Check! →

LHS is proper subset of C.K. →
RHS is non-prime att.

↓

for partial dependency i.e,
if true then not in 2nd NF.

	AB → C	C → DE	E → F	F → A.
BCNF	✓	✗	✗	✗
3NF	✓	✗	✓	✓
2NF	✓	✗ (bcz both condⁿ true). ↳ not in 2NF.	✓	✓
1st NF	✓	✓	✓	✓

↳ [Ans is 1st NF.] Ans.

bcz,

→ for 1st NF → that we don't want any multivalued attribute in the table. All att. must be single (atomic).

& in

Table → R (ABCDEF)

all are general att.,

we can't tell them as atomic or multivalued by seeing them.

↳

Table already 1st NF में ही मिलती है।
↳ (assume already).

↳

[1st NF.] Ans.

proper Subset is always less than a set.

31. Find Out Normal Form of a Relⁿ :→
(from 1NF - BCNF).

Q:- R(ABCDEF),
FD's : {AB → C, C → D, C → E, E → F, F → A}

C.K. = {AB, FB, EB, CB}.

P.A. = {A, B, C, E, F}
NPA = {D}

Solⁿ:- Now, let we assume that R(ABCDEF) is already in 1st NF.

→ Check ∀ 2nd NF :→ (partial depen) P.D.

Condⁿ ⇒ { LHS must be proper Subset of any C.K. [and] RHS must be a Non-P.A }

F.D's {AB → C, C → D, C → E, E → F, F → A}
F & F, T & T, T & F, T & F, T & F
FD, PD, FD, FD, FD.

(Full dependency).

∴ not in 2nd NF.

⇒ Make it in 2nd NF :→
We do it by decompose the Table.
(Divide into 2 parts).

Condi^n :→

Common attribute must be C.K.

1.) Lossless Decomposi^n

2.) Dependency should be preserved.

Now, जो problem दे रहा, $C \to D$ उसको Alag कर दो, Alag Table बना दो।

Now,

$$R(ABCDEF)$$

R_1 $(ABCEF)$ $\qquad$ R_2 (CD)

Now, we have to make a common att. b/w these 2 table :-

Criteria of common :- can be C.K. of any R_1 & R_2.

So, in R_2 (CD)

$C \to D$ $\qquad$ $C^+ = CD$ (all 2).

$\Rightarrow$ $\boxed{C \text{ is C.K.}}$

So,

Make 'C' common in both R_1 & R_2.

Now, again

$$R(ABCDEF).$$

R_1 $(ABCEF)$ $\qquad$ R_2 (CD)

R_1 $\{AB \to C, C \to E, E \to F, F \to A\}$

2nd NF.

R_2 $\{C \to D\}$

& F.D. $\qquad$ T | C.F. $\not\subset$

& C itself is not its proper subset

2nd NF.

Now :

Check ∀ 3NF : →

{ LHS must be C.K. [or] . RHS [redacted] be P.A. } → 3NF.

So,

R_1 (ABCEF)

{AB→C, C→E, E→F, F→A}

C.K. = {AB, FB, EB, CB}

PA = {A, B, C, E, F}.

→ It is 3rd NF.

R_2 (CD)

{C→D}

C.K. = {C}.

PA = {C}.

It is also 3rd NF.

Now :

Check ∀ BCNF : →

[Condi^n → L.H.S. must be a C.K.]

R_1 {AB→C, C→E, E→F, F→A}

not in BCNF form.

→ There is Redundancy.

R_2 {C→D}

It is BCNF form.

Redundancy — 0%.

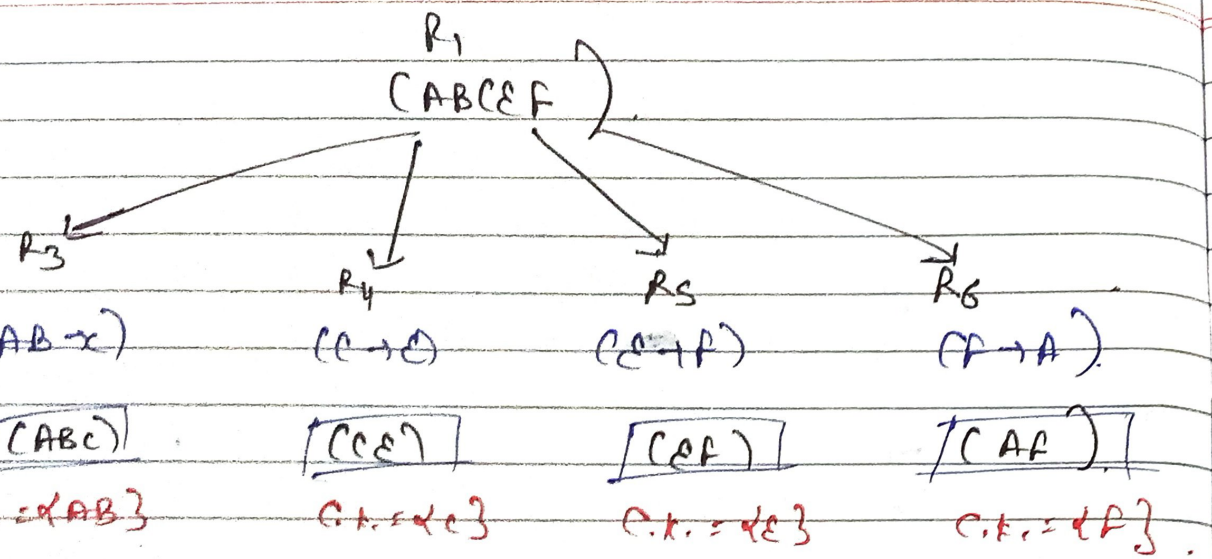
Now,

Again Decompose R_1 (ABCEF).

(जो problem करे रहे, उनको Alag करदो।)

Normalisation aim → To make redundancy 0%.

4 file decomposes, normalise



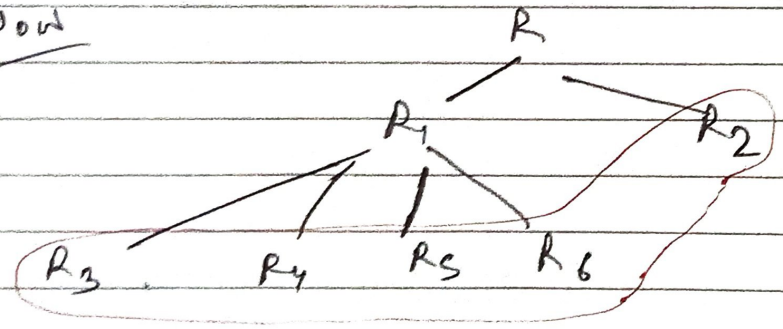
Now, these all 4 tables have C.K.

So,

These all 4 are in BCNF now.

← (कोई - भी - कोई common भी नहीं है इस में)

Now

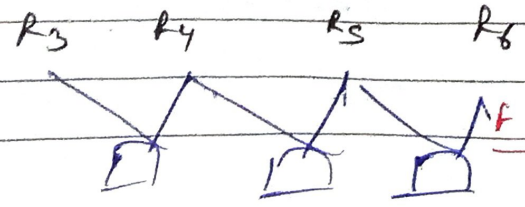


Now, In all these 5 tables,

Redundancy is 0%.

But, there are now multiple tables & if we join them again, it is complex.

Let, have to Join R3 & R4 (both have common (C))



→ But, Join R_3 & R_5.
(ABC) (EF).

We can't join them directly. (bcz nothing is common) So, take help from R_4.

→ So, Combine R_3 & R_4, let, R_3R_4
(ABCE)

Now,
We can combine it with R_5 bcz both have now 'E' as common att.

$R_3R_4R_5$ (ABCEF).
↳ E is C.K. now

---

(32) Normalisation Questions :→

→ A relⁿ is in BCNF, then it is in 2NF also

Q: A Relⁿ R has 8 att. (ABCDEFGH)

F = { CH → G, A → BC, B → CFH, E → A, F → EG }

how many Candidate keys in R.

a) 3 b) 4 c) 5 d) 6.

Solⁿ: First, check on RHS, which att is absent.
D → So, now it comes with every.

$D^+ = D$

$AD^+ = ADBCFHGE$ (all 8).

Now, check A on RHS,
so,
ED^+ also a C.K.
Now, check E on RHS.
so,
FD^+ also a C.K.
Now, check F on RHS.
so,
BD^+ also a C.K.
Now, check B on RHS.
Now, completed.
so,
$\boxed{AD, ED, FD, BD}$ 4 C.K.

---

33. Ques^n Explained on Normalisation :→

Schema is a structure of a Table.

→ so, $\boxed{\text{Scheme (or) Table}}$ → same.

→ Non trivial F.D. means in which.

$\boxed{LHS \cap RHS = \phi}$ → we have to check that it is valid or not.

→ Trivial F.D. are always valid.

Schema 1 : Registration (Roll. No, Courses)

Non trivial F.D of roll no → Courses } :

Ans → We have to check that this table is in which form.

→ So, as always, start from higher form.

→ Check for BCNF :-

condi^n : LHS of every FD must be C.K. or S.K. or P.K.

and,

Roll. no is already given as a C.K.

So,

LHS is a C.K. of F.D.

So,

→ Table is in BCNF form.

i.e, also in 1st NF, 2nd NF & 3NF.

Schema 2 : Registra^n (roll no, Course id, email)

Non trivial FD of roll no, Course id → email
email → roll no. }

Ans → C.K. = { roll no Course id }.
PA = { roll no, Co. Id }.
NPA = { email }.

→ Check BCNF : → 1st F.D. LHS is C.K. but not in 2nd F.D.

So, not in BCNF form.

→ Check for 3NF:

Condⁿ: LHS must be a CK or SK or PK
OR
RHS must be a P.A.

Now, 1st FD is already valid,
&
2nd FD: email → roll no.
F P.A.
T.

∴ (T)

It is in 3NF.

Scheme 3: Registⁿ (roll no, Co-Id, marks, grade).

Non trivial FD: { roll no, Co-id → marks, grade
marks → grade }

Ans: C.K. = { roll no C-id }.
PA = { roll no, C-id }
NPA = { marks, grade }.

→ check if BCNF:

1st FD → valid.
2nd FD → not valid.
∴ not in BCNF.

→ check if 3rd NF:
1st FD → already valid
2nd FD → marks → grade.
F F ∴ not in 3rd NF.

→ check ∀ 2nd NF :→

condⁿ :-

LHS must be a proper subset of CK.
AND
RHS must NPA.

↳ for P.D. → i.e, not in 2nd NF.
if both condⁿ are True.

→ 1st FD → already valid.
2nd FD → marks → grade
f T
(→ f)

↳ not in partial Dependency
∴ It is in 2nd NF. Ans :-

Schema 4 :→ Registⁿ (roll no, C-Id, Credit).

Non trivial FD { roll no, C-Id → Credit
C-Id → Credit }.

Ans :→ C.K. = { roll no C-Id }
PA = { roll no, C-Id }.
NPA = { credit }.

→ check ∀ BCNF :→
1st FD → ✓
2nd FD → ✗ } ∴ not in BCNF.

→ Check ∀ 3NF :-
1st FD → ✓
2nd FD → C-Id → Credit
f f → (f)

So, not in 3rd NF.

→ Check ∀ 2NF ! →

1st FD → ✓

2nd FD → C-Id → Credit
T T
(+ T)

→ It is Partial Dependency (P.D.)

So,
It is not in 2nd NF.

∴
→ Table is in 1st NF.
(we already assume it).

---

34. Cover & Equivalence of F.D. : →

If X covers Y $Y \subseteq X$
& If Y covers X $X \subseteq Y$
& if both are True.
then, $\boxed{X \equiv Y}$ → Equivalent.

Q:- $X = \{A \to B, B \to C\}$ | $Y = \{A \to B, B \to C, A \to C\}$.

i) X covers Y : → $(Y \subseteq X)$
(इसमें Y की FD. में LHS का closure लेंगे लेकिन X वाले में से।)

So,

$A^+ = ABC$ $\qquad$ $B^+ = BC$.

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

$\Rightarrow$ all 3 covers in these closure forms.

$\Rightarrow$ X covers Y. ✓

ii) Y covers X :→

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

(X का closure Y से लेंगे।)

$A^+ = ABC$

$B^+ = BC$

$\Rightarrow$ Y also covers X. ✓

$\Rightarrow$ both symbols cancels each other & become Equivalent.

$X \not\subseteq Y$ $\qquad$ $Y \not\subseteq X$

$\boxed{X \equiv Y}$ Ans. ✓

Q:- $X = \{AB \to CD,\ B \to C,\ C \to D\}$.

$Y = \{AB \to C,\ AB \to D,\ C \to D\}$.

→ i) X Covers Y :→

$Y = \{AB \to C,\ AB \to D,\ C \to D\}$.

$AB^+ = ABCD$, $\qquad$ $C^+ = CD$ $\qquad$ → True

ii) Y covers X? →

$X = \{AB \to CD,\ B \to C,\ C \to D\}$.

$AB^+ = ABCD$, $B^+ = B$, $C^+ = CD$ (from Y).

∴ Y not covers X.
It becomes false.

∴ $\boxed{X \not\equiv Y}$ ok.

& $\boxed{X \supseteq Y}$ ✓, True & $\boxed{Y \supseteq X}$ ✗, false ✓.

---

35. Dependency Preserving Decomposi^n :→

Def^n :→ Let, any table $\boxed{R(ABCD)}$
& also some F.D., $\boxed{FD\{A \to B,\ B \to C\}}$

Now,
we find closure & then find c.k., & then we also find hidden dependencies :→
like.

$FD^+ \{A \to C\}$ by Transitive prop.

and,

→ In normalisa^n, we do decompos^n, then, we divide

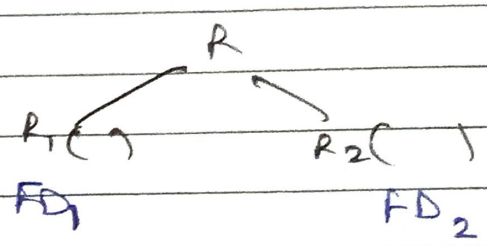
R

$R_1(\ )$ $R_2(\ )$

→ We distribute attr. of R in R_1 & R_2

& union of att. of R_1 & R_2, must be equal to the att. of R.

Now, F.D. also divides, like



Now,

Check! - Dependency should be preserved, (कहीं कोई F.D. lost ना हो यदि हो, सबको preserve करके रखना है।)

ie, $\boxed{FD_1 \cup FD_2 = FD^+}$ (original F.D. के Equal)

Q: Let R(ABCD) with F.D.

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

R is decomposed into R_1(AB), R_2(BC), R_3(BD)

R_1(AB)	R_2(BC)	R_3(BD)
~~A → A~~	B → C ✓	B → D ✓
~~B → B~~	C → B ✓	D → B ✓
A → B ✓		
B → A. ✗		

B^+ = BCD
(don't have A).

union of attr R_1, R_2 & R_3 are Equal to R. So, True. (Right decompos'n)

Now

→ Check Dependency preservation :→

- R_1 (AB), So, It has only A, B attr.

So, whatever Dependency we can made from these 2, make them & write in table.

We don't want Trivial F.D. :→

→ Trivial F.D.: in which intersec^n (∩) is not null.

i.e.,

Ex-1 $A \to A$ has $\cap \neq \phi$ (common is A).

Ex-2 $AB \to A$ has $\cap \neq \phi$ (common is A).

We don't want these bcz, trivial are by default, true. तो होती ही हैं like

$\boxed{A \to A}$ is definitely true.

So,

→ Only take non trivial F.D.: →

$(\cap = \phi)$

→ Now:

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

$A^+ = ABCD$
$B^+ = BCD$
$C^+ = CDB$
$D^+ = DB$

So, acc. to these. select from the non-trivial F.D. (which we finally select). from table. =

Now
F.D. which we get from Table :-

$A \to B$
$B \to C$
$C \to B$
$B \to D$
$D \to B$.

Now, check whether ~~original~~ F.D. are possible through these or not.

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

Now,
$C \to D$ is not direct.
So, now take 'C' closure.

$C^+ = CBD$ ∴ $C \to D$ also preserved.

All F.D. are preserved. True, Ans.

---

36. Dependency Preserving Decompos^n Example

Q.2: Let R(ABCD) with F.D.
{ $AB \to CD$, $D \to A$ }.

Decompose $R_1(AD)$ & $R_2(BCD)$.

$R_1(AD)$	$R_2(BCD)$
$A \to D$ ✗	$B \to CD$ ✗
$D \to A$ ✓	$C \to BD$ ✗
	$D \to CB$ ✗
	$BC \to D$ ✗
	$BD \to C$ ✓
	$CD \to B$ ✗

attributes of $(R_1 \cup R_2)$ make R ✓

Now,

FD: $\{AB \to CD,\ D \to A\}$.

$A^+ = A$, $C^+ = C$

$B^+ = B$, $D^+ = DA$

$BC^+ = BC$,

$BD^+ = BDAC$

$CD^+ = CDA$

So,

F.D. we get from table:→

$\{D \to A,\ BD \to C\}$

Now, check ∀ original F.D.:→

$\{AB \to CD,\ D \to A\}$.

from table

$AB^+ = AB$

$D^+ = DA$ ✓

We can't get

$AB \to CD$, from our decomposed F.D.

Hence,

$\boxed{\text{Dependency preserved not}}$.

37 Joins & Its Types :→

1. **Join :-** When we have to join 2 or more table, to get result.

E_No	Ename	Add.
1	Ram	Delhi
2	Varun	Chd.
3	Ravi	Chd.
4	Amit	Delhi
5	Nitin	Noida

'Employee'

Dep. No	Name	E.no
D_1	HR	1
D_2	IT	2
D_3	MRKT	4
D_4	Finance	5

'Department'.

Select Address from Employee where Ename = 'Varun'
Output → Chd.

↳ In these basic ques^n, we don't need Join, we easily done those by their separate tables.

Now :-

Q :- find Ename of Emp whose working in HR.
Here, HR is in department Table
& Ename is in Employee Table
So,
Here we need Joins.

Note :- There must be some common att. in Tables to use Join. Here, E_no is common.

→ Join is Cross product
+
Select Statement (condition).

Its Types :→

- Cross Join (or) Cross product.
- Natural Join.
- Conditional Join
  - Equi "
  - Self "
  - Outer "
    - Left Outer.
    - Right "
    - full "

---

(38) "Natural Join" :→

→ Join = Cross product + Condi^n.

(m × n)

m

E.No	E name	Add.
1	Ram	Delhi
2	Varun	chd.
3	Ravi	Chd.
4	Amrit	Delhi

Employee

n

Dep No.	Name	Eno.
D_1	HR	1
D_2	IT	2
D_3	MRKT	4

Department.

→ first find :- Table name

Q :- "find the Emp. Names who is working in a department".

(Emp, Dept Means Cross product)

→ Here, we need both, - Table Employee name
- Department Table.

We do it by "Natural Join".

→ Common Att - 'E_No.'

So,

Whenever we have to equalise (बराबर) the values of the common Att, then, we always use NATURAL JOIN.

→ Now, Write Query :→

Select Ename from Emp, Dept where Emp. Eno = Dept. Eno ;

→ Output :→ Ram, Varun, Amrit

'E_No' - must be same in both the Tables.
i.e. (E_No, Emp_No.)

Emp		Dept	
E_No	E name	Dep No.	E No.
1	Ram	D_1	1
1	"	D_2	2
1	"	D_3	4
2	Varun	D_1	1
2	"	D_2	2
2	"	D_3	4
3	Ram	D_1	1
3	"	D_2	2
3	"	D_3	4
4	Amrit	D_1	1
4	"	D_2	2
4	"	D_3	4

So, finally we get 3 rows.

Eno	Ename	DepNo	Eno
1	Ram	D_1	1
2	Varun	D_2	2
4	Amit	D_3	4

Ram
Varun
Amit → Ans.

Now, how to write Actual Query :→

→ Select E_name from Emp Natural Join Dept;

---

39. Self Join :→

→ in which the Table is joined to itself.

S-id

Referencing Cid Table.

S-id	C-id	Since.
S_1	C_1	2016
S_2	C_2	2017
S_1	C_2	2017

student course

Study

Q: find student id who is enrolled in not least 2 co

→ Now, we do self join.

- SQL query always start from - 'from' (sequal)

Query :→

Select ____ (from study as T_1, study as T_2;)

';' → means Cross product.

- So, here we make same Table 2 times & name as T_1 & T_2.

S_1	C_1	2016
S_2	C_2	2017
S_1	C_2	2017

T_2 (m)

S-id	C-id	Since
S_1	C_1	2016
S_2	C_2	2017
S_1	C_2	2017

(n) T_1

m×n

T_1		T_2	
S_1	C_1	S_1	C_1
S_1	C_1	S_2	C_2
S_1	C_1	S_1	C_2 ✓
S_2	C_2	S_1	C_1
S_2	C_2	S_2	C_2
S_2	C_2	S_1	C_2
S_1	C_2	S_1	C_1 ✓
S_1	C_2	S_2	C_2
S_1	C_2	S_1	C_2

Condi^n :→

1 student (S-id) need 2 course (C-id)