname then Sid Cid → Sname Cid

④ If $\left.\begin{matrix} X \rightarrow Y \\ X \rightarrow Z \end{matrix}\right\}$ Then $X \rightarrow YZ$ ⟶ $X \rightarrow Y$, $X \rightarrow Z$

Example :- if $\left.\begin{matrix} Cid \rightarrow Cname \\ Cid \rightarrow fee \end{matrix}\right\}$ then $\boxed{Cid \rightarrow Cname\,fee}$

⑤ If $X \rightarrow YZ$ Then $\boxed{\begin{matrix} X \rightarrow Y \\ X \rightarrow Z \end{matrix}}$

$\left\{\begin{matrix} YZ \rightarrow X \\ Y \rightarrow X \\ Z \rightarrow X \end{matrix}\right.$ ✗ not valid

A	B	C
1	2	1
1	3	2
2	1	3
2	1	3
2	3	4

AB → C

AB → C ↓ means ab-same, c-same ✓

But A → C ✗, B → C — do not follow

Example :- Identify non-trivial FD's.

① $X \cap Y = \phi$
② $X \to Y$ a F.D

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

Non-trivial

» $X \cap Y = 0$ —— ok for all

» $\boxed{X \to Y \to \text{follow definition of FD's}}$

$A \to B$ ✗ | $B \to C$ ✓ | $C \to B$ ✗

$A \to C$ ✓ | $B \to A$ ✗ | $C \to A$ ✗

$A \to BC$ ✗ | $B \to AC$ ✗ | $C \to BA$ ✗

✗ $A \to B$

✓ $A \to C$

$B \to A$

$B \to C$

$AB \to C$ ✓

$BC \to A$ ✗

$AC \to B$ ✗

Thus, $A \to C$, $B \to C$, $AB \to C$ are valid non-trivial FD's

$\Rightarrow A \to AC$] Semi-trivial

$A \to A \Rightarrow X \to Y =$ ✓ (Trivial)

$A \to C \Rightarrow X \cap Y = 0$ ✓ Non Trivial

$A \to C$

1	1	3
1	1	3
2	2	3
2	2	3
2	2	3

$A \to AC$

Trivial → $\boxed{A \to A}$ – same determine same

Non-trivial → $\boxed{A \to C}$ – other

Semi-trivial → $\boxed{A \to AC}$

All follow the FD condition i.e whenever x is same corresponding Y values must be same.

$$\boxed{t_1.X = t_2.X \text{ then } t_1.Y = t_2.Y}$$

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

$A \to A$
$AB \to A$
$ABC \to AB$
$ABC \to C$
$AB \to AB$
$B \to B$
⋮

Trivial FD's always true

Attribute Closure (X^+): Set of attributes that are functionally determined by X.

$R(A,B,C,D)$ $\{A \to B, B \to C, C \to D\}$

Trivial FD always exist in R: $\{X \to X\}$

$A \to A \to$ always

$(A)^+ = \{A,B,C,D\}$ — Attribute closure set of A. $\boxed{A \to ABCD}$

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

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

$(D)^+ = \{D\}$

Example :- $\{AB \to CD, AF \to D, DE \to F, C \to G, F \to E, G \to A\}$

Which of the following is False?

✓ (a) $\{CF\}^+ = \{ACDEFG\}$ $\quad \{CF\}^+ = \{CF\,G\,E\,A\,D\}$ ($CF \to CF$, $C \to G$, $F \to E$, $G \to A$, $AF \to D$)

✓ (b) $\{BG\}^+ = \{ABCDG\}$ $\quad \{BG\}^+ = \{BGACD\}$

✗ (c) $\{AF\}^+ = \{ACDEFG\}$ $\quad \{AF\}^+ = \{AFDE\}$

✗ (d) $\{AB\}^+ = \{ACDFG\}$ $\quad \{AB\}^+ = \{ABCDG\}$

SUPER KEY :-

Let R be the relational schema with FD set F and X be the set of attributes over R. If X^+ determine all the attributes of R. The X is the Super Key of R.

Let R(ABCDE) and $(AB)^+ = \{ABCDE\}$ Then

AB = superkey of R.

$\boxed{\text{Minimal Superkey is the candidate key.}}$

To determine candidate key???

What to do??

Determine A^+, B^+. (Check whether A or B is superkey?)

Let $A^+ = \{$not determine all attributes of R$\}$

$B^+ = \{$not determine all attributes of R$\}$

Then

single super key = AB

$\boxed{AB = \text{Candidate Key}}$

⇒ $\boxed{\text{Minimal Super key}}$: none of the proper subset of (AB) superkey should also be the superkey.

Let AB = super key

{A}
{B} } Proper subset of AB

{A}: superkey　　{A}: not a super key
{B}: not a "　　{B}: "

⇓　　⇓

A: Candidate key → AB: CK

$(ABC)^+ = \{$all attributes of R$\}$

Super key = ABC

Proper subsets
A　AB
B　BC
　　AC } none of these is a superkey

$(ABC)^+$ ⇓ Candidate Key

Q:- Identify Candidate keys of the Relations

(i) R(ABCDE), {AB→C, C→D, B→E}

Sol^n $(AB)^+ = \{ABCDE\}$
L super key.

Proper subsets of AB = $\{A\} \Rightarrow \{A\}^+ = \{A\}$ — not a superkey
$\{B\} \Rightarrow \{B\}^+ = \{BE\}$ — not a superkey

Thus AB : Candidate key.

Now
Prime attributes = {A, B}
Non-Prime attributes = {C, D, E}

\# $\boxed{\text{If } X \rightarrow \text{Prime attribute}}$ more than one candidate key possible.

Key Point:- Determine one candidate key. Then prime attributes. Then to check whether multiple candidate key exists or not use

(ii) R(ABCDE), {AB→C, C→D, B→EA}

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

Proper Subsets closure
$(C)^+ = \{CD\}$
$(B)^+ = \{BEACD\}$
$(A)^+ = \{A\}$

Since B^+ = subset of AB, also determine all attributes

Thus AB ≠ Candidate key

$\boxed{B = \text{Candidate key}}$

Now $\boxed{X \rightarrow B}$ no where in R. Hence

B is the only candidate key for R.

(iii) R(ABCDE), {AB→C, C→D, D→E, B→A, C→B} –

$(AB)^+ = \{ABCDE\}$: Super key

$(A)^+ = \{A\}$

$(B)^+ = \{BACDE\}$ ⇒ $\boxed{B = \text{Candidate key}}$

How check whether multiple candidate key th??

$X \rightarrow B \equiv C \rightarrow B$

$\{C\}^+ = \{CDEBA\}$ = $\boxed{C = \text{Candidate key}}$

Prime attributes := {B, C}

How, check whether any more candidate key exist.

⇒ $X \rightarrow B$
⇒ $X \rightarrow C$ } → $AB \rightarrow C$

└ But it is not a cand key.

Thus only two candidate key C and B exist.

(iv) R(ABCDE), {A → BCDE, BC → ADE, D→E}

Super key $\{A\}^+ = \{ABCDE\}$ —— $\boxed{A = \text{Candidate key}}$

$\boxed{A = \text{Prime attribute}}$

AB ✗ || BC D ✗

AE ✗

ADC ✓

BC → ADE } BC → A
BC → D
BC → E

BC → ADE

$(BC)^+ = \{BCADE\}$ — Super key

$(B)^+ = \{B\}$
$(C)^+ = \{C\}$ } not determining all attributes

Thus

BC = Candidate

(V) R(ABCD) , {A→B, B→C, C→D, D→A} —

$(A^+) = \{ABCD\}$ ⟹ A = Candidate Key

$(D)^+ = \{DABCD\}$ =

$(C)^+ = \{ABCD\}$ All are $\boxed{\text{Candidate Key}}$ also.

$(D)^+ = \underbrace{\{ABCD\}}_{\text{all attributes thus super key}}$

(VI) R(ABCDEF) , {AB→C, C→D, D→E, E→F, F→B}

$(AB)^+ = \{ABCDEF\}$ — Super Key

$A^+ = \{A\}$
$B^+ = \{B\}$ $\boxed{AB = \text{Candidate Key}}$

F→B

$(A,F) = \{AFBCDE\}$

$\{F\}^+ = \{FB\}$ → not candidate

$\boxed{(AF) = \text{Candidate Key}}$ Prime att = {BAF}

How E→F

$(A,E) = \{AEFBCD\}$

$(E)^+ = \{EFB\}$

$\boxed{(AE) = \text{Candidate Key}}$ Prime att = {A, B, E, F}

How D→E

$(A,D) = \{ADEFBC\}$ ⟹ $\boxed{(AD) = \text{Candidate Key}}$

$(D^+) = \{DEFB\}$

$C \to D$ C = Prime attribute

Now $(A,C)^+ = \{ABCDEF\}$

$C^+ = \{CDEFB\}$ x $\to$ AC = candidate key

Candidate keys : {AB, AC, AD, AE, AF}

*

(vii) R(ABCDE), {AB → C, BC → D}

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

Now E is not determined by any F.D.

Whenever any attribute, not there in Functional Dependency, then that attribute must be the part of Candidate key.

Thus $(ABE)^+ = \{ABCDE\}$ {Adding E both sides}

ABE = Candidate key

(viii) R(ABCDEFGH), {AB → C, AC → B, AD → E, B → D, BC → A, E → G}

⇒ Attributes not the part of functional dependency means FH will be the part of candidate key.

x FH = candidate key

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

$(ABFH)^+ = \{ABCDEFGH\}$ = Super Key

$(A)^+ =$ $\quad (AB)^+ =$ $\quad (BF)^+ =$ $\quad (FH)^+ =$

$(B)^+ =$ $\quad (AF)^+ =$ $\quad (BH)^+ =$

$(F)^+ =$ $\quad (AH)^+ =$

$(H)^+ =$

$(ABFH)$ — $\{ABCDEFGH\}$ — Candidate Key

$AC \to B \;//\; BC \to A$

$(CBFH) = \{ABCDEFGH\}$

agar B ki jagah replace karte to coshi ab jayga.

$AB \to C$

$(ACFH) = \{''\}$

$\{ABFH, ACFH, BCFH\}$ = Candidate key

(19) $R(ABCDEFGHIJ)$, $\{AB \to C, A \to DE, B \to F, F \to GH, D \to IJ\}$

Attributes not part of FD = None.

$(AB)^+ = \{ABCDEFGHIJ\}$ — Super Key

$\left.\begin{array}{l}(A)^+ = \{ADEIJ\} \\ (B)^+ = \{BFGH\}\end{array}\right]\times$

AB = Candidate key

Multiple keys ??

No. bcoz RHS of FD do not contain prime attributes.

AB = Candidate Key

(X) R(ABDLPT) , {B → PT, T → L, A → D}

Attribute not part of FD = {none}

$B^+ = \{BPTL\}$ $(AB)^+ = \{ABPTLD\}$

$T^+ = \{TL\}$ AB = Candidate key

$A^+ = \{AD\}$

(XI) R(ABCDE) , {A → BC, CD → E, B → D, E → A}

$A^+ = \{ABCDE\}$ ⟹ A = candidate key

{A, E, CD, BC}

(XII) R(ABCDE), {AB → C, C → D, D → E, E → A, D → B}

ABE

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

$\boxed{E \to A}$

$(EB)^+ = \{BEACD\}$ EB = Cand key

$\boxed{D \to B}$

$(ED)^+ = \{EDBACD\}$ ED = Cand key

$\boxed{C \to D}$

$(EC)^+ = \{EACDB\}$ EC = Cand key

AB → C

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

$A^+ = \{A\}$, $B^+ = \{B\}$ $\boxed{AB = C.K}$

D → B // E → A

$(AD)^+ = \{ADBCE\}$

$D^+ = \{DEABC\}$ ⟹ $\boxed{D = CK}$

C. Keys

XIII R.C

Sr.

$(AB)^+ = ABCDE$ $A^+ = A$ $B^+ = B$

AB

$AD \neq$

$\cancel{ABE} \rightarrow A$

$(EB)^+ = EABCD$ $E^+ = EA$

BE

$ED \neq$

$D \rightarrow E$

$(DB)^+ = \{ABCDE\}$ $D^+ = \{ABCDE\}$

D

D = candidate key

D — candidate key

$C \rightarrow D$

$(C)^+ = \{ABCDE\}$

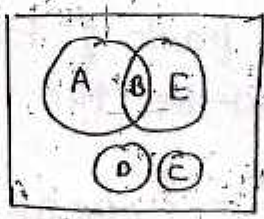
C

AB

$D \rightarrow B$

X (A D) = D = Cand key

$\{AB, BE, D, C\}$



EB

$D \rightarrow B$

D

X (ED) = D = Cand

XIII

(xiii) $R(ABCDEFGH)$ $\{AB \to CD,\ D \to EG,\ E \to H,\ C \to EF,\ H \to A,\ G \to B,\ A \to B\}$

$(AB)^+ = \{ABCDEFGH\}$ $\quad C \to E$, $C \to F$

$A^+ = \{ABCDEFGH\}$

$B^+ = \{B\}$

A = Candidate key

H

F

C

Cand. Key = $\{A, H, F, C\}$

(xiv) $R(ABCDEFG)$, $\{AB \to CDEF,\ C \to ADE,\ D \to EBF,\ E \to DA,\ BE \to AF\}$

G not part of FD. So it must be the part of candidate key.

$(ABG)^+ = \{ABCDEFG\}$

$\quad (AB)^+ = ABCDEF \quad (BG)^+ = BG \qquad (ABG)$

$\quad (AG)^+ = AG \quad A^+ = A,\ B^+ = B,\ G^+ = G$

$(CBG)^+ = \{ABCDEFG\}$

$C^+ = CADBEF$

$GC = ABCDEFG$

$\{ABG, CG, GD, FG, BEG\}$

30th Aug 2011.

Functional Dependency Set Properties:-

① Membership Test

Let F be the FD set and $X \rightarrow Y$ any F.D $X \rightarrow Y$ logically implied in F only if X^+ determines Y.

Example: $F = \{A \rightarrow B, B \rightarrow C\}$

Check whether $A \rightarrow C$, $C \rightarrow B$ member of FD?

Solⁿ (i) $A \rightarrow C$

$A^+ = \{ABC\}$ mem $A \rightarrow C$ ✓ is a member

(ii) $C \rightarrow B$

$C^+ = \{C\}$ → not a member.

② $F = \{AB \rightarrow C, A \rightarrow D, CD \rightarrow E\}$ $AC \rightarrow E = ?$

$(AC)^+ = \{ACDE\}$ ✓ valid

③ $\{A \rightarrow B, B \rightarrow C\}$ $A \rightarrow C = ?$

$A \rightarrow B$ $B \rightarrow C$

$A^+ = ABC$ · $(A \rightarrow C)$ a member of F.D set

④ $\{AB \rightarrow C, BC \rightarrow D\}$ $A \rightarrow C$

$A^+ = \{A\}$ $(A \rightarrow C)$ not a member

→

② Functional Dependency set closure :- (F^+)

Set of all functional dependencies determined by the given FD set.

No. of functional dependencies due to an attribute A is $= 2^m$ where m = no. of elements in closure of A

$A^+ = xyz$ then $2^3 = 8$ — FD possible

Example:

R(ABC)   F: {A→B, B→C}

$(F^+) = \{$ A→A, B→B, C→C, AB→A, AB→B, ⋮ | A→B, AC→BC, AC→B, AC→C, ⋮ | B→C, AB→AC, AB→A, AB→C, ⋮ | A→C, AB→BC, AB→B, AB→C ⋮ $\}$

0 Attributes : $\phi^+ = \phi$ $\phi \to \phi$ 2^0 = 1

1 Attributes : $A^+ = ABC$ ..... $2^3 = 8$ F.D {A→φ, A→A, A→B, A→C, A→BC, A→AC, A→AB, A→ABC} ($2^3 = 8$)

$B^+ = BC$ 2^2 ...... {B→φ, B→B, B→C, B→BC} ($2^2 = 4$)

$C^+ = C$ ..... 2^1 ...... {C→φ, C→C} ($2^1 = 2$)

2 Attributes :- $(AB)^+ = ABC$ {AB→φ ...... AB→ABC} ($2^3 = 8$)

$(BC)^+ = BC$ {BC→φ ..... BC→BC} ($2^2 = 4$)

$(AC)^+ = ABC$ {AC→φ ..... AC→ABC} ($2^3 = 8$)

3 attributes $(ABC)^+ = \{ABC\}$ {ABC→φ ..... ABC→ABC} ($2^3 = 8$)

Total 43 FD possible] 41 Redundant F.D's   43

Q. $F^+ = ?$ R(AB) = {A→B, B→A}

0 — φ→φ

1 → A→B $= 2^1 = 2$ (A→φ, A→B)

B→A $= 2^1 = 2$ (B→φ, B→A)

2 — AB→A $= 2^2 = 4$ (AB→φ, AB→A ......)

φ→φ = 1

$A^+ = AB$ = 4

$B^+ = AB$ = 4

$AB^+ = AB$ = 12

13   11 are redundant

③ Equality between FD sets:

Let F and G are two FD sets.

ⓐ F & G are equal only if

$$\boxed{F^+ = G^+}$$

Dep. G ⇒ mem. of F dependency

$F = \{ - (\ldots) - - - \ldots \}$ $\qquad G^+ = \{ - \ldots (\ldots\ldots) \}$

Dependencies of F must be a memb. of G

ⓑ F and G equal only if.

① F covers G :- All dependencies of G should be logically implied in F.

$F \supseteq G$ is true

⑪ G covers F :- All dependencies of F, should be logically implied in G.

$G \supseteq F$ is true

when

F covers G	True	False	F	True
G covers F	False	True	F	True
	$F \supset G$	$G \supset F$	Uncomparable	$F = G$

Example:- $F = \{ AB \to CD, B \to C, C \to D \}$

$G = \{ AB \to C, AB \to D, C \to D \}$

— Check whether $F = G$???

$\cdot$ $\underline{G}\downarrow$

$\Rightarrow$ ① $\underline{AB \to C}$

$(AB)^+ = \{ABCD\}$ member of F ✓

② $\underline{AB \to D}$ — also a member of F ✓

③ $\underline{C \to D}$ — " " " ✓

Thus $F \supset G$ till here. $\underline{(F \text{ cover } G)}$

$\Rightarrow$ F ⇓

$\underline{AB \to CD}$

$(AB)^+ = \{ABCD\}$ — $\in$ to G ✓

$\underline{B \to C}$

$(B)^+ = \{B\}$ — $B \to C$ $\notin G$ ✗

$\underline{C \to D}$ — $C \to D$ $\in G$ ✓

G does not cover F

Hence $\boxed{F \neq G}$

③ $F = \{A \to C, AC \to D, E \to AD, E \to H\}$

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

① $\underline{F \text{ cover } G = ?}$

$A \to CD \Rightarrow A^+ = \{ACD\}$ ✓

$E \to AH \Rightarrow E^+ = \{EHAD\}$ ✓

$\underline{F \text{ cover } G}$ ✓

② $\underline{G \text{ cover } F = ?}$

$A \to C \Rightarrow A^+ = \{ACD\}$ ✓

$AC \to D \Rightarrow$ ✓

$E \to AD \implies E^+ = \{EHADC\}$ ✓

$E \to H \implies$ ✓

F covers G } True means $\boxed{F = G}$
G covers F }

PROPERTIES OF DECOMPOSITION :-

① Lossless Join Decomposition :- Let R be the relational schema with instance r decomposed into $R_1, R_2, \ldots R_N$ with instances $r_1, r_2, r_3 \ldots\ldots r_n$.

In General $r_1 \bowtie r_2 \bowtie \ldots\ldots \bowtie r_n \supseteq r$

» If $\boxed{r_1 \bowtie r_2 \bowtie r_3 \ldots\ldots\ldots \bowtie r_n = r}$

Then decomposition is Lossless Decomposition

» If $\boxed{r_1 \bowtie r_2 \bowtie \ldots\ldots \bowtie r_n \supset r}$

Then decomposition is Lossy Decomposition

Example: Illustration:- Let R(ABC) is decomposed into $R_1(AB)$, $R_2(BC)$.

R

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

r

A	B
1	2
2	2
3	1

$r_1 = \pi_{AB}(r)$

A	C
2	1
2	2
1	2

$r_2 = \pi_{BC}(r)$

π : Projection
↓
used to project column of Relation.

X — Cross Product.

R X S :- results all attributes of R followed by all attributes of S

⇒ Relational Algebra operations

Π : Projection : used to project columns of a Relation.

Eg $\Pi(r)$ AB from R(ABC)

$\Pi(r)$ BC from R(ABCD)

X : CROSS Product

R×S : results all attributes of R followed by all attributes of S, with all Combinations of tuples from r and s.

⇒ $r_1 \times r_2$ =

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

R_1

A	B
1	2
2	2
3	1

R_2

B	C
2	1
2	2
1	2

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

A X BC

↳ P.K. of r_1

σ : Selection :

$\sigma_p(r)$: used to select tuples from r those are satisfied condition 'p'.

⋈ : Natural Join : R⋈S : Cross product followed by Selection of the tuples that are equality b/w the common attributes and then projection of distinct column.

$r_1 \bowtie r_2 =$ (i) Cross Product
(ii) Selection
(iii) Projection

⇒ Cross Product ✓

⇒ Selection

Common attribute = B

$$\boxed{\sigma_{R_1.B = R_2.B}(r_1 \times r_2)}$$

⇒ Projection

$$\pi_{ABC}\left(\sigma_{R1.B = R2.B}(r_1 \times r_2)\right)$$

A	B	B	C
1	2	2	1
1	2	2	2
1	2×	1×	2
2	2	2	1
2	2	2	2
2	2×	1×	2
3	1×	2×	1
3	1×	2×	2
3	1	1	2

$\boxed{r_1 \bowtie r_2}$

O/P

$r_1 \bowtie r_2 =$

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

Q:- $R_1(ABCD)$ $R_2(CDE)$

$$\boxed{R_1 \bowtie R_2 = \pi_{ABCDE}\left(\sigma_{\substack{R_1.C = R_2.C \\ R_1.D = R_2.D}}(R_1 \cancel{\bowtie} R_2)\right)}$$ ✓

(2) $R_1(ABC)$ $R_2(DE)$ ⟶ no attribute common then

$$\boxed{R_1 \bowtie R_2 = R_1 \times R_2}$$

R(ABC)

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

$R_1(AB)$

A	B
1	2
2	2
3	1

$R_2(BC)$

B	C
2	1
2	2
1	2

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

$r_1 \bowtie r_2$

Extra tuples becoz of Lossy decomposition

$\Downarrow$

$r \subset r_1 \bowtie r_2$

Lossy Join Decomposition

» Getting extra tuples bcoz tables combined based on B which is not primary key for any of the table i.e R_1 or R_2. B itself has duplicate values.

⇒ If $R_1(AB)$ & $R_2(AC)$ then

R_1

A	B
1	2
2	2
3	1

R_2

A	C
1	1
2	2
3	2

$R_1 \bowtie R_2$

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

⇒ Lossless Join Decomposition.

$R_1 \bowtie R_2 = r$ ✓

» Table combined based on common attribute A, which is not having duplicate value. Hence no extra tuples.

Conclusion:- If the common attribute b/w Relations $(R_1 \cap R_2)$, is not the super key of either R_1 or R_2 then Decomposition is <u>Lossy</u>

If $R_1 \cap R_2$ = Common attribute is the superkey of either R_1, R_2 or both, Then decomposition is <u>LOSSLESS</u>.

Exam:
Eg R(ABC) F = {A→B, A→C}, decomposed into $R_1(AB)$, $R_2(BC)$. Check decomposition is lossy or lossless.

Lossy

⇒ We know that
if Common attribute = Superkey then
(I) decomposition — lossless else lossy.

Now Common attribute = B
Check whether B = superkey
↳ if it derive all attributes.

$B^+ = \{B\}$

i.e B (common attribute) not a superkey of either relation. Thus Lossy decomposition.

⇒ R(ABCD), F {A→B, B→C, C→D}
(II) (i) Demposed into $R_1(ABC)$, $R_2(CD)$ ** Imp

$R_1 \cap R_2 = \{C\}$
$\{C\}^+ = \{CD\}$ → deriving all attributes of R_2
i.e C = Superkey of R_2

(III) Hence Lossless Decomposition.

(ii) $R_1(AB)$ & $R_2(CD)$
lossy ∵ no attribute common, no superkey.

* Imp
(iii) Decomposed into $R_1(AB)$ $R_2(BC)$
(IV) lossy ∵ one attribute (D) not there in any subrelation R_1 or R_2. Thus $R_1 \bowtie R_2$ is not equal to R.

(4) Decomposed into

$R_1(AB)$ $R_2(BC)$ $R_3(CD)$

Lossy or lossless = ?

⇒ Three decomposed relations.
Then take two-two relations, all should satisfy the condition.

PROCEDURE (To check Lossy/Lossless Decomposition)

Let R be the Relational schema with FD set F decomposed into $R_1, R_2, R_3, \ldots R_n$. Given decomposition is lossless, only if

$\boxed{(1)\ R_1 \cup R_2 \cup \ldots \cup R_n = R}$ → Koi bhi attribute miss nai hona chaiye / All attributes should be covered in one of the other subrelation

(all attributes of R should belong to some subrelation)

nd [2] (I) R_i and R_j be any two relations, can be merged into single relation $R_{ij} = R_i \cup R_j$ only if

[a] $R_i \cap R_j \neq \phi$ // attributes common

[b] $R_i \cap R_j \rightarrow R_i$ // Common attribute, superkey of R_i

or

$R_i \cap R_j \rightarrow R_j$ // common attribute, superkey of R_j

(II) Repeat (I) until N relations become single relation.

→ If N relations become single relation: (lossless join decom)

else (lossy join decomposition).

Q:- R(ABCDEG)

$F = \{AB \rightarrow C, AC \rightarrow B, AD \rightarrow E, B \rightarrow D, BC \rightarrow A, E \rightarrow G\}$

① $D_1 = \{AB, BC, ABDE, EG\}$ — Decomposed set 1

② $D_2 = \{ABC, ACDE, ADG\}$ — Decomposed set 2

① $R_1(AB)$ $R_2(BC)$

Common attribute B

$\{B^+\} = \{BD\}$ — not a superkey of R_1 or R_2

– $R_2(BC)$ $R_3(ABDE)$

Common attribute = B — not superkey of R_2 or R_3.

– $R_1(AB)$ $R_3(ABDE)$ $R_1 \cap R_3 = AB$

$(AB)^+ = \{ABCDEG\} \Rightarrow AB =$ superkey for both $R_1(AB)$, $R_3(ABDE)$

Thus R_1 and R_3 can be joined into single relation

$R_{13}(ABDE)$

– $R_2(BC)$ $R_4(EG)$ — no common attribute

– $R_{13}(ABDE)$ $R_4(EG)$ — $R_{13} \cap R_4 = E$

– $\{E\}^+ = \{EG\} \rightarrow$ E superkey of R_4

Hence R_{13} and R_4 can be joined.

– Now $R_{134}(ABDEG)$ $R_2(BC)$ $R_{134} \cap R_2 = B$

$B \Rightarrow$ not superkey

Can't join them. Hence Lossy decomposition.

$R_1(AB)$ $R_2(BC)$ $R_3(ADBE)$ $R_4(EG)$

$R_{13}(ABDE)$ $R_2(BC)$ $R_4(EG)$

$\underbrace{R_{134}(ABDEG) \quad R_2(BC)}$

Can't be Joined

∵ B not the super key of either relation

LOSSY DECOMPOSITION.

(ii) $R_1(ABC)$ $R_2(ACDE)$ $R_3(ADG)$

$(AC)^+ = (ACBDEG)$ ↳ super key of both R_1 & R_2

$R_{12}(ABCDE)$ $R_3(ADG)$

$(AD)^+ = (ADEG)$ superkey of R_3

$\underbrace{R_{123}(ABCDEG)}$

Lossless

Q. $R(ABCD)$, $F = \{A \to B, B \to C, C \to D\}$

Decomposed into $\{AB, BC, CD\}$ Lossless/Lossy = ?

$R_1(AB)$ $R_2(BC)$ $R_3(CD)$

B superkey of R_2 $B^+ = \{BCD\}$

$R_{12}(ABC)$ $R_3(CD)$

C superkey of R_3 $C^+ = \{CD\}$

II Dependency preserving decomposition :-

Relational Schema
with FD set (F)

Relation schema R1 with functional dependency set F1

R1 with F1 R2 with F2

$\Rightarrow$ a) $F_1 \cup F_2 \supset F$ ✗ ------ always false

b) $F_1 \cup F_2 = F$: Dependency Preserving

c) $F_1 \cup F_2 \subset F$: Not preserving dependency

d) None

$$\boxed{\begin{array}{l} F_1 \cup F_2 \subseteq F \\ r_1 \bowtie r_2 \supseteq r \end{array}}$$

$F_1 \cup F_2 \subseteq F$ — Dependency Preserved

$r_1 \bowtie r_2 \supseteq r$ — lossless decomposition

Let R be the Relational Schema with FD set 'F' decomposed into $R_1, R_2, R_3, \ldots\ldots R_N$ with FD sets $F_1, F_2, F_3, \ldots\ldots F_n$ respectively.

In general, $F_1 \cup F_2 \cup F_3 \ldots\ldots\ldots \cup F_n \subseteq F$

if $F_1 \cup F_2 \cup \ldots F_n \subset F$
(Not preserving dependency)

if $\boxed{F_1 \cup F_2 \cup \ldots\ldots F_n = F}$
(Dependency preserved)

$\{F_1 \cup F_2, \ldots F_n\}$ dependencies all covered by F
if F dependency covered by $F_1 \cup F_2 \ldots F_n$ $\rightarrow$ Depend. Preserved

$\{F_1 \cup F_2 \cup \ldots\ldots F_n\}$ dependencies always covered by F

If F dependencies covered by $\{F_1 \cup F_2 \cup \ldots F_n\}$

Then Dependency preserved

Otherwise Dependency not preserved.

∵ these are derived from F only

Q3. R(ABCD)

$F = \{ A \to B; B \to C, C \to D, D \to A \}$

$D = \{ AB, BC, CD \}$

Dependency preserved or not = ??

Sol^n **Step 1** Identify non-trivial functional dependency in the subrelation.

I

Transitively we determine dependencies

AB	BC	CD
$A \to B$	$B \to C$	$C \to D$
$B \to A$	$C \to B$	$D \to C$

Non-trivial

$A \to A$ Trivial
$B \to B$

$A^+ = ABCD$ $\quad B^+ = ABCD$

$B^+ = ABCD$ $\quad C^+ = ABCD$

$F_1 = \{ A \to B, B \to A \}$

$F_2 = \{ B \to C, C \to B \}$

$F_3 = \{ C \to D, D \to A \}$

$F_1 \cup F_2 \cup F_3$

Step 2: Check wheth $F = \{ A \to B, B \to C, C \to D, D \to A \}$ member of subrelation or not

II

$A \to B$ ✓ $\quad C \to D$ ✓

$B \to C$ ✓ $\quad D \to A$. — $D^+ = \{DABC\}$ ✓

→ All dependencies of original relation preserved. Thus it is a dependency preserving relation. we can see that

$[F_1 \cup F_2 \cup F_3] \Rightarrow F$ ⟶ Hence dependency preserved.

Q In given decomposition, dependency preserved or not.

⇒ R(ABCDEF)

F: {AB → CD, C → D, D → E, E → F}

D = {AB, CDE, EF}.

∵ F is not there in relation

⇒

R_1 AB	R_2 CDE	R_3 EF
	C→D, C→E, ~~C→F~~ D→E, ~~D→F~~ ~~E→F~~ CD→E * CE→D	E→F

① Determine non-trivial dependencies

AB $A^+ = \{A\}$ A→A
$B^+ = \{B\}$ B→B] Trivial i.e non-trivial dependency not there in Relation R_1.

CDE
$C^+ = \{CDEF\}$ — [C→D, C→E, C→F] Non-trivial
$D^+ = \{DEF\}$ — [D→E, D→F] Non-trivial
$E^+ = \{EF\}$ — E→F] N.Trivial

* $(CD)^+ = \{CDEF\}$
$(CE)^+ = \{CDEF\}$
$(DE)^+ = \{DEF\}$

EF $E^+ = \{EF\}$ — E→F
$F^+ = \{F\}$ — F→F (Trivial)

② Check F dependency preserved in subrelations or not

AB → CD —— or [AB → C, AB → D] × not preserved in sub relations

$(AB)^+$ in subreltn $= \{AB\}$

C → D ✓
D → E ✓
E → F ✓

⇓

Thus dependency not preserved.

Q. $R(ABCDE)$

$F = \{A \to BC,\ CD \to E,\ B \to D,\ E \to A\}$

$D = R_1(ABC) \quad R_2(ADE)$

	ABC	ADE	
$A \to BC$ $AB \to C$ $BC \to A$ $AC \to B$	$A \to BCD$ $B \to D$ $AB \to CDE$ $AC \to B$ $ABC \to DE$	$E \to ABC$ $AD \to BCE$ $AE \to BC$ $DE \to ABC$ $ABC \to DE$	$A \to DE$ $E \to AD$ $AD \to E$ $AE \to D$ $DE \to A$

(1) Identify non-trivial dep.

ABC

$A^+ = (ABCD)$ $\quad (AB)^+ = (ABDCE)$ $\quad (ABC)^+ = (ABCDE)$

$B^+ = (BD)$ $\quad (AC)^+ = (ABC)$

$C^+ = (C)$ $\quad (BC)^+ = \cancel{(BC)}\ [BCDEA]$

ADE

$D^+ = \{D\}$ $\quad (AD)^+ = \{ABCDE\}$ $\quad (ADE)^+ = \{ADEBC\}$

$E^+ = \{EABCD\}$ $\quad (AE)^+ = \{ABCE\}$

$(DE)^+ = \{ADEBC\}$

$\left.\begin{array}{l} CD \to E \times \\ B \to D \times \end{array}\right\}$ not preserved.

Q R(ABCDEG)

$F = \{ AB \to C,\ AC \to B,\ AD \to E,\ B \to D,\ BC \to A,\ E \to G \}$

$D = \{ ABC, ACDE, ADG \}$

ABC	ACDE	ADG
$AB \to C$ $AC \to B$ $BC \to A$ $\cancel{B \to D}$	$AC \to DE$ $AD \to E$ $ACD \to E$ $ACE \to D$	$AD \to G$

D not there in R

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

$AC^+ = ABCDEG$
$AD^+ = ADEG$
$AE^+ = AEG$
$CD^+ = CD$
$CE^+ = CEG$
$DE^+ = DEG$

$^4C_2 = 6$ combination

$ACD^+ = ABCDEG$
$ADE^+ = ADEG$
$CDE^+ = CDEG$
$ACE^+ = ABCDEG$

$^4C_3 = 4$ combi.

Belong or not ??

$\Downarrow$

B^+ consider these dependencies

$B^+ = \{B\}$ not preserved

Now

$AB \to C$ ✓
$AC \to B$ ✓
$AD \to E$ ✓
$B \to D$ —
$BC \to A$ ✓
$E \to G$

$E^+ = \{E\}$ not preserved.

∴ Dependency not preserved.

Q:- $R(ABCDEFGHIJ)$

$F: \{AB \rightarrow C,\ A \rightarrow DE,\ B \rightarrow F,\ F \rightarrow GH,\ D \rightarrow IJ\}$

1. $D_1: \{ABC, ADE, BF, FGH, DIJ\}$
2. $D_2: \{ABCDE, BFGH, DIJ\}$
3. $D_3: \{ABCD, DE, BF, FGH, DIJ\}$

(1)

ABC	ADE	BF	FGH	DIJ
~~ABD~~	$AD \rightarrow E$	$B \rightarrow F$	$FG \rightarrow H$	~~DI~~
$AB \rightarrow C$	$AE \rightarrow D$		$FH \rightarrow G$	~~D→IJ~~
	$A \rightarrow DE$		$F \rightarrow GH$	$DI \rightarrow J$
				$DJ \rightarrow I$
				$D \rightarrow IJ$

$A^+ = ADEIJ$ $D^+ = DIJ$ $F^+ = FGH$ ~~$D^+ = DIJG$~~

$B^+ = BFGH$ $E^+ = E$ $G^+ = G$ ~~$J^+ = IG$~~

$C^+ = C$ $AD^+ = ADEIJ$ $H^+ = H$ $DI^+ = DIJ$

$AB^+ = ABCDEFGHIJ$ $AE^+ = ADEIJ$ $FG^+ = FGH$ $I^+ = I$

$BC^+ = BCFGH$ $DE^+ = DEIJ$ $GH^+ = GH$ $DJ^+ = [DIJ]$

$AC^+ = ACDEIJ$ $FH^+ = FGH$ $IJ^+ = [IJ]$

F: $AB \rightarrow C$ —— Preserved

$A \rightarrow DE \Rightarrow\ =?_o\ A^+$ in table's $F = A^+ = [ADE]$ — Preserved

$B \rightarrow F \Rightarrow$ Preserved

$F \rightarrow GH \Rightarrow$ Preserved

$D \rightarrow IJ \Rightarrow$ Preserved

Hence in $D_1 \Rightarrow$ Dependency Preserved.

D_2 =

ABCDE	BFGH	DIJ
A → DE AB → CDE AC → DE	B → FGH F → GH BF → GH BG → FH BH → FG	D → IJ DI → J DJ → I

$CD^+ = [CDIJ]$
$CE^+ = [CE]$
$DE^+ = [DEIJ]$

$BG^+ = \{BFGH\}$
$BH^+ = \{BFGH\}$
$BF^+ = \{BFGH\}$

$DI^+ = DIJ$
$DJ^+ = DIJ$
$IJ^+ = IJ$

$ABC^+ = ABCDEFGHIJ$
$ACD^+ = ACDEIJ$
$ABD^+ =$

No need to calculate all these 3 variable, 4 variable fd's Bcoz there exist no such dependency in F.
So no need to check for these.

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

Hence dependency preserved.

D_3

ABCD	DE	BF	FGH	DIJ
A → D AB → CD AC → D		B → F	F → GH FG → H FH → G	D → IJ DI → J DJ → I

$CD^+ = CDIJ$

Now $F =$
AB → C — Preserved
$\boxed{\text{A → DE — Not Preserved}}$
B → F — Preserved
F → GH — Preserved
D → IJ — Preserved

Hence Dependency not preserved.

Minimal Cover or Canonical Cover :-

Goal :- Elimination of redundant FD's from functional dependency set (FD set).

→ Redundant FD means, RFD is the dependency such that deletion of that not effecting expressive power of FD set.

Eg. F : { $A \to B$, $B \to C$, $A \to C$ }

G = { $A \to B$, $B \to C$ } ($A \to C$ deleted)

Hence $\boxed{F \equiv G}$ # { F covers G : ✓ ; G covers F : ✓ } Hence $\boxed{F \equiv G}$

Extraneous Attribute :- (Removal create no plbm)

As E.T. ① F : { $AB \to C$, $A \to C$ }, Extraneous attribute = ?

Let A be extraneous attribute.

G : { $B \to C$, $A \to C$ }

Now if $F \equiv G$, then A = extraneous

To determine ? $F \equiv G$? — F covers G, G covers F

F cover G :-

$B \to C$ in F

check B^+ in F $B^+ = \{B\}$

& not C so

Hence $F \not\equiv G$.

(2) B = Extraneous element (Let)

$F: \{A \to C\}$
$\{AB \to C\}$

$G = \{A \to C\}$

(i) $F \equiv G = ?$

F cover G: $A \to C \in F$ ✓

G cover F: $A \to C \in G$ ✓

$\boxed{AB \to C \in G} = ?$

$\boxed{(AB)^+ \text{ in } G = (ABC)}$ means $AB \to C$. ✓

Thus $F \equiv G$. Thus B = extraneous element.

$A \to C$
$AB \to C$
Only A is sufficient enough to determine C. then why B → it is Extraneous

कुछ नहीं Baabu...?? OK OK aa gaya
$(AB)^+ = (ABC)$
Yes Yes I got it
पुरानी ही आ गया आज

$x \to w$
$xyz \to w$] $\boxed{Extr = yz} \Rightarrow \boxed{x \to w}$

w determined by x only, no need of y and z.

CASE 2 :- $\{AB \to C, A \to B\}$ Extraneous elem = ?

$F: \{AB \to C, A \to B\}$

(1) A (Let) then $G: \{B \to C, A \to B\}$

(i) F cover G :- X $A \to B$ ✓
$B \to C$ $(B^+ \cancel{\text{in } F} = \{B\})$ X

⇒ Let B = Extraneous element

$F = \{AB \to C, A \to B\}$

$G = \{A \to C, A \to B\}$

① F cover G

$A \to C$ ...... A^+ in F = $(ABC) \Rightarrow A \to C$ ✓

$A \to B$ ✓

(ii) G cover F

$A \to B$ ✓

$AB \to C$ — means $(AB)^+$ in G

$(AB)^+ = \{ABC\}$ $AB \to C$ ✓

Hence

$F \equiv G$

Thus B is the extraneous attribute.

$AB \to C$ $A \to B$

Sath sath

Sath Sath

↳ B not required.

$xyz \to \omega$

$y \to z$ — sath-2 so not req.

$x \to y$

yz = extr. attribute

$xyz \to \omega$

$\boxed{y \to z}$

↳ then

Z is extraneous

⇒

$xy \to \omega$

$\boxed{x \to y}$

then y is extraneous

Thus

$\boxed{x \to \omega}$

Transitivity

Case 3 :- $\{A \to BC, B \to C\}$: $\{A \to B, B \to C\}$ — minimal cover

C = extraneous attribute ∵ we can get it transitively.

Pseudo transitivity

Case 4: $\{AB \to CD \quad BC \to D\}$

From AB we get C

from BC we get D

$\boxed{\text{Pseudo Transitive Property}}$

So no need of D in $AB \to C$

$AB \to D$ X

D = Extraneous.

Extraneous Elements / attributes :

Cases	Minimal Cover	Extraneous Element
(1) $\{AB \to C, A \to C\}$	$\{A \to C\}$	B
(2) $\{AB \to C, A \to B\}$.	$\{A \to C, A \to B\}$	
(3) $\{A \to BC, B \to C\}$	$\{A \to B, B \to C\}$	
(4) $\{AB \to CD, BC \to D\}$	$\{AB \to C, BC \to D\}$	

Q $\{AB \to C \quad A \to B \quad B \to A\}$ minimal cover Find.

⇒ Consider $(AB \to C \quad A \to B)$

A is extraneous

$M.C = \{A \to B, B \to C, B \to A\}$

– Consider $(\cancel{A}B \to C \quad B \to A)$

A is extraneous attribute

m.c : $\{(B \to A)\ (\cancel{A \to C})\ (\cancel{A \to B})\}$

$[(B \to A)\ (B \to C)\ (A \to B)]$

$\{AB \to C,\ A \to B,\ B \to A\}$

$\{A \to C, A \to B, B \to A\}$

$\{A \to BC, B \to A\}$

MC_1

$\{B \to C, A \to B, B \to A\}$

$\{B \to AC, A \to B\}$

MC_2

Minimal Cover.

$\Rightarrow$ Minimal Cover not unique.

But two minimal cover are logically equivalent.

31^{st} Aug. 2011

Q: Determine the minimal cover.

$\{ABCD \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$

~~AC → D~~ ~~ABCD → E~~

$A\cancel{B}CD \rightarrow E \quad A \rightarrow B$

$\boxed{ACD \rightarrow E}$

$AC\cancel{D} \rightarrow E \quad AC \rightarrow D$

$\boxed{AC \rightarrow E} \quad AC \rightarrow D$

$\boxed{AC \rightarrow DE}$ #

$\{AC \rightarrow DE, E \rightarrow D, A \rightarrow B\}$

$AC \rightarrow \cancel{D}E \quad E \rightarrow D$

$\underline{AC \rightarrow E} \quad \underline{E \rightarrow D}$

$\{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$ # minimal cover.

Q:- $\{ABC \rightarrow CD, BC \rightarrow D, A \rightarrow B, C \rightarrow D\}$

$ABC \rightarrow CD \quad BC \rightarrow D$

$\overline{ABC} \rightarrow C$

$\cancel{A}BC \rightarrow D \quad BC \rightarrow D$

$\{BC \rightarrow D\}$

$\cancel{B}C \rightarrow D \quad C \rightarrow D$

$\{BC \rightarrow D, A \rightarrow B, C \rightarrow D\}$

$\{A \rightarrow B, C \rightarrow D\}$

$\{AC \rightarrow \cancel{CD},\ BC \rightarrow D,\ A \rightarrow B,\ C \rightarrow D$

$AC \not\rightarrow C,\ BC \rightarrow D,\ A \rightarrow B,\ C \rightarrow D$

$BC \rightarrow D \quad A \rightarrow B,\ C \rightarrow D$

$\{A \rightarrow B,\ C \rightarrow D\}$ m.c

Q $\{A \rightarrow BC,\ B \rightarrow AC,\ C \rightarrow AB\}$

$A \rightarrow BC \quad B \rightarrow A \quad C \rightarrow A$

$B \rightarrow C \quad C \rightarrow B$

$A \rightarrow B$

$A \rightarrow C$

$\{A \rightarrow BC \quad B \rightarrow AC \quad C \rightarrow A \quad C \rightarrow B\}$ Let

Ans $\{A \rightarrow B,\ A \rightarrow C\ \ B \rightarrow AC,\ C \rightarrow AB\}$

$\{A \rightarrow B\ \ A \rightarrow C\ \ B \rightarrow A\cancel{C}\ \ C \rightarrow A\} \rightarrow$ ~~$A \rightarrow BC\ \ B \rightarrow AC\ \ C \rightarrow A$~~ B

$\rightarrow A \rightarrow B\cancel{C},\ B \rightarrow C,\ C \rightarrow A$

$\{A \rightarrow B,\ A \rightarrow C\ \ B \rightarrow A\ \ C \rightarrow A\}$

$[A \rightarrow B\ \ B \rightarrow C\ \ C \rightarrow A]$

$[A \rightarrow BC,\ B \rightarrow A,\ C \rightarrow A]$

2: $\{A \to BC,\ CD \to E,\ E \to C,\ D \to EAH,\ ABH \to BD,\ DH \to BC\}$

$\cancel{C}D \to E$ $\qquad$ $D \to E$, $D \to A$, $D \to H$

$[A \to BC,\ E \to C,\ D \to EAH,\ ABH \to BD,\ DH \to BC]$

$\{A \to B,\ A \to C,\ E \to C,\ D \to EAH,\ A\cancel{B}H \to BD,\ DH \to BC\}$

$\{A \to BC\quad E \to C\quad D \to EAH\quad AH \to BD\quad D\cancel{H} \to BC\}$

$D \to EA$, $D \to H$

$\{A \to BC\quad E \to C\quad D \to EAH\quad AH \to \cancel{BD}\quad D \to B,\ D \to C\}$

$\{A \to BC\quad E \to C\quad D \to EAH\quad AH \to D\quad D \to BC\}$

$\cancel{D \to E}$, $\cancel{D \to AH}$, $\cancel{AH \to D}$

$\{A \to BC\quad E \to C\quad D \to AB\cancel{C}EH\quad AH \to D\}$

$\{A \to BC\quad E \to C\quad D \to ABEH\quad AH \to D\}$

$\{A \to B,\ A \to C\quad E \to C\quad D \to A\cancel{B}EH\quad AH \to D\}$

$\{A \to BC\quad E \to C\quad D \to AEH\quad AH \to D\}$ — minimal cover.

NORMAL FORMS :

- Set of rules to identify the redundancy.
- Used to reduce/eliminate redundancy/anomalies #
- Various normal forms:

  - 1NF
  - 2NF
  - 3NF
  - BCNF
  - 4NF

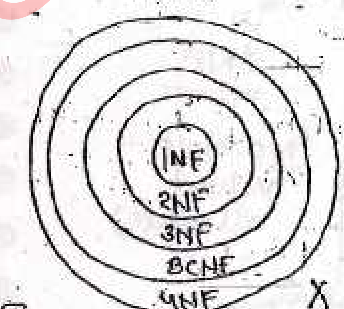
1NF, 2NF, 3NF, BCNF } Single valued f^n dependencies ($X \to Y$)

BCNF →
- Highest NF becoz of single valued FD.
- 0% Redundancy " " " ".
- may suffer from redundancy becoz of multi valued functional dependencies (MVD).

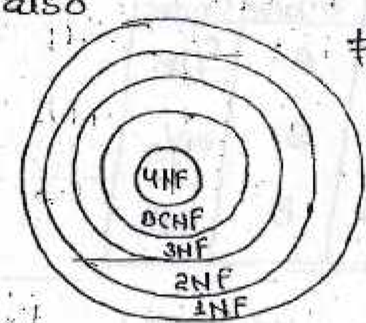
4NF →
- Multivalued FD (MVD) ($X \to\to Y$)
- used to eliminate redundancy due to MVD.

⇒ Every Higher Normal form is also lower normal form

1NF
↑
2NF —— 2NF is 1NF also
↑
3NF
↑
BCNF —— BCNF is 3NF, 2NF, 1NF also
↑
4NF —— 4NF is BCNF also



Every 2NF is not 1NF X



Every 2NF is 1NF.

Expressive Power = No. of relations satisfied by the NF.

FIRST NORMAL FORM:-

» Relation R is in 1NF only if (atomic)

R does not contain any multivalued attributes.

(OR)

All attributes of R are atomic (single valued).

Sid	Sname	Cid/Cname
S1	A	C/DB
S2	B	DB/C++
S3	B	DB

Here, Multivalued attribute

Sid & Sname = Atomic

Cname = Multivalued

Primary key = Sid

- Single valued dependency

$\boxed{Sid \rightarrow Sname}$ ∵ Sid (Key) always determine other attributes except multivalued attributes.

Sid → Sname means
S1 → A
S2 → B
S3 → B

always S1 associated with A
" S2 " B
" S3 " B

- Conversion of

Multivalued attribute into single value attribute

Sid	Sname	Cname
S1	A	C/DB
S2	B	DB/C++
S3	B	DB

Primary Key : Sid

Sid	Sname	Cname
S1	A	C
S1	A	DB
S2	B	DB
S2	B	C++
S3	B	DB

How it is in 1NF

(No multi-valued attribute)

NOTE

R(ABCD)   (multi-valued)

F: {A → B, B → C}

D not part of F.D. means . D is a part of candidate key. (D is not single valued / it is multivaled attribute)

$A^+ = \{ABC\}$

$(AD)^+ = \{ABCD\}$ — Augmentation Property

$\boxed{\text{Candidate Key} = AD}$

→ Adding D to functional dependency, is done to remove the multi-valued D. (duplication as in previous table).

⇒ To remove the multivalued attribute, (D) it is made part of candidate key.

⇒ 1NF ensured using Candidate Key.

R(ABCD), {A → B}

Candidate Key = (ACD)

R(ABC), {A → C}

Candidate key = (AB)

Problem with 1NF

⇒ High Redundancy due to elimination of multivalued attributes.

NOTE :- If Candidate Key determine all attributes of the relation R. Then the relation R is surely in 1NF.

$X \rightarrow Y$ (Let) : Non-trivial F.D in R

not SuperKey ⟹ Redundancy exists b/w (X,Y) attributes sets.

means x value can be duplicated.

X	Y
x_1	a
x_1	a
x_1	a

also $X \rightarrow Y$ means for same value of x y is same ≡ Redundancy

$X \rightarrow Y$ do not suffer from redundancy when X = superkey

#

Sid	Sname	Cname
S_1	A	C
S_2	B	DB
S_3	B	C++

Sid → Sname (Here)

not a Superkey —— Hence redundancy possible in R. (table).

$X \longrightarrow Y$ (determiner)

(determinent)

Determinent ⟶ Determiner

Possibilities of Redundancies Bcoz of FD's :-

- Prime Attributes (Key attributes)
- Non-prime attributes (Non-key attributes)

① Proper subset of Candidate Key → Non-Key attribute → Eliminated by 2NF

② Non-Key attribute → Non-Key attribute → Eliminated by 3NF

$R(ABCD) : \{AB \rightarrow C,\ C \rightarrow D\}$

$AB^+ = ABCD$

$AB =$ candidate key

$C \rightarrow D \Rightarrow$ Redundancy

non-Key non Key

③ Proper Subset of one cand. key → Proper subset of other cand. key → Eliminated by BCNF

\# (Non-key) $\longrightarrow$ (Proper Subset of Key) ✗ not possible case

↑non-prime ↑Prime attribute

R(ABCDE) $\{AB \to C,\ C \to D,\ B \to E,\ C \to A\}$

{Prime attribute} (non key)

Cand key = AB

Since $C \to A$ (A determined by C)

AB : Cand. Key

CB ⇒ Cand. Key

{we can substitute (C) non prime in place of prime(A) The C = prime. Thus, such dependency not possible.}

no. P.D.

SECOND NORMAL FORM :

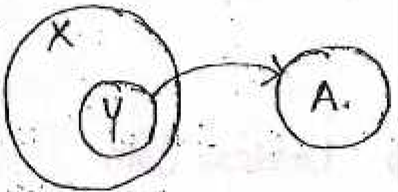
Let R be the relational schema, is in 2NF if and only if

[1] R should be in 1NF
{Every candidate key should determine all attributes}

[2] R should not contain any partial dependencies

(Proper Subset of cand. Key) $\longrightarrow$ (Non-key attribute) } Partial Dependency

Partial Dependency



X: Any candidate key
Y: Proper subset of "
A: Non-prime attribute.

$Y \rightarrow A$: PARTIAL DEPENDENCY

only if

[a] Y should be proper subset of cand. key

and

[b] A is a non-prime attribute

⇒ NKey ⟶ NKey (not a partial dependency)

⇒

Sid	Sname	Cname
S1	A	C
S1	A	DB
S2	B	C++
S2	B	C
S3	B	DB

Primary Key = (Sid, Cname)

Prime Attr = { sid, cname }

Non-Prime " = { sname }

Now Sid ⟶ Sname

Sid ← Proper subset of cand. key

Sname ← non-key attribute

} Partial Dependency exist b/w these two attributes

Hence it is not in 2NF.

Bcoz, partial dependency exist in this.

To achieve that

$\boxed{\text{DECOMPOSITION}}$

→ Decomposition into 2NF

R_1

Sid	Cname
S_1	C
S_1	C++
S_2	DB
S_2	C++
S_3	DB

Cand.Key: [Sid, Cname]

R_2

Sid	Sname
S_1	A
S_2	B
S_3	B

(Sid → Sname) ⟹ Dependency Preserved.

[Cand.Key = Sid]

⟹ Lossless join Decomposition + Dependency Preserved.

⟹ No PD (Partial Dependency) ⟹ 2NF.

Q: Check PD

R(ABCDEF) : {AB → C, C → D, B → E, B → F}

PD PD

AB : Cand. Key

PD: B → E, B → F

Hence not in 2NF

But it is in 1NF ∵ AB → determine all attributes

Decompose it to 2NF:

R_1 [A B C D]

AB → C

C → D

AB : Cand Key

R_2 [B E F]

B → E

B → F

{B} : Candidate Key

→ 2NF decompn + lossless - cand key from 1 reln + dependency

Q R(ABCDEF), {AB → C, B → F, A → E, E → G}
P.D P.D P.D

Cand key = ABD

· PD → Hence not in 2NF
It is in 1NF (ABD determine all attributes)

3 PD & determinants alag alag hai. Thus three diff tables to remove P.D & one more (table) relation to preserve original relation Candidate key.

2NF Decomposition

| A B C D |

AB → C
ABD: Cand key
still P.D is there

$(B^+ = BF)$
B → F

| B F |

{B}: Cand. Key

$(A^+ = AEG)$

| A E G |

A → E
E → G
A: Cand key

PD from suffer AB → C

| A B C | | AB D | | B F | | A E G |

{AB: Cand Key}

Thus total 4 tables required.

Q: Decompose to 2NF
R(ABCDEF) {AB → C, C → D, B → E, A → C, C → F}
PD PD

Cand key: (AB)

Three PD exist.

~~AB → C~~
E D

| ~~A B C D E~~ |

| AB |

[AB]: CK

PD1

| B E |

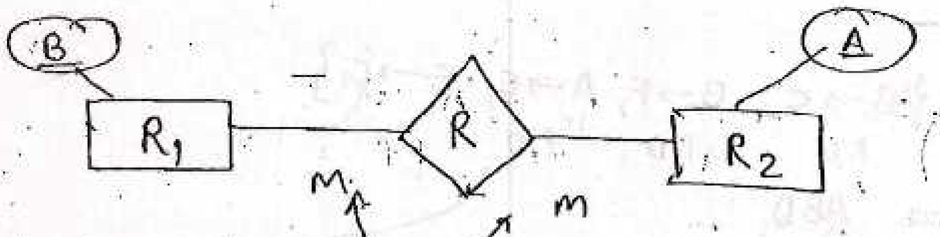
Cand key: B

PD2

| A C F | D |

Cand key: A

R R1 R2



Cardinality = m : m

2NF may not be free from redundancy but redundancy is less than 1NF.

Non Key → Non Key

Proper subset of Cand key → Proper subset of CK

} Dependencies allowed in 2NF that causes Redundancy.

THIRD NORMAL FORM:-

$X \rightarrow Y$ — superkey or Prime attribute

Let R be the Relational schema,

$X \rightarrow Y$ any non-trivial FD over R is in 3NF only if

[a] : X : superkey

or

[b] : Y : Prime attribute

Proper subset of CK → Non Key

Non key → Non Key

} Dependencies not allowed in 3NF

Proper Subset of CK → Proper Subset of other CK } Not allowed in 3NF
⇒ cause Redundancy.

Example:

R(ABCDEF), { AB → C , C → D, D → E , B → F }

L 3NF = ?? How many FD's satisfy 3NF

Cand Key = AB

1. AB → C , AB = Cand. Key = Superkey ✓
2. C → D , X
3. D → E , X
4. B → F , B = Prime X, not Cand. Key — not in 2NF also
   P.D

Decompose into 3NF.

We need to keep separate relations for

C → D
D → E
B → F

If we keep C → D, D → E

| C D E | — violate 3NF

C → D
D → E

∵ C = Cand. Key

C → D — OK

D → E X

(D: not Superkey; E: not Prime attribute)

| AB C | | C D | | D E | | B F |

AB → C C → D D → E B → F

4 relations required.

↓

can't combine any of the above relations, bcoz combining them violate 3NF condition.

$\boxed{A\ B\ C\ D}$	$\boxed{A\ B\ C\ F}$
ok 3NF ✓ $AB \rightarrow C$	$AB \rightarrow C$
X — $C \rightarrow D$ (C: not super key; D: not prime)	$B \rightarrow F$ .... not in 3NF (B: not super key; F: not prime attribute)

3NF Definition

$X \rightarrow Y$ (X: super key or Y: prime attribute)

R is in 3NF only if

[a] R should be in 2NF

[b] Non key should not be transitively determined by key.

3NF case:

Cand. Key (X) → Non Key (Y) | Non key (Y) → Non key (Z)

X:

Cand. Key → Cand. Key (Y) | Cand. Key (Y) → Non Key (Z)

Z determined directly by cand. key, not transitively

Cand. Key (X) → Proper sub. of other Cand. Key (Y) | Proper Subset of other CK (Y) → Non Key (Z)

Transitively non key determined key

X → Y, Y → Z

BCNF :- Let R be the relational schema
$X \to Y$ any non-trivial F.D is in
BCNF only if
X should be candidate Key (super Key)

$$\boxed{\underset{\substack{\uparrow \\ \text{(Super Key)}}}{X} \longrightarrow Y}$$: Non-trivial F.D

→ BCNF is free from redundancies occuring due to single valued dependencies.

→ ① Proper subset CK → Non Key — P.D
② NK → NK
③ P. subset CK → P. subset C.K

} not allowed in BCNF.

Q. Check whether given relation is in 3NF or not.
R(ABCD) , F = {AB → CD, D → A}

$(AB)^+ = (ABCD)$
Cand. Key = AB
$\underset{\substack{\uparrow \\ \text{Super Key}}}{AB} \longrightarrow CD$

AB , D → A
* DB = Cand. Key

$\boxed{\text{Cand Key} = AB, BD}$

$X \rightarrow Y$

Super key or / prime attribute

{AB, BD} = Candidate Key

#

$AB \rightarrow CD$ —— OK (3NF)

Super key

$D \rightarrow A$ —— OK (3NF)

Prime attribute

In 3NF

Dependencies not allowed are

non key → non key

Proper subset of CK → Proper subset of other CK

Hence R is in 3NF

BCNF = ?

$X \rightarrow Y$

Super key

$AB \rightarrow CD$ —— OK

Super key

$D \rightarrow A$ ✗

not a superkey

Hence not in BCNF.

Decomposition into BCNF

→ Attribute suffering from BCNF definition separated.

| | A D |

D: candidate key

D → A

| B C | A D |

For lossless decompositi // [illegible].

| B C D | A D |

↑

not trivial dependency in R_1 = ?

$B^+ = B$ $BC^+ = BC$

$C^+ = C$ $BD = ABCD$

$D^+ = AD$ $CD = ACD$

$BD \longrightarrow C$

BD: Cand. Key

⇒ BCNF Decomposition
\+
Lossless decomposition
\+
(Not preserving dependency)

NOTE :-

Some relations may not be possible to decompose into dependency preserving BCNF decomposition.

↳ WHY so??? WHY not in 1,2,3NF???

∴ ⟹

2NF

Proper Sub. CK (X) ⟶ Non Key (Y)

.... X

X	Y

3NF

Non Key (A) ⟶ Non Key (B)

.... A

A	B

BCNF

Proper Sub. of CK (X) ⟶ Proper subset of other CK (Y)

.... X

X	Y

» Original candidate keys are not maintained as it is.

one prime attribute removed from one relation & kept in new relation. ∵ of this movement of prime attribute from original relation to some other relation, leads to, loss of one original cand. key And hence dependencies due to that cand. key are no more maintained (sustained) or preserved. i.e lost.

Q:- Decompose into 2NF if not 2NF
Decompose into 3NF if not 3NF
Decompose into BCNF if not BCNF.

(1) R(ABCDEFGHIJ)
{AB→C, A→DE, B→F, F→GH, D→IJ}

(2) R(ABDLPT), {B→PT, T→L, A→D}

(3) R(ABCDEFGH), {AB→C, AC→B, AD→E, B→D, BC→A, E→G}

(1) ⟹ R(ABCDEFGHIJ)

(a) Check for 2NF

No. P.D ------ Prop sub C.K → non key

Cand. Key = (AB)

{AB→C, A→DE, B→F, F→GH, D→IJ}
✓ PD PD ✓ ✓

left side not equal so two tables

| A D E | | B F |

A^+ = [ADIJE] B^+ = [BFGH]

If PD exist and A→DE, B→F and LHS of PD not equal then two relation bnane padengee with R_1 – A as key, R_2 – B as key and attributes ∈ R_1 = [A^+] and attribute ∈ R_2 = [B^+]

| A B C | | A D E I J | | B F G H |

ABC	ADEIJ	BFGH
AB→C	A→DE, D→IJ	B→F, F→GH
{AB} Cand Key	{A} Cand Key	{B} Cand Key

2NF Decomp + lossless + Dependency Preserved.

(i) For 3NF

X → Y

Super key or Prime attribute

R_1 | A B C |

✓ AB → C

R_2 | A D E I J |

✓ A → DE

D → IJ ×

not superkey, not prime

R_3 | B F G H |

✓ B → F

F → GH ×

not S.K, not prime

Thus R_2 and R_3 are not in 3NF

| A B C | | A D E I J | | B F G H |

R_1 | A B C |

AB → C

S.K

R_2 | A D E |

A → DE

S.K

R_3 | D I J |

D → IJ

S.K

R_4 | B F |

B → F

S.K

R_5 | F G H |

F → GH

S.K

⇒ BCNF also

3NF + BCNF

\+

Dependency Preserving (all dependency preserved)

\+

Lossless (one table CK common in other and ~~so~~ on)

Lossless = ?

↳ B/w R_1 & R_2 — A is common
which is also the C.K of R_2.
Thus R_1 & R_2 can be combined.
$\boxed{R_1 \; R_2}$

↳ B/w R_2 R_3 — D is common, which is also the C.K of R_3. Thus R_2 & R_3 can be combined.
$\boxed{R_1 \; R_2 \; R_3}$

↳ B/w R_4 & R_5 → F is common (C.K of R_5)
$\boxed{R_4 \; R_5}$

↳ B/w R_1 and R_4 — B is common, which is also the C.K of relation R_4.
Hence combined.

$\boxed{R_1 \quad R_2 \quad R_3 \quad R_4 \quad R_5}$

Thus Lossless Decomposition.

1 Sep. 2011

Ganpati Bappa Moriya

Jai Ganesh

2NF

R(ABCD) $\{AB \rightarrow C, B \rightarrow D\}$

Cand Key = AB

non prime

2NF Decomposition

lossless

$\boxed{A\ B\ C}$ $\boxed{B\ D}$

Cand Key {AB}

$B \rightarrow D$

{B}: Cand.

Only the non-prime attribute D, is moved from original relation into another.

3NF

R(ABCD) $\{AB \rightarrow C \quad C \rightarrow D\}$.

Cand Key: AB

non prime

3NF Decomposition

$\boxed{A\ B\ C}$ $\boxed{C\ D}$

$C \rightarrow D$

$AB \rightarrow C$

{C}

CK: {AB}

↑

original CK retained

BCNF

R(ABCD) $\{AB \rightarrow CD, D \rightarrow A\}$

Cand. Key = {AB, DB}

BCNF Decomposition

$D \rightarrow A$ (fail BCNF)

make them a relation

$\boxed{A\ D}$

$D \rightarrow A$

{D}: CK

Prime A → Prime A

Now left attri $\boxed{B\ C}$

To maintain lossless decomp. Cand. key made common.

$\boxed{B\ C\ D}$ $\boxed{D\ A}$

$BD \rightarrow C$

{BD}: Cand. K

$D \rightarrow A$

{D}: CK

≠

(AB): CK lost ∵ of movement of a prime attribute from original relation to new relation lead to loss of cand. Key.

Q: R(ABCDE) {A → BC, CD → E, B → D, E → A}

Q R(ABCDEFGH) {AB → C, AC → B, AD → E, B → D, BC → A, E → G}

Cand keys : { ABFH
BCFH
ACFH }

2NF ⇒ P.D

P Dependency:

Proper Subset of CK → Non Key

AB → C ✓
AC → B ✓
AD → E ✓
B → D P.D — 2NF violate
BC → A ✓
E → G ✓

2NF Decomposition:

B → D violate

| A B C ~~D~~ E F G | | B D | (initially)

AB → C
BC → A
AC → B
E → G

B → D
{B} C'K

But AD → E is not preserved here
to achieve that maintain a separate table for that

R_1 | A B C F H |
AB → C
BC → A
AC → B
{AB FH; BCFH, ACFH}
3NF ✓

R_2 | B D |
B → D
{B}
3NF ✓

R_3 | A D E G |
$E^+ = EG$
AD → E
E → G
{AD}
Not in 3NF

Dep. Preserving
+
Lossless
+
No PD
(2NF)

$R_1(ABCFH)$	$R_2(BD)$	$R_{31}(ADE)$	$R_{32}(EG)$
$AB \to C$, $BC \to A$, $AC \to B$		$AD \to E$, $\{AD\}$	$E \to G$, $\{E\}$
Not in BCNF	BCNF ✓	BCNF ✓	BCNF ✓

3NF lossless + join Dep. Presving

BCNF Decomposition:

↳ $X \to Y$ $X \in$ Cand. key

Dep. failing BCNF = $AB \to C$
$BC \to A$
$AC \to B$

R_{11} ki koi bhi ek key common di bna di hai

$R_{11}(ABC)$	$R_{12}(\underline{AB}\ FH)$	$R_2(BD)$	$R_{31}(ADE)$	$R_{32}(EG)$
$AB \to C$, $BC \to A$, $AC \to B$	BC, AC	$B \to D$	$AD \to E$	$E \to G$
$\{AB, BC, AC\}$	$\{ABFH\}$: Cand key	$\{B\}$	$\{AD\}$	$\{E\}$

If no non-trivial functional dependencies Then relation is always in BCNF.

Now the relation is in BCNF.

Original relation splitted into 5 sub-relations to reduce it to BCNF.

Q:- R(ABCDE) $\{A \to BC, CD \to E, B \to D, E \to A\}$

Candidate Keys: {A, E, CD, BC}

The given relation is in 3NF but not in BCNF

BCNF Decomposition:

$B \to D$ violating BCNF

A B C E		B D

Non trivial dependencies: $A \to BC$, $E \to A$, $BC \to AE$; {A, E, BC}

$B \to D$; {B}

D = Prime attribute moved from original. Hence dependency $CD \to E$ is lost.

This decomposition is

$\boxed{\text{BCNF + Lossless + Not preserving dependency}}$

If every candidate key of relation R is simple candidate key (No compound candidate key in R) Then the relation R should be 2NF but may or may not be in 3NF or BCNF.

— Simple Candidate Keys = Single attribute

2NF: $Y \to A$ (Y: Proper subset of CK; A: Non key)

$A \to$, $B \to$, $C \to$ CK = {A, B, C}

Proper subset = ϕ

Thus no possibility of partial dependency. Hence 2NF 100% ensured.

Example R(ABCDE) { A→B B→C C→D D→E C→A }

Cand. Keys = {A, B, C}

→ Relation in 2NF
(No PD exist)

→ Not in 3NF ∵ of (D→E)

Non Key → Non Key

If every attribute of relation R is prime attribute (No non-prime attribute). Then the relation always in 3NF but may or may not be in BCNF.

3NF X → Y

Super Key — prime attribute

↳ in this case this condition always true for any non-trivial dependency.

{F/T OR T} ⇒ Always true

Example:- R(ABCD) { A→B, B→C, C→A }

Candidate key = { AD, BD, CD }

In 3NF but not in BCNF

∵ A, B, C ≠ Cand. key

{A→B, B→C
C→D, D→A}

A^+ = ABCD
B^+ = ABCD
C^+ = ABCD
D^+ = ABCD

↳ Candidate key

every attribute C.K

If R is in 3NF and all candidate keys are simple candidate keys Then Relation R is always in BCNF.

BCNF:

Proper Subset of CK → Proper subset of other CK

(Empty Attribute)

each cand. key = only one attribute key, prop subset not possible

this non-trivial FD not possible with 3NF Relation and only simple cand. keys.

Binary Relation: (Relation with two attributes) is always in BCNF

Example R(AB)

ⓐ $A \to B$; CK = A
ⓑ $B \to A$; CK = B
ⓒ $\{A \to B, B \to A\}$; CK = {A, B}
ⓓ No non-trivial F.D ; C.K = {AB}

Relation R with No Non trivial FD's is always in BCNF

Example R(ABC)

ABC: Candidate Key

BCNF :- $X \to Y$ non-trivial depen. is in BCNF

X: Superkey

when no non trivial dependency relation R is always in BCNF as its violation is not possible.

To violate BCNF atleast one non-trivial FD with no

Identifying Highest Normal Form

① R(ABCDEF) {AB→C, C→D, D→E, E→F, F→A}

Cand. keys : { AB, BC, BD, BF, BE }

F:	AB→C	C→D	D→E	E→F	F→A	
	✓	×	×	×	×	BCNF
(Prime attribute)	✓	✓	✓	✓	✓	$\boxed{3NF}$

② R(ABCD) {AB→C BC→D}

Cand key = { AB }

F :	AB → C	BC → D	
	✓ ✓	× ×	BCNF
	✓ ×	× ×	3NF
	✓ ×	✓ ×	2NF (PD)
	✓	✓	1NF

super key

prime attribute

super key

Prop sub C.K → not key

No

③ R(ABCDEF) Cand. key = { AB, BF, BE, BC }

F: {AB→C, C→DE, E→F, F→A}

AB→C	C→DE	E→F	F→A	
✓	×	×	×	BCNF
✓	×	✓	✓	3NF
✓	C→D (PD)	✓	✓	2NF
				1NF

(4) R(ABCDEFGH) Cand. Key = {A, F, H, C} # Simple Keys =2NF issue

{AB→CD, D→EG, F→H, C→EF, H→A, G→B, A→B}

~~BCNF~~

~~3NF~~ 3NF ✓ × ✓ ✓ ✓ × ✓

~~2NF~~ (2NF) #

2006

(5) $R(F_1, F_2, F_3, F_4, F_5)$ C.Key = $\{F_1F_2\}$

$\{(F_1 \to F_3), (F_2 \to F_4), F_1F_2 \to F_5\}$

P.D P.D ✓ Not in 2NF (1NF) #

(2003) DOB → age

~~Age → eligibility~~

Name → Rno

Rno → Name

~~Cno → Cname~~ not in Relation

~~Cno → Instruction~~

~~(Rno, Cno) → Grade~~

Q. Dob → age

Name → Rno

Rno → name

Cand. key = {(DoB, name), (~~Name~~ DoB, Rno)}

the relation is (Rno, name, dob, age)

1 NF not in 2NF

Multivalued Dependencies : MVD

If relatn consists two or more independent relations then redundancy is possible.

If two or more multivalued attributes exists in the relation, then multivalued dependency (MVD) possible while conversion of atomic attributes.

\# R(A,B,C,D)

$A \to B$

CD not included into FD so {CD = multivalued}

$\Rightarrow$ R not in 1NF

So convert it to 1NF

R (ABCD) {A→B}

{ACD} : Candidate Key

1NF

(Then there may be MVD possible)

Example:

Teacher	Course	Book
A	C/DB	K/N
B	DB	K.N

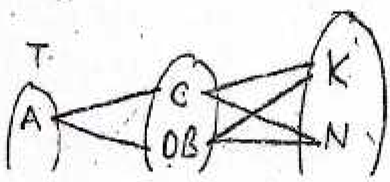
Not in 1NF

R

Should not contain mv attributes to be in

Teacher : Primary Key

Course, Book } multi-valued attributes



depend on

Teacher, Book } Depending attributes
Teacher, Course

Course, Book] Not depending.

T	C	B
A	C	K
A	DB	N
B	DB	N

Here Independent attributes (Course, Book) become Depending which is not correct.

(Data loss occurs)

⇓

T	C	B
A	C	K
A	C	N
A	DB	K
A	DB	N
B	DB	N

Redundancy exists (problem)

Cand. key = {TCB}

BCNF Normal Form

→ Redundancy exist not bcoz of single valued dependency, but bcoz of mvalued dependency.

⇒ Let R be the relational schema X, Y be the attribute sets of R and $Z = R - \{X \cup Y\}$ // All attributes except X & Y.

$X \rightarrow\rightarrow Y$ exists in R only if

t_1, t_2, t_3, t_4 tuples such that

(a) $t_1.x = t_2.x = t_3.x = t_4.x$

and

(b) $t_1.y = t_2.y$ && $t_3.y = t_4.y$

and

(c) $t_1.z = t_3.z$ && $t_2.z = t_4.z$

t	x	y	z
t_1	x_1	y_1	z_1
t_2	x_1	y_1	z_2
t_3	x_1	y_2	z_1
t_4	x_1	y_2	z_2

Hence in this table $\boxed{X \rightarrow\rightarrow Y}$

Interchanging tuples t_2 & t_3 (Complimentation Rule).

	x	y	z
t_1	x_1	y_1	z_1
t_3	x_1	y_2	z_1
t_2	x_1	y_1	z_2
t_4	x_1	y_2	z_2

$(x \rightarrow\rightarrow z)$

Complimentation Rule :- If $X \rightarrow\rightarrow Y$ exists then

$X \rightarrow\rightarrow R-(X \cup Y)$ also exists.

Eg $R(x,y,z)$

$x \rightarrow\rightarrow y$ exist

$x \rightarrow\rightarrow z$ also exist.

Trivial MVD :- $X \supseteq Y$ [Trivial F.D $X \rightarrow Y$]

If $X \supseteq Y$ or

$X \cup Y = R$ then $X \rightarrow\rightarrow Y$ is trivial MVD.

⟹ Atleast three attributes are required to get non-trivial MVD.

⟹ Non-trivial Functional Dependency = atleast 2 attributes required

Argumentation:

If $X \rightarrow\rightarrow Y$ and $Z \supseteq W$

Then $\boxed{XZ \rightarrow\rightarrow YW}$

$x \rightarrow y$ & $w \supseteq z$.

$x \rightarrow y$

$\boxed{xw \rightarrow yz}$

$A \rightarrow B$

$AC \rightarrow B$

$ACD \rightarrow BC$

- Transitivity:

If $X \to\to Y$ and $Y \to\to Z$

Then

$\boxed{X \to\to (Z-Y)}$ — // all attributes of Z but not Y.

Replication

If $x \to Y$ then $x \to\to Y$

{ Every FD is replication of MVD }

$\Rightarrow$ MVD do not obey

$x \to yz \Rightarrow x \to y$, $x \to z$

$x \to y$ & $x \to z$ then $x \to yz$

} holds for only FD but not MVD

F Identify all MVD —

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

$(A \to\to B)$

also

$(A \to\to C)$ — complimentation rule

$AC \to\to B$

$AC \to\to BC$ } Augmentation

$AB \to\to C$

$AB \to\to CB$ } "

Trivial MVD

$x \supseteq y$ or

$\boxed{x \cup y = R}$

Replication :-

$B \to A$

$C \to A$

$BC \to A$

Single V.D

$A \to B$

$A \to C$

$AB \to C$

$AC \to B$

~~X~~ not possible.

$B \to\to A$ — $BC \to\to A$, $BC \to\to AC$ — Augmentation

$C \to\to A$ — $CB \to\to A$, $CB \to\to AB$

$BC \to\to A$

Again applying properties.

#

T	C	B
A	DB	K
A	DB	N
A	C	K
A	C	N
B	DB	K

— Relation is in BCNF
— Multivalued dep:

$T \rightarrow\rightarrow C$
$T \rightarrow\rightarrow B$] Non-trivial MVD

— Still redundancy exist bcoz of MVD.

FOURTH NORMAL FORM:

Let R be the relational schema, is in 4NF only if

Imp

[a] R should be in BCNF
{Non-trivial FD $X \rightarrow Y$ allowed only when X is superkey}

and

[b] R should only allowed non-trivial MVD

$X \rightarrow\rightarrow Y$ with X should be superkey

* Super Key or Candidate keys identified using FD (Functional Dependency) only, not based on multivalued dependency (MVD).

Above relation is in BCNF but not 4NF

4NF (a) BCNF — ok
(b) $T \rightarrow\rightarrow C$, $T \rightarrow\rightarrow B$] MVD, T is not superkey
Hence not in 4NF.

Decompose into 4NF

$T \to\to C$

T	C
A	DB
A	C
B	DB

$T \to\to C$

↓

Trivial now

∴ $T \cup C = R$

$T \to\to B$

T	B
A	K
A	N
B	K

$T \to\to B$

↓

Trivial

$T \cup B = R$

No Non Trivial MVD Thus it is in $\boxed{\text{4NF}}$

Conclusion :-

	2NF	3NF	BCNF	4NF
0% Redundancy	✗	✗	✗ (MVD)	✓
Lossless Join	✓	✓	✓	✓
Dependency Preservation	✓	✓	✗	✗