Name:- Vinay Kumar Gupta.

Subject:- DBMS

Mob:- 7706022118

lecture -1 Date:- 26 - Sept - 2020

Content :-

① Integrity constraints & ER model } 2 marks

② Normalization } (2-4) marks

③ Queries — R.A, SQL, TRC } 4 Marks

④ File organizasation & Indexing } (2-4) marks

⑤ Transaction & concurrency control } (2-4) marks

Strategy

mandatory { ① Notes Revise & Practice
② WB/ Previous gate }

③ solve Text Book exercise problem. — DBMS → By Raghurama Krishan; DBMS → Navathe.

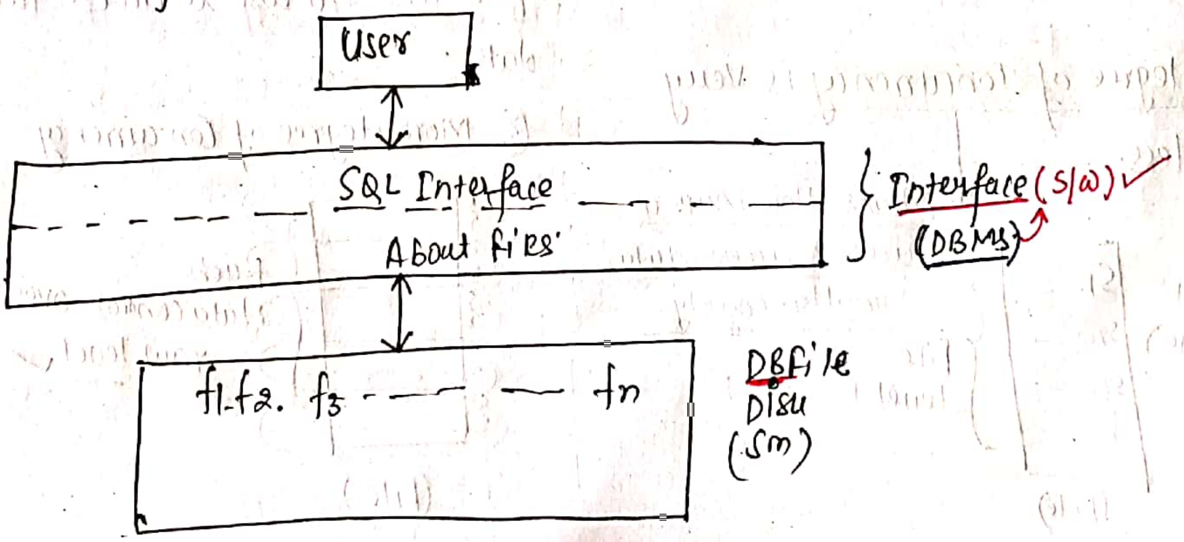
DBMS :- Intro.

database is collection of related data.

Ex self std. Information.

DBMS is a s/w used to manage & access data in more efficient way.

DBMS is Interface b/w user & D.B files.

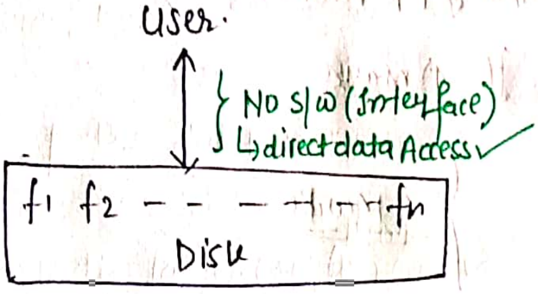


Scanned with CamScanner

FLAT File system:- (O.S Files)

DB files manages by user without DBMS S/W.

InFLATFile



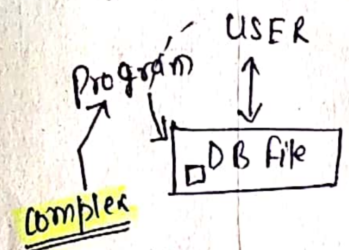
- It is useful when Database is small.

For Huge dB ← It create problem.
↳ flat file system fail to manage.

Limitations of flat file system Vs advantage of DBMS file system

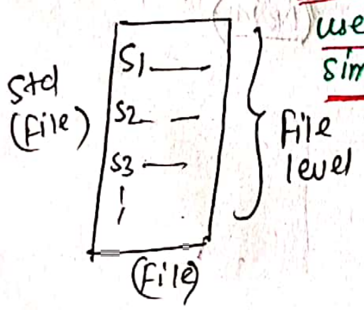
Limitation

① Too complex to manage application Program to access data. (Complex for manage the data)



② Take More I/O cost (Access Cost) to access data from DB Files

③ degree of Concurrency is Very less.
↓ Means How many user can access data simultaneously.

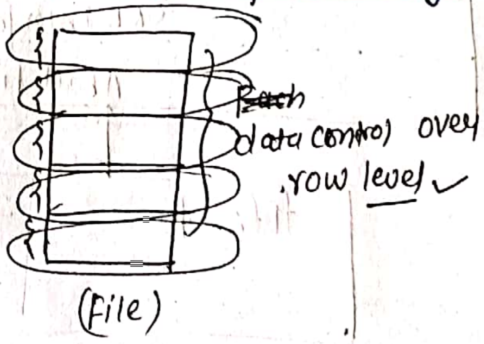


:- DBMS File system :-

① DBMS supports data independency (user can access the data using SQL interface without knowing storage Information of DB files)
Bcoz of data independency easy to develop application (SQL Query)

② Bcoz of Indexing to database file less I/O cost to access the data.

③ More degree of Concurrency



④ To complex to maintain different levels of Access control security.	④ because of VIEW's (Virtual table) Easy to manage access control.
⑤ To complex to maintain Non-redundant data.	⑤ BCZ of Normalizasation of data Easy to maintain Non-redundent data.

Integrity Constraints:-

→ RDBMS (Relational DBMS)

→ widely used data model proposed by "Codd" [Also called Codds datamodel]

→ Codd proposed 12 rules for design of DBMS s/w. (one dozen)
[These rules also called RDBMS Guidelines].

Codd Rule (RDBMS Guideline)

① Data in dbMs file must be in Tabular formate // data reside in Table
[collection of Rows & columns]

② No two row's of data files must be same. // No duplication

Relational schema

Std

Attribute [Field]

Sid	SName	DOB
S1	A	1990
S2	A	1992
S3	B	1988
S4	C	1992
S5	B	1990

Record / Tuple.

Relational Instance [Snapshot]

(RDBMS Files)

~~Relat~~. Set of Record of the Relational table called Relational Instance.

Relational Schema :- Definition / structure of data Base table

Ex Std (Sid, Sname, DOB)

Arity :- # of fields of the database Table.

(or)

attribute in the Table. (or) (# Column ~~# row~~)

Ex in Previous Table → $\boxed{\text{Arity} = \#\text{attribute} = 3}$

Cardinality :- # of Records of the database Table.

or

(# row) → Cardinality

Cardinality = # Row

Storage Look in FLAT File :-

S1, A, 1990 #, S2, A, 1992

Flat File

Candidate Key :-

Minimal set of attributes which can differentiate the records uniquely.

Put underline in below candidate key

Ex Std (Sid, Sname DOB)

unique so it is candidate key.

S1	A	1990
~~S1~~	~~A~~	~~1990~~

→ Not allow.

~~Sid Sname~~

(Sid Sname) → unique but it is Not Candidate key Because it is Not minimal.

① Find Candidate key for

Enroll (sid cid fee)

sid	cid	fee
s1	c1	–
s7	c2	–
s2	c2	–

(sid cid) is c.d key.

② ~~F.ca~~ Find Candidate key for :-

A	B	C
5	4	8
5	4	9
5	6	8
5	6	9
6	4	8

Sol^n ~~A~~, ~~B~~, ~~C~~, ~~AB~~, ~~BC~~, ~~AC~~, ABC ✓ ← c.d key

• In a Table their can be Exist More than one c.d Key.

③ Emp (eid , ename, DOB, PPNO, ~~Bank~~ Acc.No. , ifsc, Pan)

↳ c.d key ① eid
② PPNO
③ ~~Ban~~ ACC No ifsc
④ PAN

⇒ Emp (eid ename DOB PPNO Acc ifsc PAN)

Primary key

P. Key is any one C. Key whose field value is Not Null.

According to RDBMS NULL denote unknown value unexisted value

• In any RDBMS table at most One. P. Key is allowed.

Alternative key:-

Alternative key is all the C.d key of Relation schema Except P.key. called Alternate key.

***

Candidate key = Primary key + Alternate key

one or many can Exist — Candidate key: minimal key which define the all the relation of Table.

only one exist — Primary key: No Null allow

one or more than one exist — Alternate key: Null allow

It means

① It is also C.d key

② A.key fields allowed NULL Values.

In a Table (0 or more) many Alternative Key allowed.

• CREATE TABLE EMP.

```
(eid Varchar (10) PRIMARY KEY
  ename Varchar (30)
  DOB date,
  PPno Varchar (15) UNIQUE
  Acc. NO. Integer (10),
  IFSC Varchar (6),
  PAN Varchar (8) UNIQUE NOTNULL.
  (A NO, Ifsc) UNIQUE
);
```

Simple candidate key:-

Candidate key only with one attribute called simple Candidate key. Ex Eid, PPNO, PAN etc

Compound candidate key:- Candidate key with at least 2 attribute.

Ex AccNO ifsc → compound candidate key.

Prime attribute :- Attribute which belongs to some c.d key.

Relational schema.

from previous Table

↳ Prime attributes are ⇒ {eid, PPNO, ACCNO, ifsc, PAN}

Prime attribute set.

Attribute which is not belong to Any ~~can~~ candidate key are called Non prime attribute.

↳ from previous Table

Set of Non prime attribute ⇒ {ename, DOB} ✓

Lecture -2 Date

Superkey :- (used for RDBMS design)

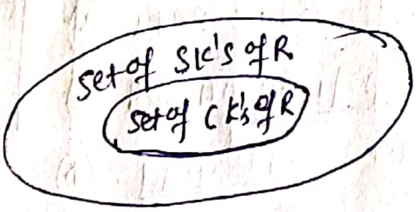
$\boxed{\text{Set of attributes which can differenciate Records uniquely.}}$

Stud (sid sname DOB)

Candidate Key : sid

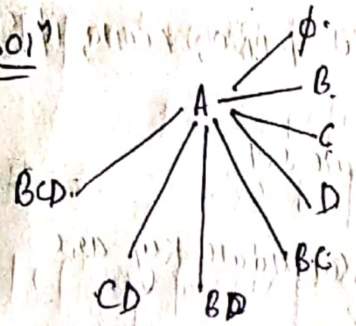
minimal super key.

Super keys {
① sid
② sid sname
③ sid Dob
④ sid snameDob
}



Que How many super keys are possible if —
R(A, B, C, D) && candidate key {A} ??

Sol^n



→ 8 super keys Possible

So → $\boxed{\text{Superkey} = \text{c.d key} \cdot [\text{Any subset other attribute}]}$

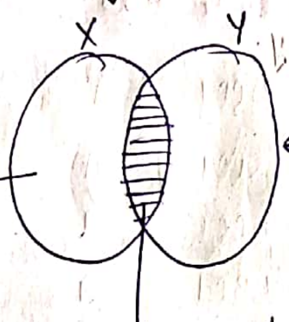
must in all s.keys.

Que:- How many SK's if $R(A_1 A_2 A_3 \text{-------} A_n)$ with c.d key $\{A_1\}$

Sol^n:- $\boxed{\# \text{ s.k's} \Rightarrow 2^{n-1}}$

Que:- How many sk's possible if Relation R(A BCDE) & cd keys ⇒ {A, BC}

Sol^n



S.K Because of cd key A → 2^4 → 16

SK Because of CK BC → 2^3 ⇒ 8

S.K b/w of cd ABC → 2^2 ⇒ 4

\# superkey = 16+8−4 ⇒ 20

Que How many SK's in $R(A_1, A_2, \dots, A_n)$ with $\{A_1A_2,\ A_2A_3\}$ cand. keys?

Sol^n

$A_1A_2 \longrightarrow 2^{(n-2)}$

$A_2A_3 \longrightarrow 2^{(n-2)}$

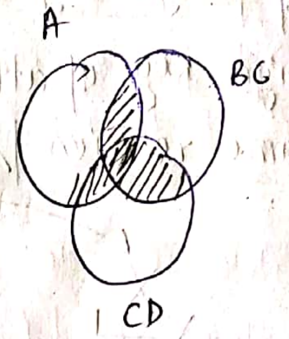
$A_1A_2A_3 \longrightarrow 2^{(n-3)}$

$\longrightarrow$ # SK's $\Rightarrow 2^{n-2} + 2^{n-2} - 2^{n-3}$

$\Rightarrow (2^{n-1} - 2^{n-3})$

$\Rightarrow 2^{n-3}[3] \Rightarrow 3 \cdot 2^{(n-3)}$ ✓

Que How many possible super keys in relation $R(ABCDEF)$ with Candidate Keys $\{A, BC, CD\}$.



Sol^n # SK $\Rightarrow 2^5 + 2^4 + 2^4 - 2^3 - 2^3 - 2^3 + 2^2$

$\Rightarrow 32 + 16 + 16 - 16 - 8 + 4$

$\Rightarrow$ (44) SK's ✓

***** Que** How many super key's Possible if ~~Cond~~ Relational schema $R(A_1, A_2, \dots, A_n)$?

Sol^n It indirectly says that max Possible super keys Possible ✓

It is possible when each individual is Cd key.

Ex for three variable $R(ABC)$. $\{A\ B\ C\}$.

{ A
B
C
AB
AC
BC
ABC } SK's

$\longrightarrow$ # super keys $\rightarrow (2^3 - 1)$

So, max No. of possible SK's for n Variable $\Rightarrow (2^n - 1)$

Foreign key :- (referencial key)

→ Used to relate b/w Tables (relationship)

→ Defined over two relations.

- → Referenced relation
- → Referencing Relation → Foreign key having Table ✓

Defn of foreign key:-

Foreign key of attributes references to primary Key / Alternative Key of same relation / some other relation.

X ← Y

X: PK/AK → Referenced attributes

Y: Foreign key Attributes. → Referencing Relation

Example :- 1

Stud (sid sname DOB) Enroll (sid cid fee) — sid in Enroll is FK

one to many (1:M)

Referenced Relation:

sid	sname	DOB
S1	A	1990
S2	A	1992
S3	B	1990
S4	B	1989
S5	A	1990

Referencing Relation:

sid	cid	fee
S1	C1	5000
S1	C2	4000
S2	C2	4000
S4	C1	4000
S4	C3	5000
~~S10~~	~~C7~~	~~4000~~

Not allowed

Example 2 :-

Emp (eid, ename, Sup ID) → FK

eid	ename	Sup ID
e1	A	Null
e2	B	e1
e3	B	e1
e4	A	e2
e5	C	Null
e6	B	e5

→ Referencing & Referenced relation within One Table only.

Ex

R (A B C) — A: PK, B: AK

A	B	C
a1	2	5
a2	4	6
a3	6	10
a4	Null	12
a5	8	6
a6	Null	12

Referenced Relⁿ

S (D E) — D: FK, E: PK

D	E
2	e1
2	e2
Null	e3
4	e4
8	e5

Referencing Relⁿ ✓

⇒ foreign key attributes allowed Null

⇒ Referencing ~~reco~~ relation record whose foreign key Value Null is Not related to any Referenced Relation Record.

⇒ b/w Referenced : Referencing.

mapping. Referenced (1) ⟶ Referencing (many)

Referencing (many) ⟶ Referenced (1)

1:m → Referencing relation ✓

Referenced Relation ✓

Always 1:m } only
1:1 } (or)

(Null) Relation

Foreign Key Integrity constraints :- (Referencial Intigrity constraints)

(I) Referenced Relation (Std)

(1) Insertion :-

No foreign Key Voilation

(2) Deletion: (May cause Voilation)

default → a) On delete ~~Cascade~~ No action

· deleting of referenced Relⁿ record is restricted if foreign key Voilation occure.

b) on delete Cascade :-

on deletion of Referenced Relation record force to delete all related referencing records.

(c) on delete ~~No~~ set Null·

If foreign Key Field allow to set Null value then set Null in related foreign Key Value

(3) updation (may cause Voilation)

(a) ON update NO Action (default)

(b) ON update Cascade

(c) ON update set Null·

(II) Refencing Relation · (Enroll)

(1) Insertion (may cause foreign key voilation)

Insertion is restricted if foreign key Voilation occurs·

(2) deletion

No Voilation

(3) updation (May cause foreign key voilation)

updation is restricted if foreign key Voilation occurs.

Lecture-3

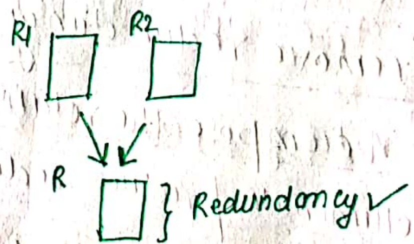
Topics

1. Introduction to DB anamalies.
2. Functional Dependencies
3. Attribute closure
4. Finding candidate keys
5. Lossless Join & Dependency preserving decomposition.
6. Normal forms
   - 1NF, 2NF, 3NF, BCNF
   - Finding Highest NF of Relational schema
   - Decomposition into Heigher Normal forms.
   - $\boxed{\text{MVD \& 4NF}}$
     - ↳ multivalued decomposition.
7. canonical cover & FDset (minimal cover)

Normalizasation :-

→ used to elliminate/reduce redundancy in DB table

→ If two or more independent rel^n stored in single relation Then forms redundancy.



① sid ⟶ sname DOB

② Cid ⟶ Cname Instructor

③ sid Cid ⟶ fee

R

Sid	Sname	DOB	Cid	Cname	Instr	Fee
S1	A	1990	C1	DB	Korth	—
S2	A	1990	C1	DB	Korth	—
S3	B	1988	C1	DB	Korth	—
S3	B	1988	C2	Algo	Coreman	—
S3	B	1988	C3	O.S	Galvin	—

Sid Cid is candidate key

Redundancy

** Problems because of Redundancy :- [DB anamalies]

(I) Insertion Anamoly (Problem) :- In above table if we Insert Student Information then it is Necessary that Cource Information also Insert (or) If we Insert Cource Information then Student Information also Insert

(II) deletion Anamoly :-
If we delete on Row of student B then all the Information Related to that student must delete.

(III) updation Anamoly :-
If we update S3 DOB from (1988 → 1990) then it say must update on all the S3 DOB (all rows of S3)

These problems causes Inconsistency in database.

If 0% redundancy in DB Table ⟺ No dataBase Anomalies

To remove redundancy

Decompose the relation R into sub Relation R1, R2, --- Rn to ellimminate / reduce redundancy.

Ex

Normalized DB design

Sid → Sname DOB

Sid	Sname	DOB
S1	—	—
S2	—	—
S3	—	—

Sid Cid → Fee

Sid (FK)	Cid (FK)	fee
S1	C1	—
S2	C1	—
S3	C1	—
S3	C2	—
S3	C3	—

Cid → Cname Instr

Cid	Cname	Instr
C1	—	—
C2	—	—
C3	—	—

Scanned with CamScanner

Functional dependency :-

$(x \rightarrow y)$

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

~~then~~ if $x \rightarrow y$ then F.D Implies (member of) Relⁿ. R.

If $\boxed{t_1.x = t_2.x}$ Then $\boxed{t_1.y = t_2.y}$ ✓

(means for same I/P) → (O/P will be same)

R

X	Y	ooo
x4	y5	
x1	y5	
x2	y2	
x3	y2	
x4	y4	
x2	y2	
x1	y5	

⇒ **$(X \rightarrow Y)$ FD. But $\boxed{Y \rightarrow X}$ is not FD's ✓

Ex $\boxed{Sid \xrightarrow{Implied} Sname}$ → says $\boxed{\text{Sid determine Sname}}$ ✓

then stud

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S1	A	C3
S2	B	C3
S2	B	C4
S3	C	C1
S4	C	C2
S5	D	C3

→ FD Table ✓

→ Just like Fⁿ ~~dep~~ $\boxed{\text{for same I/P, O/P is same}}$ ✓

** $\boxed{\text{for different I/P, O/P may or may be same}}$ ✓

Types of F^n dependency

1. Trivial F^n dependency.
2. Non Trivial F^n dependency
3. Semi- Non Trivial F^n dependency.

FD's

Trivial F^n dependency

Trivial F^n dependency means No meaning of solution means solution is exist always without saying anything

$(x \to y)$ is trivial FD of R if $\boxed{x \supseteq y}$ (superset)

Ex
sid → sid
sname → sname
sid sname → sname
sid sname → sid sname
etc are Trivial

Non trivial F^n dependency

$(x \to y)$ non trivial FD if $\boxed{x \cap y = \phi}$

Ex
sid → sname
sid cid → sname
are Non Trivial FD's

Semi Trivial F^n dependency.

combination of Trivial & Non trivial

Ex $\boxed{x \to xy}$
↳ $x \to x$ ← Trivial
$x \to y$ ← Non Trivial

Ex
1. sid → sid sname
↳ sid → sid
& sid → snam ✓
2. sid cid → cid sname
sid cid → cid → Trivial
sid cid → sname → Non Trivial

Semi Trivial Functional dependencies ✓

$$\boxed{X \to XZ \Rightarrow \underbrace{X \to X}_{\text{Trivial}},\ \underbrace{X \to Z}_{\text{Non trivial (Intersection} = \phi)}}$$

Armstrong rules over FD's :-

x, y & z some attributes sets over R.

1. Reflectivity :- (Trivial FD's)
   $X \to X$ (always is in Relation R)

2. Transitivity :-
   if $(X \to Y$ & $Y \to Z)$ then $(X \to Z)$ ✓ is in R ✓

③ Augumentation :-

if $x \rightarrow y$ then $xz \rightarrow yz$
(or)
$xz \rightarrow zy$ → order does not matter.

④ Split Rule :-

if $x \rightarrow yz$ then $x \rightarrow y, x \rightarrow z$

⑤ Merge Rule (union Rule) :-

if $x \rightarrow y, x \rightarrow z$ then $x \rightarrow yz$

Attribute closure :- (x^+)

x is some attribute set of Relation R.

x^+ = { set of all attributes which can determines by x }

is called closure of x.

Given FD set of R :-

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

① $A^+ = \{A, B, E, C, D\}$ → it means $\boxed{A \rightarrow ABCDE}$

② $(AF)^+ = \{A, F, B, C, D, E, G\}$ → it means $\boxed{AF \rightarrow ABCDEFG}$

③ $(BE)^+ = \{B, E, C, D\}$ → $\boxed{BE \rightarrow BECD}$

Super Key :-

X is some attribute set of Relational schema R.

→ $\boxed{\text{X is super key of R iff } X^+ \text{ must determine all attributes of R.}}$

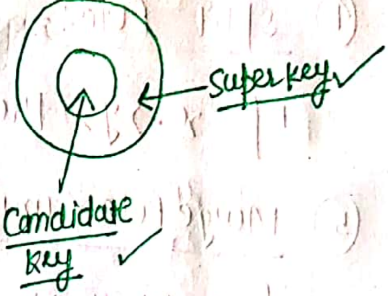
Que $R(ABCDEF) \Rightarrow \{AB \to C, C \to D, D \to E, E \to F, F \to B\}$.

Sol^n

$(AB)^+ = \{ABCDEF\}$ ✓ S.K ✓

$(A\cancel{B})^+ = \{ABCDEF\}$ ~~✗~~ ✓ S.K

$(BC)^+ = \{BCDEF\}$ ✗ Not S.K



Candidate key :- [minimal super key]

X is Candidate key of R iff

① X must be Super key of Relation R.

② & ② Every Superkey is not Candidate key [only minimal Superkey is candidate key]

③ No proper Sub set of X is super key of R.

$\forall\ Y \subset X$ such that Y^+ Not determines all attributes of R.

Que R(ABCDE)

$\{AB \to C, C \to D, B \to E\}$

Find CK of R?

Sol^n $(AB)^+ = \{A, B, C, D, E\}$ ✓

So, AB → Super key → So, It's also Cd' key (AB) ✓

✗ Not super key ✓   ✗ Not super key ✓

Imp Que

↳ $\boxed{\text{* Find all candidate Keys of R with FD set (F)}}$ ✓

** Finding CK of R with n attributes & & given FD set is NPC problem { exponential time complexity.)

↳ NP complete

↳ used in ① For direct Questions

② Find cd k's

+ Heigest Normal form.

+ decomposition

Scanned with CamScanner

Que R(ABCD)

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

Find CK's?

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

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

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

**

If Nontrivial FD $x \to \underline{y}$ with y is Prime attribute Then R has atleast two Cand Keys

** Que R(ABCDE)

$\{AB \to C, BC \to D\}$.

Find CD Keys.

Sol^n $(ABE)^+ \to \{ABCDE\}$

↑ cd Keys

*** Que R(ABC)

No Non trivial FD's then candidate key?

Sol^n It means only trivial FD's are there

So, it is possible $(ABC)^+ \to \{A, B, C\}$

↑

CD Keys

*** Que (4) R(ABCDEF)

$\{AB \to C, C \to D, CD \to BE, DE \to F, EF \to A\}$. Find all CK's?

Sol^n :- $(AB)^+ = \{A, B, C, D, E, F\}$

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

$(EFB)^+$, $(\cancel{ACB})^+$, $(EDB)^+$, $(\cancel{ECB})^+$, $(C)^+$

$\cancel{(CDE)}^+$ ↓ because (C) define all

↓ C is cd keys

(C) is cd keys

4 cd Keys

↓

$\{AB, C, BEF, BDE\}$

Que R(ABCDEF)

{AB→C, C→DE, E→F, F→B}. Find cd keys

Sol:-
(AB)+ = {ABCDEF}
(AF)+ = {AFBCDE}
(AE)+ = {ABCDE}
(AC)+ = {ACDEFB}

→ 4 CD-Keys

Que R(ABCDE)

{A→B, B→C, C→D, D→A}

Sol: Cd Keys ⇒ {AE, BE, CE, DE}

4 cd Keys

** Que:- R(ABCDEF)
***
{AB→C, C→A, BC→D, ACD→B, BE→C, EC→FA, CF→BD, D→E}.

Sol:
(AB)+ = {A, B, C, D, E, F}
~~(ABC)+ = {ABCDEF}~~
(BC)+
(ACD)+ → (CD)+
(BE)+
(BD)+
(CF)+ (EC)+
7 cd Keys Ans

Lecture -14

Membership Test :-

$X \to Y$ FD member of FD set (F).

iff X^+ must determine Y in FD set (F) $\Rightarrow$ Then it will be member

$f = \{ \text{---} \quad \text{---} \}$ $\leftarrow$ $X \to Y$

X^+ — Find yes/Not

Que Given FD set

$\{AB \to C, C \to D, D \to E, E \to F, F \to B\}$.

Which of the FD's are members of given FD set?

1) $AB \to F$
2) $AC \to B$
3) $BC \to A$

Soln

I) $(AB)^+ = \{ABCDEF\}$ ✓
II) $(AC)^+ = \{ACDEFB\}$ ✓ } Member ✓
III) $(BC)^+ = \{BCDEF\}$ ✗ $\to$ Not member ✓

Equality of FD sets :-

$F \equiv G \Rightarrow$ iff
F cover G
G cover F

F cover G $\to$ (G inside F) $\Longleftrightarrow$ G cover F :- (F inside G) then $\boxed{F \equiv G}$

F & G ~~funct~~ FD sets logically equal iff

① **F cover G** :- Every FD of G set must be member of F set.

(G inside F) $\boxed{F \supseteq G}$

② **G covers F** :- Every FD of F set must be member of G set.

(F inside G) & $\boxed{G \supseteq F}$

F covers G	G covers F	
Yes	Yes	→ F = G → Both are equal
Yes	NO	→ F ⊃ G → F covers G
NO	Yes	→ G ⊃ F → G covers F
NO	NO	→ Cannot ~~say~~ compare

can't say

Que:- $F = \{A \to B, B \to C, C \to A\}$

$G = \{A \to BC, B \to AC, BC \to A, AB \to C\}$

Which is true for given FD set

(a) $F \subset G$ (b) $F \supset G$ (c) $F = G$ (d) None.

Sol^n:-

F covers G	G covers F
$A \to B$, $B \to C$, $C \to A$ ← Test ← $A \to BC$, $B \to AC$, $BC \to A$, $AB \to C$	$\{A \to BC$, $B \to AC$, $BC \to A$, $AB \to C\}$ ← Check ← $A \to B$ ✓, $B \to C$ ✓, $C \to A$ ✗ ← Not Present

$\boxed{F \supset G \text{ But } G \not\supset F}$

Ans (b) ✓

Properties of decomposition (decomposition of Table)

① Lossless decomposition

② Dependency preserving decomposition.

① Lossless Join decomposition:-

Relational schema R with instance r decomposed into subrelations $R_1, R_2, R_3 ---- R_n$.

ⓐ In General

$R_1 \bowtie R_2 \bowtie R_3 ---- \bowtie R_n \supseteq R_n$

*ⓑ if $[R_1 \bowtie R_2 \bowtie R_3 ---- \bowtie R_n] = R$

Then lossless Join decomposition.

*ⓒ if $[R_1 \bowtie R_2 ---- \bowtie R_n] \supset R$

then Lossy Join decomposition.

Ex

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S3	B	C3

{Sid → Sname}

decomposed R into subrelations.

such as:-

R_1

Sid	Sname
S1	A
S2	B
S3	B

Key.

R_2

Sid	Cid
S1	C1
S1	C2
S2	C2
S3	C3

$\boxed{R_1 \bowtie R_2 = R}$ ← Join $R_1 \bowtie R_2$

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S3	B	C3

→ called Lossless Join decomposition.

II^{nd} decomposition

R1

Sid	Sname
S1	A
S2	B
S3	B

↑ Not Key

R2

Sname	Cid
A	C1
A	C2
B	C2
B	C3

→ $R1 \bowtie R2$

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S2	B	C3
S3	B	C2
S3	B	C3

Lossy Join ←

Spurious Tuples.
↑ Extra

$\boxed{\bowtie \rightarrow \text{Natural Join}}$

*** ∴ First decomposition is lossless Join Because of In second table their is a row ~~whic~~ sid which is a (Pkey ~~Pid~~ or C.key ~~Cid~~ of I^{st} Table). This is Not Happen in 2^{nd} Decomposition. due to that reason I^{st} decomposition was lossless & 2^{nd} decomposition was Lossy.

II^{nd} method

- Lossless Join decomposition finding (when Functional dependency set is given)

Relational schema R with FD set (F) decomposed into relations R_1, R_2.

Given Decomposition is lossless join iff.

① $(R_1 \cup R_2) = R$

&

② $(R_1 \cap R_2) \rightarrow R_1$ // $R_1 \cap R_2$ is SK of R_1

(or)

$(R_1 \cap R_2) \rightarrow R_2$ // $R_1 \cap R_2$ is SK of R_2.

Questions:-

(1) R(ABCDE) find which one is lossless & which is Lossy for given relation R.

$\{AB \to C, C \to D, B \to E\}$.

(a) decomp $\{\overset{R_1}{ABC}, \overset{R_2}{CD}\}$

Sol^n $R_1 \cup R_2 \neq R$

Lossy Join decomposition.

(b) decomp $\{\overset{R_1}{ABC}, \overset{R_2}{DE}\}$

Sol^n $R_1 \cap R_2 = \phi$

Lossy Join decomposition.

(c) decomp $\{\overset{R_1}{ABC}, \overset{R_2}{CDE}\}$

Sol^n $(R_1 \cap R_2) = C^+ = \{C, D\}$

⇑

Not SK of R_1 or R_2

Lossy Join decomposition

(d) decomp $\{ABC, CD, DE\}$

Sol^n $R_1(ABC)$, $R_2(CD)$, $R_3(CDE)$

$C^+ = CD$

$R_{12}(ABCD)$ $R_3(CDE)$

$D^+ = D$ Not SK of R_{12} or R_3 → Lossy Join

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

Sol^n is lossy Join

(4) decomp {ABC, CD, BE}

R_1 = ABC, R_2 = CD, R_3 = BE

Sol^n

ABC ⋈ CD → ABCD (C)

ABCD ⋈ BE → ABCDE ✓

It is Lossless Join {R_1 & R_2} So, Join it & Again Check with R_3 ✓

Lossless Join ✓

*** Dependency preserving decomposition :-

Relational schema R with FD set (F) decomposed into subrelations $R_1, R_2, \ldots R_n$ with FD sets $F_1, F_2, \ldots F_n$.

(a) In General

$$\boxed{\{F_1 \cup F_2 \cup F_3 \cup \ldots F_n\} \subseteq F}$$

= ← dependency Preserving ✓

⊂ ← Not dependency preserving ✓

(b) if $\{F_1 \cup F_2 \cup F_3 \cup \ldots F_n\} = F$ Then DP decomposition.

(c) if $\{F_1 \cup F_2 \cup F_3 \cup \ldots F_n\} \subset F$ Then Not Dependency Preserving decomposition

Que R(ABCDE)

{A→B, B→C, C→D, D→BE} ✓

decomposed into {AB, BC, CD, DE}.

Sol^n

R_1(AB)	R_2(BC)	R_3(CD)	R_4(DE)
A → B ✓	B → C ✓	C → D ✓	D → E ✓
	C → B ✓	D → C	
		for D → EB	
		Find D^+ = {DECB}	

{A→B, B→C, C→D, D→CE}

So, it is dependency preserving decomposition ✓

Que R(ABCD)

{AB → CD, D → A}

decomposition {ABC, BCD, AD}.

Sol^n

R1 (ABC)

{AB → C} ✓

↳ $(AB)^+ = \{ABC\}$ ✗ failed to find $\boxed{AB \to D}$

$A^+ = A$

$B^+ = B$

$C^+ = C$

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

$(BC)^+ \Rightarrow (BC)$

$(AC)^+ \Rightarrow (AC)$

R2 (BCD)

{BD → C}

$(B)^+ = B$

$(C)^+ = C$

$(D)^+ = DA$

$(BC)^+ = BCA$

$(CD)^+ = DCA$

$(BD)^+ = BDAC$

R3 (AD)

{D → A} ✓

$(A)^+ \Rightarrow A$

$(D)^+ \Rightarrow DA$

So, we find $\boxed{F_1 \cup F_2 \cup F_3 \subset F}$

↳ AB → D lost Bcoz. of decomposition.

So it is Not dependency preserving ✓

→ **Trick to Find dependency preserving :-**

Firstly find all the possible combinations of de-composed Table Then check from source relation that the data belongs from the f^n or Not if belong then ~~Kepp~~ Keep that relation as it is otherwise discard it.

At Last check that all the Relation of the source Function exist or Not. if Exist then dependency preserving otherwise Not dependency preserving.

Lecture -5 :-

Normal form:-

It's used to Identify degree of redundancy.

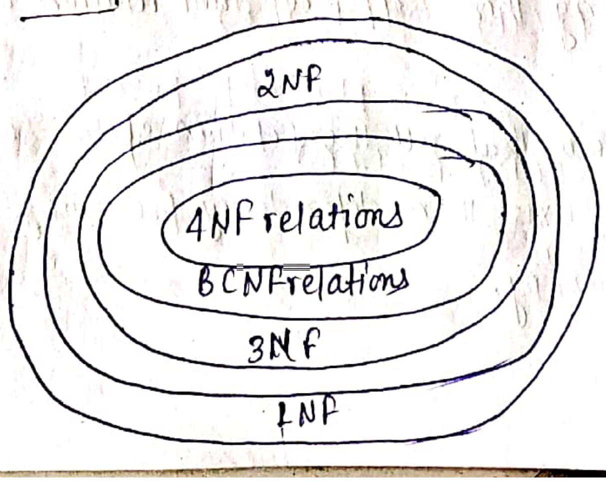
Redundancy in DB Relation.

Non-Trivial FD's	Non Trivial MVD's
$X \to Y$	$X \to\to Y$
{single valued Dependency}	{multi valued, dependency}

**

over FD set $X \to Y$

1NF
↓
2NF
↓
3NF
↓
$\boxed{\text{BCNF}}$ → 0% Redundancy over FD's
Redundancy max exist → (over MVD).
↓
$\boxed{\text{4NF}}$ → 0% Redundancy over FD's & MVD's.



$\boxed{\text{• 1NF condition is default in RDBMS Table}}$ ✓

1 NF (First Normal Form) :-

→ default Normal form of RDBMS relation. is 1NF ✓

→ Relation R is in 1NF iff No multivalued attributes in R. (Every attribute of must be atomic / single valued)

RDBMS guidelines.

⇒ R

Sid	Sname	Cid
S1	A	C1/C2
S2	B	C2/C3
S3	B	C3

MVA (multivalued attribute).

⇒ Not in 1NF so, Not in RDBMS.

→ f^n dependency ⇒ $\boxed{\text{Sid} \rightarrow \text{Sname}}$ ✓

⇓

Sid	Sname	Cid
S1	A	C1
S1	A	C2
S2	B	C2
S2	B	C3
S3	B	C3

⇒ Cid key ⇒ (Sid, Cid)

⇒ 1NF & RDBMS form ✓

Because of from source Table Cid is Not Functionally dependent so, for getting minimal superkey or SCid key we should choose (Sid, Cid) as a Candidate key.

Important Concept ✓ :-

• Any relation with two attributes is in BCNF → True

• A relation in which every key has only one attribute is in 2NF → True

• A Prime attribute can be transitively dependent on a key in 3NF relation → True

• A prime attribut can be transitively dependent on a key in a BCNF relation. → False

Que

R

Sid	Sname	Cid	Phno	email
S_1	A	c_1/c_2	$P_1/P_2/P_3$	$e_1/e_2/e_3/e_4$
		2 *	3 *	4 = 24
S_2	B	c_2/c_3	P_3/P_4	e_4/e_5
		2 *	2 *	2

FD's.

{ Sid → sname }

24 → Tuples required

For unique Identification

=) Not in 1NF

&

Not in RDBMS form

8 Tuples required.

⇓

R

Sid	Sname	Cid	Phno.	email
S_1	A	c_1	P_1	e_1
S_1	(A)	c_2	P_3	e_4
S_2	B	c_2	P_3	e_4
S_2	B	c_3	P_4	e_5

Needs 24 Tuples

Need 8 Tuples

=) 1NF + RDBMS ✓

Cd key = {Sid Cid Phno email} ✓

- If we find Cd keys for given Table then the given Table automatically change into RDBMS form.
- But In 1NF their is High degree of Redundancy found.

  To Reduce this we use (2NF, 3NF, BCNF) form.

***

$X \to Y$ FD of R forms redundancy.

iff 1] Non-trivial FD $(x \to y)$

(and)

2] "X" is not SK of R.

{ Non trivial FD $x \to y$ } Not superkey

***

$X \to Y$ FD of R not forms redundancy iff

① Trivial FD $(x \supseteq y)$

(OR)

② X : super key of R.

Ex① R(Sid Snam Cid)

Sid	Snam	Cid
S1	A	C1
S1	A	C2
S1	A	C3
S2	B	C2
S2	B	C3

[sid → Sname] → fd's

Not super keys.

Cand key : Sid Cid

→ forms redundancy.

Ex② R(eid ename DOB rating sal)

eid	ename	DOB	rating	sal
e1			8	50K
e2			8	50K
e3			8	50K
e4			10	70K
e5			8	50K

→ fd's ⇒ {Eid → Ename DOB rating}

{rating → sal}

SK

It does not Form Redundancy

Not S.K

So, it form Redundancy

Que 3

R

Sid	Sname	DOB
S1		
S2		
S3		
S4		
S5		
S6		

{Sid → Sname DOB}

SK ✓

so, No Redundancy occurs ✓

Non Prime attributes ✓

Que 4 R (A B C D E)

FD's are { AB → C
B → D
D → E
AE → F
C → A }

Prime (P-NP) Non Prime

(Non-Prime)-(Non-Prime)

forms Redundancy ✓

Non Prime

Non super keys.

combination of Prime & Non Prime

Prime

Prime

Sol^n CD Keys ⇒ {AB, BC}

Note

1. If proper subset of cd keys determine Non prime is forms Redundancy
2. Non-Prime determine other Non Prime forms Redundancy
3. Proper subset of c.d key & Non prime attribute combinly form Non SK's if this combination determine other Non Prime attribute forms Redundancy.

Scanned with CamScanner

④ Proper subset of cd keys if also determine some other Proper subset of cd keys also form Redundancy.

*** learning things

Redundancy occure when

① Proper subset of cd key → Non Prime attribute.

② $\boxed{\text{Proper subset of cd key + Non Prime att}}$ → Non Prime ~~super key~~ attribute.

↳ super keys ⊗

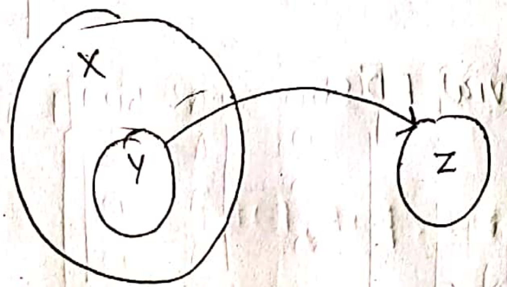
③ Non Prime Att → Non Prime Att

④ $X_1 \rightarrow X_2$, Y_1, Y_2 [Proper subset of cd key → some other Proper subset of cd keys.]

Second Normal Forms:-**

A Relational schema R is in 2NF iff No Partial dependencies in Relation R.

Partial dependencies



X is any c.Key of R

$Y \subset X$

Z is Non Prime attribute of R.

if $Y \rightarrow Z$: called Partial dependency

Third Normal Forms:-

S → P

Relational schema R is in 3NF iff every Non-Trivial FD. $X \rightarrow Y$ in R with

ⓐ $\boxed{\text{X must be superkey of R}}$

(or)

ⓑ $\boxed{\text{Y must be Prime attribute of R}}$

SK (or) $X \rightarrow Y$ → Prime.

Ex R(ABC) {$AB \rightarrow C$, $C \rightarrow A$}

S.K. → 3NF ✓ ; $C \rightarrow A$ → P.A ✓ 3NF

→ But also form Redundancy.

Sol^n C.d Keys {AB, BC}

Bcoz of

⇒ (Proper subset of C.K) ⟶ (Proper subset of other CK)

allow in 3NF & form redundancy.

BCNF (Boyce Codd NF):-

R in BCNF iff every Non-trivial FD $X \rightarrow y$ in R with X must be super key.

$X \longrightarrow y$: Non Trivial FD's.

↳ must be super key ✓

Learn:-

Restriction Increase ↓

- 1NF → if Multi Valued Row exist then Not in 1NF
- 2NF → $X, Y \rightarrow Z$ (c.d keys) called Partial dependencies
  - For ~~Prim~~ 2NF: No Partial dependencies exist
- 3NF → $\boxed{X \rightarrow Y}$ if X = super key or Y = prime attribute } → is in 3NF
- BCNF → $\boxed{X \rightarrow y}$ → X must be super keys.

Que Find Heigest Normal form of Relation R

1) R(ABCDEF)
{ABD → C, BC → D, CD → E}

2) R(ABCD)
{AB → C, BC → D}

3) R(ABCD)
{AB → C, C → A, AC → D}

Sol^n 1) R(ABCDEF)
{ABD → C, BC → D, CD → E}

1) ~~BCNF~~
2) ~~3NF~~
3) ~~2NF~~
↓
1NF

(ABD)+ ⇒ {ABDCE}.
(ABDF), (ABCF), ~~(ABEF)~~
ABEF

(A)+ = A
(B)+ = B
(D)+ = ~~AB~~ D
(AB)+ = ~~AB~~ AB
(BD)+ = BD
(AD)+ = AD
(C)+ = C
(AC)+ = AC
(BC)+ = BCDE

CD keys ⇒ {ABDF, ABCF}
PA ⇒ {A, B, C, D, F} NPA ⇒ {E}

Ans [Neither BCNF, Nor 3NF, Nor 2NF] ⇒ Partial dependency.

2NF Testing

{Proper subset of CdKey}$^+$ = {Prime attribute only} → Rel^n is in 2NF

Sol^n 2) R(ABCD)
~~[illegible]~~ {AB → C, BC → D}.

CdKeys ⇒ (AB)+ ⇒ {ABCD}   PA ⇒ {AB}, NPA = {CD}.

1) ~~BCNF~~
2) ~~3NF~~
3) 2NF

(A)+ ⇒ A
(B)+ ⇒ B
~~(AB)+ ⇒ ABCD~~

Ans 2NF

Sol^n ③

R(ABCD)

{AB → C, C → A, AC → D}

~~So~~ Cd Keys: {(AB)+ = {ABCD}

(CB)+ = {CBAD}

PA = {ABC} ✓ NPA = {D} ✓

① ~~BCNF~~

② ~~3NF~~

③ (A)+ = A

(B)+ = B

(C)+ = (ACD)

(AB)+ = (ABCD)

(BC)+ = {B, C, A, D}

(AC)+ = {A, C, D}

} → Partial dependencies ✓

↑

So ~~2NF~~

Ans 1NF

Que Find Heigest Normal form

① R(ABCDEF) {AB → C, C → DE, E → F, F → B}

② R(ABCDEF) {AB → C, C → D, D → AE, DE → F, EF → B}

③ R(ABCDE) {A → B, B → C, C → D, D → A}

④ R(ABCD) {A → B, B → AC, C → D}.

Sol^n ① R(A B C D E F)

{AB → C, C → DE, E → F, F → B}.

Sol: Cd Keys (AB)+ = {A B C D E F}

(A F)+ = {A F B, C D E}

(AE)+ = {A E F B C D}

(AC)+ = {A C D E F B}.

(~~AD~~)+ = {A D}

CD Keys → {AB, AF, AE, AC} | PA {A, B, C, E, F}, NPA {D}

BCNF → X

3NF → X

2NF → (A)+ ⇒ {A}

(B)+ ⇒ {B}

(F)+ ⇒ {F, B}

(E)+ ⇒ {E, F} → Partial dependency

(C)+ ⇒ {C, D, E, F, B}

So, Not a 2NF

1NF Ans

Sol^n ② R(A B C D E F)

{AB → C, C → D, CD → AE, DE → F, EF → B}

cd Keys → (AB)+ = {A B C D E F}

(AEF)+ = {A E F B C D}

(~~ADE~~)+ = {

(AD)+ = {D A E F B C}

(C)+ = {C D A E F B}

cd Keys ⇒ {AB, AEF, ADE, C}

BCNF → 3NF

Scanned with CamScanner

***

Note

If relational Schema R ~~wh~~ with only Prime attributes (means all element are Prime attributes) then relational schema always in 3NF but may not BCNF.

Sol^n (3) R(ABCDE)

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

Cd Keys $\Rightarrow$ $(EA)^+ = \{AEBCD\}$
$(ED)^+ = \{EDABC\}$
$(EC)^+ = \{ECDAB\}$
$(EB)^+ = \{EBCDA\}$.

PA $\Rightarrow$ {A B C D E} NPA {ϕ}.

B~~CN~~F $\to$ 3NF ✓

Ans 3NF

***

Soln (4) R(ABCD)

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

$(A)^+ = \{ABCD\}$
$(B)^+ = \{BACD\}$

CD keys $\Rightarrow$ {A, B}   PA = (A, B), NPA = {C, D}.

B~~CN~~F, 3~~NF~~, 2NF ✓

Ans 2NF

Proper subset of {A, B}

$\phi \xrightarrow{X}$ N. Prime

Not Possible ✓

Note

Cd keys $\to$ ϕ {Emptyset} ✓

$\to$ (y) Non prime

Not Possible

So, No chance of Partial dependency

Note

If all condidate key are simple (cd key with only one element) condade key then 2NF ~~gua~~ guranteed possible (Bcoz No Partial dependency Exist)

↳ kyun ki simple condade key ki hm log subset nhi bna skte so 2NF to pakka hoga ✓ kyun ki koi partial dependency nhi hogi

Lecture -(6) ✓

Que Relational schema R with No Non Trivial FD's. What is Heigest Normal form of R. ??

Sol^n When a Relational schema R with only Trivial FD's (or) No Non-Trivial FD's, then Heigest Normal form of R will be — BCNF ✓

In this situation all attribute are condade key.

• {No Non Trivial FD's in R} ⟹ {No Redundancy over FD's in R} ⟺ R in BCNF & may not in 4NF ✓

Que Relational schema R with only two attributes Rel^n R Always in —

Sol^n A Relational schema R with only two attributes Rel R Always in — BCNF ✓ (Also in 4NF) ✓

Possibility are R(AB)
(a) BCNF {A → B} → {CK ⇒ A} ✓
(b) BCNF {B → A} → {CK ⇒ B} ✓
(c) BCNF {A → B, B → A} → {CK = (A, B)} ✓
(d) BCNF {No non trivial FD's}. → {CK = {AB}} ✓

Important Points ✓

• Index key's of Index file always sorting sequence order [CON, DOC, ENG, SER, MUS] Index
↳ Ascending order of letters ✓

• A → B & A → C then A ↠ BC there fore A ↠ BC is valid

• If A → BC & A → B then A → C there fore A ↠ C is valid

But ✓

• A ↠ B & A ↠ C then A → BC ⇒ Not valid

• A ↠ BC & A → B then A → C ⇒ Not valid

Means Result A → C 𝑡𝑜 A ↠ C valid hai Pr vice versa NOT

Decomposition of relation into Higher Normal form —

Relation decompose into 2NF, 3NF, BCNF with lossless Join & DP decomposition.

2NF decomposition :-

R(o o o o o — XY) ⇒ { o o o o — — (X → Y) }

Not Partial dependency ✓

Suppose this is Partial dependency.

$X^+ = \{XY \ldots\}$

R1(o o o o o X)

R2(XY)
{X → Y}
X: Cand Key

Now this FD's is Not Partial dependency ✓ Bcoz it's Cd's keys ✓

Lossless Join decomposition
&
Dependency preserving
&
No Partial dependency.

Que Relation

R (sid sname DOB cid) {sid → sname DOB} ✓ ← FD's ✓

Cand. Key : sid cid.

Sol^n $\boxed{\text{sid → sname DOB}}$ → is Partial dependency.

So, Rel^n is Not in 2NF ✓

S. go for 2NF decomposition ✓

> Solution is Jiske wajah se problem ho rhi hai ~~uske~~ uska Alag se Table Bna do

R1 (sid cid)

Cd Keys sid cid

[No Non trivial F^n dependency]

So it is in BCNF ✓

(sid)$^+$ = (sid sname DOB)

So R2 (sid sname DOB)

sid is cd keys.

& $\boxed{\text{sid → sname DOB}}$

S.Key

So It is also in BCNF form.

Que R(ABCDEF)

{AB→C, C→D, B→E, E→F}

P. dependency. → Indirect Partial dependency Iska dhayaan Rakho

Sol^n Cand. Key = (AB)

$B^+ = \{BEF\}$

R2 {BEF}

due to Partial dependency Relation is in 1NF But Not in 2NF

decompose into 2NF :-

$B^+ = \{B,E,F\} \xrightarrow{F^n \text{ dependency}} \left\{ \begin{array}{l} B \to E \\ E \to F \end{array} \right\}$

P. Dependency Closure

Referencing key

R1 (AB C D)

{AB → C
C → D}

Referenced Key.

P.K/AK

R2 (BEF)

{B→E
E→F}

No partial dependency Losless Join & dependency preserving in [2NF]

→ In 2NF decomposition 2 min Table & 1 foreign key required

for 3NF decomposition :- Jo 3NF ko follow Nhi kr Rhe hain wo relation Bahar Nikalo aur unka Alag Tables Bnao.

↳ Bcz $\boxed{C\to D \text{ \& } B\to F}$

R1 (ABC)	R3 (C D)	R2 (BE)	R4 (E F)
AB→C	C→D	B→E	E→F

→ 3NF
Also in BCNF

4 Table & 3 foreign key

Que R(ABCDE)

{AB→C, BC→A, AC→B}.

Solⁿ {ABDE, BCDE, ACDE} ← CDKeys.

Heigest Normal form → <u>3NF</u> ✓

(ABDE) R(ABC)

↓ ↓

No Nontrivial FD's. {AB→C, BC→A, CA→B}

Cdkeys→(ABDE) ↓

BCNf ✓ {AB, BC, AC}

↑

CD Keys.

BCNf ✓

{2 Table & 1 Foreign key} ✓

No-Two Partial dependency put into one Table.
Put into different-2 Tables when decomposition done.

Foreign key can be set of attribute

Que R(ABC) {AB→C, C→A}

Solⁿ Candidate keys → {AB, BC}.

<u>Decomposition.</u>

FK

R_1{CB} R_2{CA}

No Nontrivial FD's C→A

BCNF ↑

cd keys.

↑

BCNF

LLJ ✓
BCNF ✓
DP ✗

for Dependency preserving (ABC) must be in one Table
But when (ABC) is in one Table then BCNf is Not Possible ✓

This relation can Not ~~Deform~~ Decompose ~~de~~ with the dependency preserving with ~~deco~~ BCNf. ✓

Scanned with CamScanner

So, ★★★★★

dependency preserving with BCNF is NOT possible for all type of relation. → Kyun ki kuch relations me without ~~dep of~~ dependency preserving ke BCNF mil Jyata hai.

★★★ means $\boxed{BCNF \rightleftarrows DP}$ → may or may not Both sides

Que R(ABCD)

{ABC → D, D → B}. → BCNF failed B'coz of this

Cdkey {ABC, ACD}

(ACD) R2(DB)

D → B.

↑ Nontrivial FD's

↑ CD keys.

- LL Join ✓
- BCNF ✓
- Not Dependency preservation ✓

BCNF ✓

F^n dependencies X

V. Imp

~~DataBase design goal Based on Normalization~~.

DB design goal based on Normalization	1NF	2NF	3NF (Most accurate Normal Form)	BCNF	4NF
★★★ ① Lossless Join decomposition	✓	✓	✓	✓	may not
★★★ ② dependency preserving	✓	✓	✓	may not	may not
★★ ③ 0% Redundancy	X 0% Redundant Not in 1NF	X "	X "	{Yes, over FD's} {No, over MVD}	✓ {Yes over FD's & MVD's}

Priority Increasing order ↑

Note :-

① Most accurate Normal form among all Normal form is 3NF Bcoz Lossless Join & dependency preserving is guranteed.

② Lossless Join & dependency preserving possible for Every relation into 2NF & 3NF. But Not Every Relation can Decompose into BCNF with dependency preservation.

Que R(ABCDEFG)

{AB → C, C → D, A → E, B → F, F → G}

↳ Here AB ⇒ CD Keys

Problems Arise due to → {A → E, B → F} $(A)^+ = \{AE\}$ $(B)^+ = \{BFG\}$

Partial dependencies

So, Put them into another Table

R1 (ABCD)	R2 (AE)	R3 (BFG)
AB → C C → D	A → E	B → F F → G

} → 2NF

But Not in 3NF due to (C → D) & (F → G) Problem arise

R1 (ABC)	R2 (AE)	R3 (BF)	R4 (CD)	R5 (FG)
AB → C	A → E	B → F	C → D	F → G

3NF

4 Foreign key
& 5 Tables required