$AC \to G$

$D \to E$

$D \to G$

$BC \to D$

$CG \to B$

$CG \to D$

$ACD \to B$

$CE \to A$

$CE \to G$

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

D is extraneous.

$\Rightarrow AC \to B$

Redundant

$\boxed{\cancel{AC \to G}}$

$D \to E$ ✓

$D \to G$ ✓

$BC \to D$. ✓

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

$CG \to D$.

✓ $AC \to B$.

$CE \to A$

$\boxed{\cancel{CE \to G}}$

$(AC)^+ = \{A, C, G, D, D, E\}$

$(AC_B)^+ = A, C, B, D, G, B, D, E$

$D^+ = \{D, E\}$

$(D_E)^+ = \{D, E, G\}$

$(D)^+ = \{D, E, G\}$

$(D_G)^+ = \{D, E\}$

Scanned with CamScanner

1) $A \to B$
$B \to C$
$A \to C$
$AB \to B$
$AB \to C$
$AC \to B$

Step 1 - Decomposition.

schema is already decomposed.

Step 2 → find extraneous attribute at left side.

$A \to B$ $\quad A^+ = \{A, B, C\}$
$AB \to B$ $\quad$ B is extraneous.
$\implies A \to B$

$A \to B$ $\quad A^+ = \{A, B, C\}$
$B \to C$

$AB \to C$ $\quad$ B is extraneous.
$\implies A \to C$

$A \to B$
$AC \to B$ $\quad$ B extraneous.
$\implies A \to B$

$A \to B$ ✓
$B \to C$ ✓
$A \to C$ ✗

step - 3 find extraneous attribute at rightside.

$\boxed{A \to B}$ ✓ $A^+ = \{A, B, C\}$
$A \to C$
$B \to C$

$A \to B$
$\boxed{A \to C}$ X $A^+ = \{A, C\}$
$B \to C$

FC

$$\boxed{\begin{array}{l} A \to B \\ B \to C \end{array}}$$

Ex-2 $AB \to C$
$A \to B$
$B \to C$

$AB \to C$ A extraneous.
$B \to C$
$\Rightarrow B \to C$.

FC.

$$\boxed{\begin{array}{l} A \to B \\ B \to C \end{array}}$$

Ex-3

$AB \to CD$.
$B \to C$.
$BC \to D$.
$CD \to EF$
$E \to F$.

step-1 **Decomposition.**

$AB \to C$
$AB \to D$.
$B \to C$
$BC \to D$.
$CD \to E$
$CD \to F$
$E \to F$

step 2 $AB \to C$ A B extraneous
$B \to C$ $\Rightarrow$ $B \to C$,

$AB \to D$ $A^+ = \{A\}$
$B^+ = \{B, C, D, E, F\}$.

$BC \to D$.   C B extraneous
$B \to C$ $\Rightarrow$ $B \to D$
$B \to C$

$B \to D$.
$AB \to D$ $\Rightarrow$ A B extraneous
$\Rightarrow$ $B \to D$.

$B \to C$ ✓

$B \to D$ ✓

$CD \to E$ ✓

$\boxed{CD \to F}$ ✗

$E \to F$ ✓

$CD = C\,D\,E\,F$

FC

$B \to CD$

$CD \to E$

$E \to F$

Ex

$A \to BC$

$CD \to E$

$E \to C$

$D \to AEH$

$ABH \to BD.$

$DH \to BC$

Step-1 Decomposition.

$A \to B$

$A \to C$

$CD \to E$

$E \to C$

$D \to A$

$D \to E$

$D \to H$

$DH \to B$

$DH \to C$

$ABH \to B\cancel{D}$

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

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

$CD \to E$ C D

$D \to E$ extraneous

$\Rightarrow D \to E$

$ABH \to D$

$\Rightarrow$ $ABH \to B$ B is extraneous.

$\Rightarrow AH \to B$

$ABH \to D$

$\Rightarrow AH \to D$

$\Rightarrow$ $DH \to B$ $D \to H$

$DH \to C$ H is extraneous.

$D \to B$ $D^+ = \{AH$

$D \to C$

$\Rightarrow$ $D \to$

$A \to B$ ✓

$A \to C$ ✓

$D \to E$ ✓

$E \to C$ ✓

$D \to A$ ✓ $D \to A$ $A \to B$ $A \to C$

$D \to E$ ✓

$D \to H$ ✓ $D \to B$ & $D \to C$

$D \to B$ ✗   are extraneous

$D \to C$ ✗

$AH \to B$ ✗

$AH \to D$ ✓

FC

$A \to BC$

$D \to AEH$

$AH \to D$

$E \to C$

Ex :- $AB \to C$
$A \to B$
$B \to A$

$AB \to C$ $\Rightarrow$ $A \to C$
$A \to B$ $\quad$ $A \to B$

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

FC
$A \to BC$
$B \to A$

Ex $WXY \to Z$, $X \to V$, $V \to Z$.

$X^+ = \{X, V, Z\}$

$WXY \to Z$ are extraneous.

$X \to Z$
$X \to V$
$V \to Z$

$X \to VZ$
$V \to Z$

DBMS

(*) Functional Dependencies

$f: \alpha \to \beta$

$\alpha \subseteq R \quad \beta \subseteq R$

R

	α	β
t_1	a	b
t_2	a	b
	c	3
	d	4

$\alpha \to \beta$

if $t_1[\alpha] = t_2[\alpha]$

$\Rightarrow t_1[\beta] = t_2[\beta]$

$f: \alpha \to \beta \leftarrow$ Dependent

Determinant

$\alpha \to \beta$

Trivial

$AB \to A$

if $\beta \subseteq \alpha$

non Trivial ✓

$\beta \not\subseteq \alpha$

$A\cancel{B} \to ABC$

(*)

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

(1) ✓ $A \to BC$ holds good

(4) $DC \to A$ ✓

(2) $DE \to C$ ✓

(3) $C \to DE$ ✗ not hold

Scanned with CamScanner

for any α if get two different value of B then the dependency does not hold.

→ if α is unique → dependency always hold.

$A \to C$

A	B	C	D	E
a	2	3	4	5
b	a	3	4	5
c	2	3	6	5
d	2	3	6	6

→ If $\alpha \to B$

1. check the value of α. if value of α is different then dependency hold.

2. if all value of B is same then also dependency hold

R (A B C)

$A \to B$

$B \to C$

$F = F_1 + F_2$

Directly visible $A \to C$

$A \to B$

i.e. $A \to BC$

Page
Date

Closure set of attributes

A closure set of FDs is a set of all possible FDs that can be derived from given set of FDs.

→ also referred as complete set of FDs.

functional dependency is denoted by F

& closure of F is denoted by F^+

Ⓧ R (A B C)

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

Closure set of attributes

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

Pseudo transitivity

$\alpha \rightarrow \beta$ & $\beta\gamma \rightarrow \delta$

$\Longrightarrow \alpha\gamma \rightarrow \delta$

Scanned with CamScanner

(*) Inference Rules.
(Armstrong Axioms)

→ let A, B, C is subsets of the attributes of relation R.

① Reflexivity. A → set of Attributes

if B is subset of A $B \subseteq A$
then $A \to B$.

② Augmentation
if $A \to B$ then $AC \to BC$

③ Transitivity. C → set of Attributes
if $A \to B$ and $B \to C$
then $A \to C$

④ Self Determination.
$A \to A$

⑤ Decomposition.
if $A \to BC$ then
$A \to B$ & $A \to C$

⑥ Union
if $A \to B$ and $A \to C$
then $A \to BC$.

⑦ Composition.
if $A \to B$ & $C \to D$
then $AC \to BD$

Scanned with CamScanner

Redundant functional dependencies

A functional dependency in a set is redundant if it can be derived from other functional dependencies in the set

Algorithm : Membership Algorithm

I/P : Let F be a set of FDs for relation R.

steps :

1. $F' = F - f$

2. $T = A$

   T = Determinant of $A \to B$

3. for each FD : $X \to Y$ in F' Do

   a) If $X \subseteq T$ then X is contained in T

   $T = T \cup Y$

4. if $B \subseteq T$ then

   $f : A \to B$ is redundant

R (A, B, C, D, E)

F = (A → B, C → D, BD → E, AC → E)

(1) AC → E Redundant or not

Step 1 : F (A → B, C → D, BD → E)

Step 2 : T = AC

Step 3 : T = AC + B (A → B)
= ACB

T = ACB + D (C → D)
= ACBD

T = ACBD + E (BD → E)
= ACBDE

Step 4 : f : AC → E
redundant

Closure Set of Attributes

(1) R (ABCDEFG)

A → B

BC → DE

AEG → G

$(AC)^+$ = AC + B (A → B)

= ACB

= ACB + DE (BC → DE)

= $\boxed{ACBDE}$

(2) R (ABCDE)

A → BC

CD → E

B → D

E → A

B^+ = BD (B → D)

(3) R (ABCDEF)

AB → C

BC → AD

D → E

CF → E

$(AB)^+$ = ABC (AB → C)

= ABC + AD (BC → AD)

= ABCD

= ABCDE (D → E)

Page
Date

(*) $A \to BC$
$CD \to E$
$E \to C$
$D \to AEH$
$ABH \to BD$
$DH \to BC$
$\boxed{BCD \to H\ ?}$

$(BCD^+) = BCD$
$= BCDE$
$= ABCDEH$

$\boxed{BCD \to H \text{ is valid.}}$

as closure contains H.

(*) Equivalence of functional dependencies

f	G
$A \to C$	$A \to CD$
$AC \to D$	$E \to AH$
$E \to AD$	
$E \to H$	

$(A)^+ = ACD$ i.e. $A \to C$ hold in f
$(AC)^+ = ACD$ i.e. $AC \to D$ hold
$(E)^+ = EAHCD$ $E \to AD$, $E \to H$ hold.

$f \subseteq G$

$A^+ = ACD$
$E^+ = ADHCD$
both dependencies hold
$G \subseteq f$

$\Rightarrow f \subseteq G$ & $G \subseteq f$
$\Rightarrow f = g$.

Scanned with CamScanner

Page
Date

Irreducible set of Functional Dependencies (Canonical form)

R(wxyz)

$x \to w$
$wz \to xy$
$y \to wxz$

$\alpha \to \beta y$ Decomposition
$\Rightarrow \alpha \to \beta$ & $\alpha \to y$

Step 1 Decomposition

$x \to w$
$\boxed{wz \to x}$ Redundant element
$wz \to y$
$\boxed{y \to w}$ Redundant possible
$y \to x$
$y \to z$

$(x^+) = xw$

$(x^+) = x$ not Redundant

$(wz)^+ = wxyz$

$(wz)^+ = ywzx$

$\boxed{wz \to x \text{ Redundant}}$

$(wz)^+ = wzy$

A — D A →
1 → 3
1 → 4

Page
Date

$(Y)^+ = wxyz.$

$y^+ = xyzw$

$y^+ = yz$

$y^+ = xwy$

$x \to w$
$wz \to y$
$y \to x$
$y \to z$

$(wz)^+ = wzyx$

$w^+ = w$ i.e. z is essential attribute

$z^+ = z$ i.e. w is also essential

$\boxed{\begin{array}{l} x \to w \\ wz \to y \\ y \to xz \end{array}}$

Canonical form
irreducible form.

sometime you have different canonical form

Keys

Superkey : Set of attributes by which you can find the row uniquely

Candidate key
→ Candidate key is a superkey for which no proper subset is super key.

Primary key :-

A primary key is a candidate key that chosen by database designer to identify touples uniquely in relation.

Foreign key :

A foreign key is set of one or more attributes whose values are derived from the primary key attributes of another relation.

Alternate key :- unique key.

allows only one NULL value

(4) Relational Algebra :

is a language for expressing relational database queries.

procedural query language.

algebraic Operations :-

(1) Selection

Operation :- Selects tuples from a relation that satisfy a given condition.

It is used to select particular touples from relation.

Symbol - σ (sigma)

Notation : σ (condition) < Relation >

Op

Keys :

A	B	C	D
1	a	x	
2	b	y	
3	b	x	
4	c	y	

A → BC

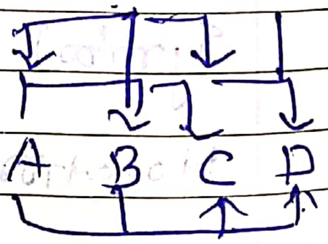
BC ← A

key is a set of attributes by which any row can be identified uniquely.

C → AB ✗

Candidate key. — minimal set of superkey is c.k.

		Candidate key
A → BCD.	S.k.	✓
AB → CD	S.k.	✗
ABC → D	✓	✗
BD → AB	✓	✓
C → AB	✗	✗



Supe

subset of a superkey is also a super key then its not a c.k.

Primary key :-

(1) R (A B C D E F G H)

AB → C

A → DE

B → F

F → GH

AB do not have any incoming edge. i.e. AB are essential. AB are the part of C.K.

$(AB)^+$ = ABCDEFGH.

(2)

AB → C
BD → EF
AD → G
A → H

A B C D E F G H

ABD ~~essecial~~ essential.

$(ABD)^+$ — ABD CEFGH
C.K.

(3) R (A B C D E)

BC → ADE

D → B

C is essential.

AC BC CD CE

BC, D C.K.

$(C)^+$ = C

$(AC)^+$ = AC

$(BC)^+$ = BCADEB

(*) R (A B C D E)

AB → CD
D → A
BC → DE

$(B^+) = B$

$(AB)^+ = ABCDE$ ✓ C.K

BC = BCDEA ✓ C.K.

BD = ABCDE ✓ C.K.

BE = ✗

(*) R (W X Y Z)

Z → W
Y → XZ
WX → Y

$(W)^+ \rightarrow W$ ✗

$(X)^+ \rightarrow X$ ✗

$(Y)^+ \rightarrow (XZWY)$ C.K ✓

$(Z)^+ \rightarrow W$

Page
Date

$(wa)^+ \rightarrow$ wayz c.k ✓

$(az)^+ \rightarrow$ azUy c.k ✓

$(yz)^+ \rightarrow$ waуz c.k. X

w.z → X

(*) R (ABCDEFGH). (4)

A → BC
B → CFH

Page
Date

(*) Normalization - process to eliminate Redundancy.

student

R. N	Name	age	B Id		HOD	Pho
	A	20	101	C S	XYZ	123
	B	19	101	CS	XYZ	123
	C	21	101	C S	XYZ	123

DSV

1. Insertion anomaly. —
2. Deletion anomaly. —
3. Update anomaly. —
4. Inconsistency & 5. errors. increase size & time

Redundancy leads to inconsistency.

Student ← F.K.

1.

Roll No	Name	Age	B-ID

Branch

2.

B ID	HOD Name	Branch	HOD No

Redundancy - when some data are stored multiple times in database unnecessarily.

- 1 NF
- 2 NF
- 3 NF
- BCNF

F.D.

- 4 NF
- 5 NF
- DK NF

1st NF

Roll No	Name	Course.
101	Modi	CN, OS
102	Sonia	DBMS, CO

Every cell contains atomic value.

no multivalued attributes in same cell

101	Modi	CN
101	Modi	OS
102	Sonia	DBMS
102	Sonia	CO

2NF :-

R(ABCD)

AB are essential. $(AB)^+ = ABCD$

$AB \leftarrow$ C.K.

prime attribute :-

A & B both are prime attribute as they are part of C.k

C,D are non prime attribute.

Using prime attribute you can find non prime attribute

$B \rightarrow C$

C is dependent on part of candidate key

this kind of dependency is called partial dependency.

→ there should be no Partial Dependency. & it must be in 1st NF.

→ Conversion to 2NF

R_1 (A B D) Candidate key + attributes totally dependent on key

R_2 (B C)

(3) 3NF – It should be 2NF & there should be no transitive Dependency.

R (A B C D)

AB are essential

$(AB)^+ = ABCD$

CK.

i.e. AB are prime attribute

CD are non prime attribute.

$AB \rightarrow C$

$C \rightarrow D$

There is no Partial dependency.

→ non key attribute forms non key attribute is transitive dependency

Conversion :-

R_1 (A B C)

R2 (C D) make C primary key.

BCNF — nonprime attribute dependent on prime attribute

$\alpha \to \beta$

P/NP P

R(A B C)

$(A^+) = A$

$AB \to C$ $(AB)^+ = ABC$

$C \to B$ $(AC)^+ = ABC$

P/NP prime attribute.

2NF ✓ no transitive dependency.

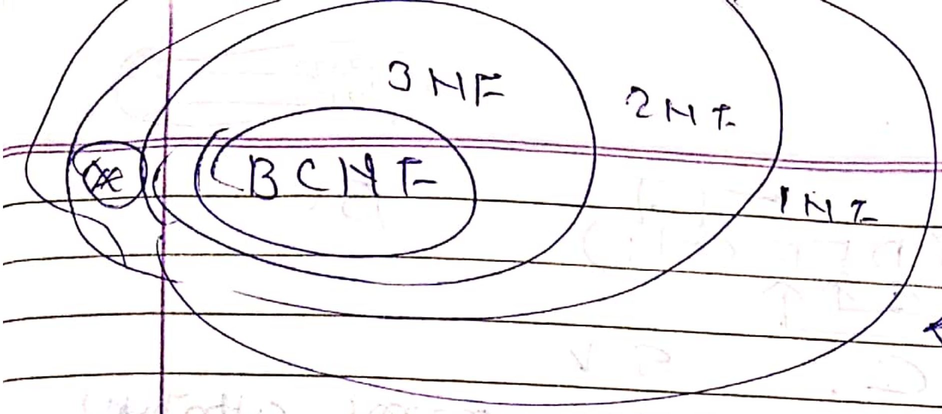
3NF ✓

BCNF ✗ as dependency on prime attribute

→ $\alpha \to \beta$

α: superkey.

as C is not superkey it is not BCNF



① $\alpha \rightarrow \beta$
α is a superkey.
then it is in BCNF

② When BCNF fails 3NF

$\longrightarrow$ $\alpha \rightarrow \beta$

P	N.P.	P.D.
N.P	N.P.	T.D.

C.K. if α is not S.K. or C.K. and β is prime then table is in 3NF

③ 2NF

$\alpha \rightarrow \beta$
prime is β is prime.

Page
Date

① R (ABCDEFGH) BC

AB → C. S. K.

not C.K → A → DE (non prime attribute)

B → F D.B. n.p and

F → GH depends not part of C.K

AB S. K.

not in BCNF

not in 3 NF

not B 2 NF

1 NF

It B in 1 NF.

② R (ABCDE) P = CD

NP – A B E

CE → D S. K. CE candidate key

D → B not S. K.

C → A B B not prime

A D N. P and are dependent on part of C. K

not in BCNF

not in 3 NF

not in 2 NF

1 NF

(3) ABD BCD

AB → C
DC → AE
E → F

DC → E
n.k n.k
non prime Attribute
E depend on
part of c.k
D.C.

BCNF ✗
3NF ✗
2NF ✗
1NF

(4) R (ABCDEFGHI)

ABD

AB → C   SK ✗
BD → EF   not in 3NF
AD → GH
A → I

C depends on part of C.k

2NF ✗

(5) AB BD BC
AB → CD
D → A
BC → DE

not in BCNF
It is in 3NF

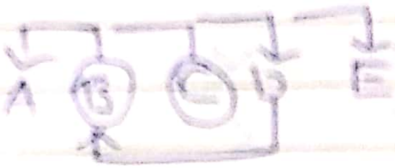
(c) BC → AD ~~E~~

D → B prime

BC CD

BCNF ✗

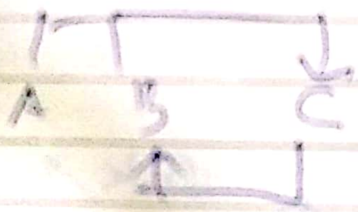
3NF ✓



BC → ADE

D → B

(BC) = Ck



B C

① R (A B C D E F)

C → F

CE essential.

E → A

$(CE)^+$ = CEFABD
= R.
= C.K.

EC → D

A → B

BCNF = X
3NF = X C → F
2NF = X C → F
1NF = ✓

② R (A B C D E H)

A → B ✓
BC → D
E → C X
D → A

EH essential

$(EH)^+$ = EHC X

$(EHA)^+$ = EHCBDA ✓

BCNF = X

$(BEH)^+$ = EHCBDA ✓

3NF = E → C
X

$(CEH)^+$ = HEC X

2NF = X
E → C

$(DEH)^+$ = EHCABD ✓

1NF ✓

Page
Date

(A B C D E F G)

AB → CD   essential
DE → P.   A B
C → E
D → C $(AB)^+$ = ABCDPEG
B → G

BCNF = X
3NF = X   DE → P   P, B not prime
2NF = X   B → G   G partly depend on B

1 NF

(*) R (A B C D E F G H)

CH → G   (B) essential

A → BC   (AD) – ADBCEFHG

B → CFH   (BD) – BDCFHEGA
                       CD – CD X
E → A   (DE) = DEABCCFH
                       (DF) = DFEGABCH
F → EG   DG = DG X
                       DH = X

CDG – DCG X
DGH = X
DCH = G X
CGH = X
CDGH = X

AD BD DE ~~DA~~

BCNF ✗

3NF → $CH \to G$ G is not prime ✗

2NF → $A \to BC$ C is partly dependent on A ✗

1NF ✓

(x) R(ABCD)

$A \to B$ A ~~essential~~

$B \to C$ $(A^+) = ABCD$

$C \to BD$ c.k.

BCNF = ✗

3NF = ✗ $B \to C$

~~2NF~~ = ✓

if ~~candidate key~~ is ~~single~~ it is always in 2NF ~~size of c.k. is one~~

(6) R (A B C D E F)

AB → CD

CD → EF

BC → DEF

D → B

CE → F

A essential

(AB) = ABCDEF ✓

AC = X

(AD) = ✓

AE = X

AF = X

ACE = X

ACF = X

AEF = X

ACEF = F X

(AB) (AD)

BCNF X

3NF X

2NF ✓

AB → EF non prime

car reg, hire date, car no, cust

make, model, outlet No, outlet loc.

Decomposition.

(*) R(Car_Reg, hiredate, Car no, cust. make, model, outlet No, outlet loc)

Car Reg, Hire date essential.
C.K.

R(Car Reg hiredate.

BCNF — X
3NF — X
2NF = X
1NF

⇒ 2NF

R1 (Car Reg, hiredate, carno, cust name)

~~R2 (Carno, cust~~

R2 (Car Reg, model, outletno
make outletloc)

⇒ 3NF

R11 (Car Reg, hiredate Car no)

R12 (Car no, cust no)

R21 (Car Reg, model, outlet no)

R22 (model, make) R(3 (outletno, outletloc)

Page
Date

(X) BCNF already in BCNF

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

$(AB)^+$ — C D E A B G H I J

2NF

R_1 (A , B , C)

R2 (A , D , E , I , J)

R_3 (B F G H)

3NF

R_1 (A , B , C)

R_{21} (A , D , E)

R_{22} (D , I , J)

R_{31} (B , F)

R 32 (F , G , H)

also in BCNF

Page
Date

Closure set of Attributes

for normalization

R (ABC) 3 attributes

A → B
B → C

A → BC

(A^+) → if we can identify some attribute from A

(A^+) → ABC

Attribute set A

set of Attributes which can be functionally determined from it denoted by F^+

R(ABCDEF)
A → B
C → DE
AC → F
D → AF
E → CF

$(A)^+ = AB$

$(DE)^+ = DEAFBCF$
ABCDEF

Page
Date

1 ~~Sound~~ knowledge
Famous
2 Correction

R(ABC)
$A \to B$
$B \to C$

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

(1) R(ABCDEFG)

$A \to B$
$BC \to DE$
$AEG \to G$

$AC^+ = [A, B, C, D, E]$

(2) R(ABCDE)

$A \to BC$ $\quad$ $B^+ = \{B, D,$
$CD \to E$
$B \to D$
$E \to A$

Page
Date

R (A B C D E F)

$AB \to C$
$BC \to AD$
$D \to E$
$CF \to B$

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

Equivalency

$X : P \to Q$
$Q \to R$
$R \to S$

$Y : P \to QR$
$R \to S$

$P^+ = PQRS$
$Q^+ = Q$
$R^+ = RS$

$X \not\subseteq Y$

$X \neq Y$

$(P)^+ = PQRS$

$(R)^+ = RS$

$Y \subseteq X$

F | G

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

$A \to BC$
$B \to A$
$C \to A$

$F \subseteq G$

$A^+ = ABC$
$B^+ = ABC$
$C^+ = ABC$

$A^+ = ABC$
$B^+ = CAB$
$C^+ = CAB$

$G \subseteq A$

$F \subseteq G \implies F = G$
$G \subseteq B$

Equivalent functional dependency

⇒ Canonical form / Irreducible Form

(*) Redundant or Extraneous Attribute

Attribute A of FD is said to be redundant if we can remove it without changing the closure of set of FDs.

(*) A canonical cover F_c for F is a set of dependencies in F_c and F_c logically implies implies all dependencies in F.

- Contains no extraneous attribute.
- Each left side of functional dependency should be unique
  i.e. there are no two dependencies $\alpha_1 \to B_1$ and $\alpha_2 \to B_2$ in F_c such that $\alpha_1 = \alpha_2$

- $A \to BC$
  $B \to C$
- $A \to B$
  $AB \to C$

① Decompose

$\boxed{A \to B}$ redundant
$A \to C$
$B \to C$ essential
$A \to B$
~~AB~~ $C \to AB$ essential

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

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

$C^+ = \{AB\ C\}$

② $A \to C$
$B \to C$
$AB \to C$

Canonical Cover

classmate
Date
Page

fd.

(1) A functional dependency FD is on relational schema R is Fd: $\alpha \to B$ where $\alpha \subseteq R$, $B \subseteq R$, is

if $t_1[\alpha] = t_2[\alpha]$ then $\Rightarrow$

$t_1[B] = t_2[B]$

(2) Set of all FDs or attributes that can be derived from given set of FDs. denoted by F^+

(3) Sk - Set of all the attributes by that you can find out all other attributes

(4) Ck. = minimal set of Sk
key whose
Ck is a proper subset should not be a super key

(5) C. Equivalence of FDs.

if F: $a \to b$ & g: $a \to b$
then if $f \subseteq g$ $g \subseteq f$
$\Rightarrow f = g$
is called equivalence

(6) Canonical cover Fc

Closure Set of Attributes F^+

is set of all possible FDs derived from given set FDs.

F^+

R (A B C)

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

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

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

$C^+ = \{C\}$

Ex R (A B C D E)

$A \rightarrow B$
$B \rightarrow CD$
$C \rightarrow D$
$EF \rightarrow A$

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

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

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

$E^+ = \{E\}$

$F^+ = \{F\}$

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

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

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

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

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

$(AC) = \{A, B, C\}$

(8) $A \to BC$ $\qquad R(A, B, C, D)$

$B \to CD$.

$C \to D$

	S.K.	C.K.
$D^+ = \{D\}$	✗	✗
$A^+ = \{A, B, C, D\}$	✓	✓
$B^+ = \{B, C, D\}$	✗	✗
$C^+ = \{C, D\}$	✗	✗
$(AB)^+ = \{A, B, C, D\}$	✓	✗
$(AC)^+ = \{B, C, D\}$	✓	✗
$(AD)^+ = \{A, B, C, D\}$	✓	✗
$(ABC)^+ = \{A, B, C, D\}$	✓	✗
$(BCD)^+ = \{B, C, D\}$		
$(CDA)^+ = \{C, D, B, A\}$	✓	✗
$(ABCD)^+ = \{A, B, C, D\}$	✓	✗
$(BC)^+ = \{B, C, D\}$	✗	✗
$(BD)^+ = \{B, C, D\}$	✗	✗
$(CD)^+ = \{C, D\}$	✗	✗
$(ABD)^+ = \{A, B, C, D\}$	✓	✗

~~ABCD~~

S_k

Super key :
By which you can identify any row uniquely

or

Set of all the attributes by which you can find out all other attribute

Ex: R(A, B, C)

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

$A^+ = \{A, B, C\}$ = Sk
$B^+ = \{B, C\}$ ×
$C^+ = \{C\}$ ×
$(AB)^+ = \{A, B, C\}$ = Sk
$(BC)^+ = \{B, C\}$ = ×
$(AC)^+ = \{A, B, C\}$ = Sk
$(ABC)^+ = \{A, B, C\}$ = Sk

R(A, B, C, D)

$A \to B$. $D^+ = \{D\}$
$B \to C$
$C \to A$

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

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

Sk Ck

Candidate key

C.K

minimal set of super key or no proper subset of super key is Ck.

Primary :- Ck selected by DBA is Pk.

$(BD)^+ = \{A, B, C, D\}$ Sk. Ck.

$(CD)^+ = \{A, B, C, D\}$ Sk Ck

$(ABD)^+ = \{A, B, C, D\}$ Sk X

$(BCD)^+ = \{A, B, C, D\}$ Sk X

Equivalence functional dependencies

if $f \subseteq g$ &

$g \subseteq f \Rightarrow$

$f = g$.

$f : A \to B$

$g : A \to B$

① R

f:

$A \to B$

$B \to C$

$C \to A$

g:

$A \to BC$

$B \to AC$

$C \to AB$

$f \subseteq g$

$g \subseteq f$

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

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

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

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

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

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

$f = g$. hence equivalence holds.

②

$A \to B$

$B \to C$

$C \to A$

$A \to BC$

$B \to A$

$C \to A$

$f \not\subseteq g$.

$g \subseteq f$.

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

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

$C = \{A, C\}$ C

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

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

$C = \{a, b, c\}$

R(A, B, C, D, E)

Fd:
- $A \rightarrow B$
- $C \rightarrow D$
- $BD \rightarrow E$
- $AC \rightarrow E$

	SK	CK
$(A^+) = \{A, B\}$	✗	✗
$(B^+) = \{B\}$	✗	✗
$(C^+) = \{C, D\}$	✗	✗
$(D^+) = \{D\}$	✗	✗
$(E^+) = \{E\}$	✗	✗
$(AB)^+ = \{A, B\}$	✗	✗
$(AC)^+ = \{A, B, C, E\}$	✗	✗
$(AD)^+ = \{A, D\}$	✗	✗
$(AE)^+ = \{A, E\}$	✗	✗
$(BC)^+ = \{A, B, C, D, E\}$	✓	✓
$(BD)^+ = \{B, D, E\}$	✗	✗
$(BE)^+ = \{B, E\}$	✗	✗
$(ABC)^+ = \{A, B, C, D, E\}$	✓	✗
$(ACD)^+ = \{A, B, C, D, E\}$	✓	✗
$(A, D, E)^+ = \{A, B, C, D, E\}$	✓	✗

$(BCD)^+ = \{B, C, D, E\}$ ✗ ✗

$(BDE)^+ = \{B, D, E\}$ ✗ ✗

$(CDE)^+ = \{C, D, E\}$ ✗ ✗

$(A, B, D)^+ = \{A, B, D, E\}$ ✗ ✗

$(A, B, E)^+ = \{A, B, E\}$ ✗ ✗

$(A, B, C, D)^+ = \{A, B, C, D, E\}$ ✓ ✗

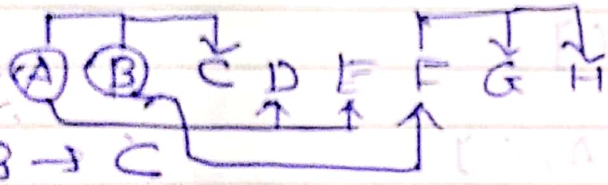
$(A, C, D, E)^+ = \{A, C, D, E, B\}$ ✓ ✗

$(A, B, D, E)^+ = \{A, B, D, E\}$ ✗ ✗

$(BCDE) = \{B, C, D, E\}$ ✗ ✗

$(ABCDE)^+ = \{A, B, C, D, E\}$ ✓ ✗

Ex



$AB \to C$

$A \to DE$

$B \to F$

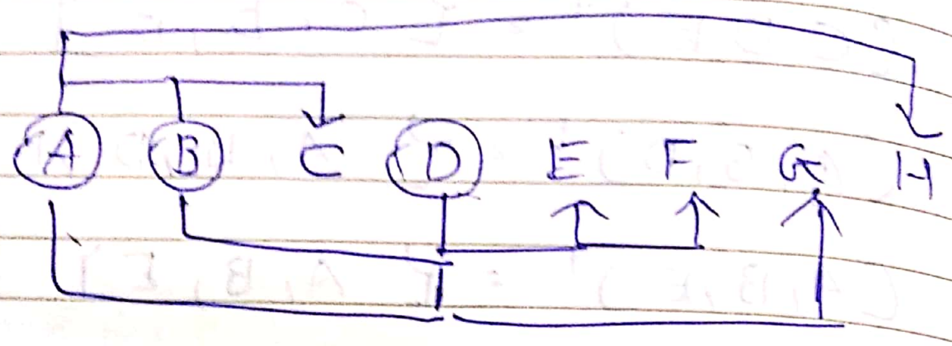
$F \to GH$

AB Donot have any incoming edges.
i.e AB are essential

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

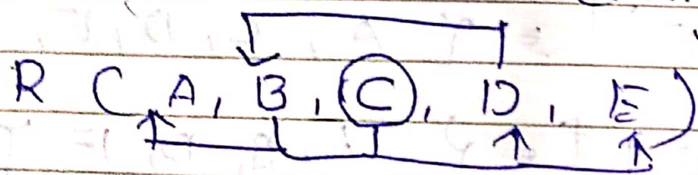
C.K ✓

② AD → C
BD → EF
AD → G
A → H



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

③ R (A, B, C, D, E)



BC → ADE
D → B

C is essential.

$C^+ = \{C\}$

$\{AC\} = \{A, C\}$

$\{B, C\}^+ = \{A, B, C, D, E\}$ ← C.K ✓
BC is CK.

4) R (A (B) C D E)

AB → CD
D → A
BC → DE

B is essential.

$B^+ = \{B\}$

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

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

$(BD)^+ = \{A, B, C, D, E\}$ CK.

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

5) R (w x y z)

z → w
y → xz
wx → y

$w^+ = \{w\}$

$x^+ = \{x\}$

$y^+ = \{w, x, y, z\}$ → CK.

$z^+ = \{z, w\}$

$\{x, w\}^+ = \{x, y, z, w\}$ CK.

$(x z)^+ = \{x, y, z, w\}$ CK

$(y, z)^+ = \{w, x, y, z\}$ X

SK.

$wz = \{w, z\}$

Lossless Join

OR

Non additive decomposition

→ Had Mandatory property must always holds good.

→ If Relation R is decomposed into two relations R_1 & R_2 then it will be lossless iff

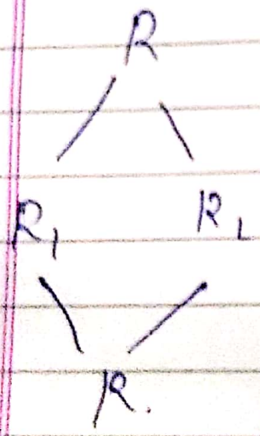
① $attr(R_1) \cup attr(R_2) = attr(R)$

② $attr(R_1) \cap attr(R_2) \neq \phi$

③ $attr(R_1) \cap attr(R_2) \rightarrow attr(R_1)$

OR

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



A	B	C
1	a	p
2	b	q
3	a	r

A	B
1	a
2	b
3	a

B	C
a	p
b	q
a	r

Date
Page

Ex

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

① R_1 (AB) R_2 (CD) ✗

② R_1 (ABC) R_2 (DE) ✗

③ R_1 (ABC) R_2 (CDE) ✗

④ R_1 (ABCD) R_2 (ACDE) ✓

⑤ R_1 (ABCD) R_2 (DE) ✓

⑥ R_1 (ABC) R_2 (BCD) R_2 (DE)
✓

$R_{(12)}$ (ABCD) R_2 (DE)

✓

Dependency Preservation

if a table R having FD set F is decomposed into table R1 & R2 having FD set F1 & F2 then

$F_1 \subseteq F^+$

$F_2 \subseteq F^+$

& $(F_1 \cup F_2)^+ = F^+$

R (A B C)

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

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

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

R_1 (A B) R_2 (B C)

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

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

(A) =

~~$F_1 \cup F_2 \rightarrow \{A, B, C\}$~~

R (A B C D)

$AB \rightarrow CD$

$D \rightarrow A$

R_1 (A D), R_2 (B C D)

$A^+ = \{A\}$ $\{B^+\} = \{B\}$

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

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

~~$F_1 = A, D$~~ ~~$F_2 = A, B, C, D$~~

4NF :- It should be in BCNF & No MVD.

MVD :-

A →→ B is, for single value A more than one value of B exist

Table should have at least 3 c

Columns B & C should be independent

	Sid	Course	Hobby
1	1	Science	Cricket
1	1	Maths.	Hockey
1	2	C #	Cricket
1	2	PHP	Hockey

(S-id , Course) (Sid → hobby)

Date
Page

A table can have both FD & MVD

S_id address Course Hobby

S_id → address } FD.
S_id →→ course } MVD
S_id →→ hobby }

S_id, course
S_id, hobby
S_id, address

⑤ 5th NF (PJNF)
should be in 4NF
it should not have join dependency

Supplier	Product	Customer
ACME	72XSW	FORD.

Supplier	Product
ACME	72 XSW

Supplier	Cust
ACME	FORD

Cust	Product
Ford	72 SXW

ACME sells 72X to Ford

Date ____
Page ____

Ex:

Prod_ID → Description

(OrderNo, Product_ID, Description)

partial Dependency

Not in 2NF ⇒ in 1NF

Ex:- Part_ID → Description

Part_ID → Price.

Part_ID, Comp_ID → Part_No

Ex

Stud_ID, SName, Course_ID, Units

R_1 (Stud_ID, SNAME)

R_2 (Stud_ID, Course ID, Units)

2NF

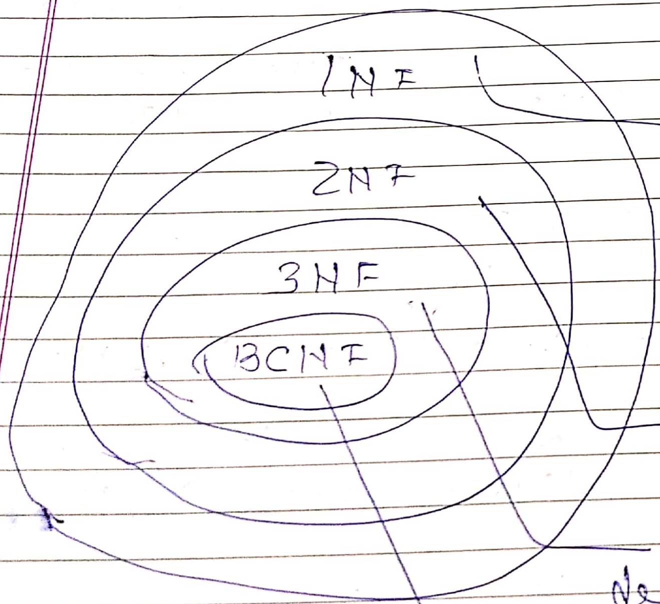
R_1 (Stud_ID, SNAME)

R_2 (Stud_ID, Course_ID)

R_3 (Course_ID, Units)

Ex:- EmpID → FNAME, LNAME, Dept

Dept ID → Dept Name



No Repetitative groups. No multivalued Attribute.

~~Primary key~~ all the attributes should be identified by key

No partial Dependency

$\alpha \to \beta$

$P \to N.P.$

No Transitive dependency

$\alpha \to \beta$

$Np \to Np$

Every determinant is a Candidate key/ S.k

$\alpha \to \beta$

S.K / CK

for all dependencies

Normalization

→ Normalization spilits a large size of table into smaller tables to to define relationships between them.

→ Database Normalization Rules:

(1) 1NF:-

→ To be in 1NF any Relation should have no repeating groups i.e. no multivalued attributes are allowed.

→ Every attribute should have atomic value.

→ If Relation schema is defined then it is already in 1NF.

Ex:

PROJ_NUM	PROJ_NAME	EMP_NAME
11	XYZ	EMP-101 EMP-102 EMP-103
12	ABC	EMP-114 EMP-115
13	UVW	EMP-121 EMP-122

→ The above table is not in 1 NF as there multivalued attributes

eg.

Scanned with CamScanner

Date ______
Page ______

(1) Normalization of Table (Relation schema)

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

$(AB)^+ = R$

$AB \rightarrow C$

$A \rightarrow DE$

$B \rightarrow F$

$D \rightarrow IJ$

$F \rightarrow GH$

2NF

R_1 (A B C)

R_2 (A D E I J)

R_3 (B, F, G, H)

3NF

R_1 (A B C)

R_{21} (A D E)

R_{22} (D I J)

R_{31} (B F)

R_{32} (F G H)

Date
Page

Ex Normalize the given table

Order No	Order Date	Item lines		
		Item code	Quantity	Price/Unit
1456	26-12-99	3687	52	50
		4627	38	60
		3214	20	20
1886	04-03-99	4629	45	20
		4627	30	60
1788	4-4-99	4627	40	60

~~Order~~ (Order No → order date)

~~Order~~ Order No, Itemcode → Quantity, Price

→ $(\text{Order No}, \text{ItemCode})^+ = R$

2NF

Order No → order date

P. D.

Decomposition

R_1 (OrderNo, Itemcode, Quantity, Price

R_2 (Order No, Order Date)

Scanned with CamScanner

3NF

R1 (OrderNo, Itemcode, Quantity)

R2 (OrderNo, orderDate)

R3 (Itemcode, Price)

Ex Proj_Num, Proj Name, Emp_Num, Emp_Name, Job_class, CHG_Hour, Hours

1) Proj_Num → Proj_Name.
2) ~~Proj_Num~~, EmpNo → ~~Proj Name~~, EmpName, jobclass, CHGHour, ~~Hours~~.
3) EmpNo → EMPName, JobClass, CHGHour
4) Job_class → Charge_hour

Page

Ex: ~~stud_ID~~ → ~~SNAME~~

~~course_ID~~ → Units.

Ex-1

ISBN → title
ISBN → pub.
pub → Address

R (I , T , P , A)

$(I)^+ = R$ C.K.

No, P.D. ⇒ in 2NF ✓

P → A
N.P N.P ⇒ not in 3NF

Date ____
Page ____

Ex:

Prod_ID → Description

(OrderNo, Product_ID, Description)

partial . Dependency

Not in 2NF ⇒ in 1NF

Ex:- Part_ID → Description
Part_ID → Price.
Part, ID, Comp_ID → Part_No

Ex: Stud_ID, SName, Course_ID, Units

R1 (Stud_ID, SNAME)

R2 (Stud_ID, Course ID, Units)

2NF

R1 (Stud_ID, SNAME)

R2 (Stud_ID, Course_ID)

R3 (Course_ID, Units)