Normalization-

→ To remove unnecessary redundancy.

See ease of access
vs
redundancy of data
Inconsistency of data
comes because of a constraint.
decompose the table.

any generic constraint can be defined using trigger.
using roll back.

Lossy decomposition & lossless decomp
do interms of dependency.

Atomic domain of an attribute.

1st Normal Form (1NF) → table → if all domains of attributes are atomic.
→ database → if all tables are in 1NF.

Draw instances of tables

Page No.
Date

Name depends on ID because ek ID ka ek hi name hoga.

∴ Name(ID) → Name → func of ID
Street(ID)
Sal (ID)

but Sal (Street) ✗

ID → Name

ID determines name or name is funtionally dependent on ID

Function dependency → $\boxed{f : \alpha \rightarrow \beta}$

legal instance → instance following all contraints.

functional dependency where α determines β and α & β are subsets of R.

always defined on single table.

Formal def^n of functional Dependency →

legal instance $r(R)$ satisfies $f : \alpha \rightarrow \beta$ iff $(t_1[\alpha] = t_2[\alpha])$
$\Rightarrow (t_1[\beta] = t_2[\beta])$

how database actually ensures constraints

$\forall\ t_1, t_2 \in r$

tuples   instance of R

way how user specifies constraints.

f Holds on R when every legal instance $r(R)$ satisfies 'f'.

Note → If K is a superkey of R then $K \to R$.

Trivial functional dependency → $f: \alpha \to \beta$ where $\beta \subseteq \alpha$

F : {set of functional dependencies}

F — used by database engine checks minimized set of constraints

F^+ — closure — Maximized form — used by user to check if relation is in a specific NF or not with respect to FD.

F_c — canonical cover — Minimized form — no extra functional dependency and no extra attribute in any FD. such that $F^+ = F_c^+$ — extraneous attribute

$F^+ = F \cup$ {set of all functional dependencies logically inferred from F}

DA → DB → D union B

DA means D union A

DBMS 2nd Lecture 10th March →

Calculation of F^+ → invented by Armstrong (moon wala).

- R – Reflexivity — triviality — $f: \alpha \to \beta$ when $\beta \subseteq \alpha$
- A – Augmentation — $\alpha \to \beta \Rightarrow \gamma\alpha \to \gamma\beta$
- T – Transitivity — $\alpha \to \beta, \beta \to \gamma$ then $\alpha \to \gamma$

Rules

→ repeatatively apply these 3 on F to get F^+

Armstrong's Axioms

Also proved that these 3 rules are sound and complete.

sound — galat cheez ko bhi sahi bolna

Date

Write a program to find F^+ using above 3 rules.

input is F

U - Union $\Rightarrow \alpha \to \beta, \alpha \to \gamma \Rightarrow \alpha \to \beta\gamma$

D - Decomposition $\Rightarrow (\alpha \to \beta\gamma) \Rightarrow \alpha \to \beta$ and $\alpha \to \gamma$

P - Pseudotransitivity $\Rightarrow \alpha \to \beta, \gamma\beta \to \delta \Rightarrow \gamma\alpha \to \delta$

rules

~~prove using armstrong's axioms~~

$\alpha \to \beta$

$\alpha \to \gamma$

$\beta\alpha \to \beta\gamma$

$\beta\alpha \to \beta$

$\beta\alpha\gamma \to \beta\gamma$

$\alpha \to \beta$ then $\alpha\gamma \to \beta\gamma$

$\alpha \to \beta, \beta \to \gamma$ then $\alpha \to \gamma$

$\alpha \to \gamma$

$\alpha \to \beta$

proof $\to \alpha \to \beta$ using augmentation $\to$

$\alpha \to \alpha\beta$ (as $\alpha \cup \alpha \to \alpha$)

and $\alpha \to \gamma$

$\Rightarrow \alpha\beta \to \beta\gamma$

$\therefore$ Transitivity $\to \alpha \to \beta\gamma$

$\alpha \to \beta\gamma$

using augmentation $\to$

$\alpha \to \alpha\beta\gamma$

reflexivity $\to$

$\alpha\beta\gamma \to \beta$

transitivity $\to$

$\alpha \to \beta$

$\alpha \to \beta\gamma$

for α

$\to \alpha\beta \to \beta\gamma$

$\to \alpha\gamma \to \beta\gamma$

$\alpha \to \alpha\beta\gamma$

$\alpha\beta\gamma \to \beta\gamma$

Page No.
Date

$\alpha \to \beta$, $\beta\gamma \to \delta$ then $\alpha \to \delta$

$\beta\gamma \to \beta$ reflexivity

$\beta\gamma \to \gamma$

$\alpha\beta\gamma \to \alpha\delta$ $\qquad\qquad$ $\alpha \to \alpha\beta$

$\alpha\delta \to \delta$

$\therefore \boxed{\alpha\beta\gamma \to \delta}$

$\alpha\beta\gamma \to \beta\gamma$ $\qquad$ $\alpha \to \alpha\beta$

$\alpha\gamma \to \alpha\beta\gamma$

$\alpha\gamma \to \delta$

Some more examples →

F : { A → B, ⓪
A → C, ①
CG → H, ②
CG → I, ③
B → H ④
}

Prove ① A → H (From A → B & B → H)
② CG → HI (② ③ union)
③ AG → I (① pe G ka aug then ① ③ transition)

BCNF - check if R is in BCNF with respect to F -

$\forall f \in F$

① f is trivial $f: \alpha \to \beta$

$\Downarrow$ $\beta \subseteq \alpha$

or

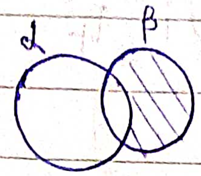
② α is a superkey.

$\Downarrow$

$\alpha \to R$

If not in BCNF then decompose →

R → R_1, R_2



R_1: ~~$(R-\beta)$~~ $(R-(\beta-\alpha))$

R_2: $R_2(\alpha, \beta)$

Here $f: \alpha \to \beta$ is FD which is creating problem.

2 things to check while decomposing -

1. lossless or not ?
2. dependency preserving or not ?

BCNF decomposition is always lossless.

definition of lossless decomposition → $r(R)$ → $r(R_1)$, $r(R_2)$

iff

$$\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) = r$$

→ Condition of lossless decomp → $R_1 \cap R_2 \to R_1$ or $R_1 \cap R_2 \to R_2$

$r(A, B, C, D)$

F: $AB \to CD$ Candidate Key : AB (satisfies 2^{nd} $cond^n$)

$B \to D$ doesnot satisfy either of the 2

Thus, not in BCNF.

∴ Do BCNF decomposition -

$R-(\beta-\alpha) \Rightarrow R-(D-B)$ ∵ B and D are individual attributes

∴ nothing common so $D-B = D$

∴ $D-B = D$

$\Rightarrow R-D = A, B, C$

∴ $R_1 = A, B, C$

$R_2 = B, D$

But here it is not dependency preserving.

Ensure functional dependency using Trigger. But trigger works only on single table.

If we have attributes from different tables then asure Functional dependency using View.

In $AB \rightarrow D$ make a view A, B, D and make A, B as ~~primary~~ Candidate Key (declare unique).

Question → Dependency preserving decomposition nahi hone me problem Kya hai?

To solve the problem 3^{rd} NF was introduced.

When dependency that is creating problem in BCNF, uska right ~~left~~ side jab candidate key Ka part hai to BCNF Ke according decompose Krne se dependency preserving nahi hoga.

Page No.
Date

3NF → $\forall f \in F$

① } → Same as BCNF
②
③ Each individual attribute of RHS is part of any candidate key.
or
every attribute should be prime attribute.

Examples -

prime attribute → attribute which is part of any candidate key.

F: { A → BC,
CD → E,
B → D,
E → A }

$\pi(A, B, C, D, E)$

Calculate A^+, B^+, C^+, D^+, E^+.

$A^+ = \{ABCDE\}$

start scanning in F ki LHS is subset of A^+ or not.
If yes, then add RHS to A^+
do this till no new addition to A^+.

closure of an attribute α means α se apan kya kya determine kr skte hai.

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

$BC^+ = \{BCDEA\}$
$BD^+ = \{BD\}$
$CD^+ = \{CDEAB\}$

CD, BC, A and E are candidate keys because unke closure me poora R aa raha hai

Q→ Given an FD F and another FD f such that $f \notin F$ check if $F \rightarrow f$ or not.

example → ① f: $AB \rightarrow CD$ ✓ as AB^+ me CD hai
② f: $BD \rightarrow AC$ ✗ BD^+ me AC nahi hai.

F is previous wale ka.

To prove any FD is correct → use only using UDPRAT rules.
To check we can use by finding closure.

Q→ R(A, B, C, D, E) is in BCNF or not ?

Ans: not BCNF

$B \rightarrow D$ is problem because B is not superkey.

Q→ What will be decomposition?

$R_1 = \{A, B, C, E\}$ $R_2 = \{B, D\}$

Example → R(A, B, C, D, E, F)

$$F = \{ A \rightarrow BCD \\ BC \rightarrow DE \\ B \rightarrow D \\ D \rightarrow A \}$$

$B^+ = \{BDAC\cancel{D}E\}$

$AF^+ = \{AFBCDE\}$

Q→ Write an SQL Query to check whether $b \to c$ holds on $r(a, b, c)$ or not.

Ans: select count(*) (select R as R_1 join R as R_2 on $R_1.b = R_2.b$)

group by b

next week friday 3 to 5 normalisation quiz

In first week April only lab class no other class for DBMS

this week friday no tut class.

Page No.

Date

Ans: select * from r group by b having count (distinct c > 1)

or

select * from r r1, r r2 where r1.b = r2.b and r1.c != r2.c;

⇓

if b → c is true then ∀b, there will be value for c.
So, this query must be empty for b → c to be true.

minimal form of a set of FD → canonical power

no extra attribute in any $f \in F_c$
no extra f.

Extraneous attribute →

$F' = F - f \cup f'$

$f : \alpha\gamma \to \beta$ here γ check is extraneous or not.
$f' : \alpha \to \beta$

γ is extraneous iff → $F \Rightarrow F'$

if $f : \alpha \to \beta\gamma$
$f' : \alpha \to \beta$
then γ is extraneous iff → $F' \Rightarrow F$

To check practically → see if we can derive $\alpha \to \beta$ from F.

→ check if $\alpha \to \gamma$ is derived from F' or not.

Date

To remove extra Fd → ensure the left hand sides are unique. (if not simply take union)

Q→ Calculate F_c. $F = \{ A \to BC$
$B \to C$
$A \to B$
$AB \to C \}$

$A \to BC$, $A \to B$ → $A \to BC$
$(BC \cup B = BC)$

Check unique LHS

∴ $A \to BC$
$B \to C$
$AB \to C$

extra as $B \to C$ is already there

⇒ $A \to BC$
$B \to C$

now $A \to B$ & $B \to C$
so $A \to C$ also
∴ C is extra

∴ final ans → $\boxed{\begin{array}{l} A \to B \\ B \to C \end{array}}$

Q→ $F = \{ A \to BC$
$B \to AC$
$C \to AB \}$ $F_c = ?$

Ans. if B remove →

$A \to C$ ✓
$B \to AC$
$C \to AB$

if C remove →
$B \to A$
$A \to C$
$C \to AB$

if A remove →

$B \to A$
$A \to C$
$C \to B$ ✓

Date

F_c $\begin{pmatrix} A \to C \\ B \to A \\ C \to B \end{pmatrix}$ $\begin{pmatrix} A \to C \\ B \to C \\ C \to AB \end{pmatrix}$ $\begin{pmatrix} C \to B \\ B \to A \\ C \to A \end{pmatrix}$ all 4 are correct

Q→ Find all prime attributes of $F: \{A \to BCD, BC \to DE, B \to D, D \to A\}$

$R = (A, B, C, D, E, F)$

⇓

Candidate Keys → {AF, BF, DF}

So prime attributes are A, B, D & F.

Q→ $F = \begin{Bmatrix} AB \to C \\ AD \to GH \\ BD \to EF \\ A \to I \end{Bmatrix}$ $R = \{A, B, C, D, E, F, G, H, I\}$

find all prime attributes.

Ans: prime attributes → {A, B, D}

~~$A^+ = \{AI\}$~~

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

~~$AB^+ = \{ABCI\}$~~

~~A~~

right me A, B, aur D ko determine koi kr hi nhi rha hai.

Isliye min ABD to hona hi chahiye candidate key me.

ab ABD^+ dekho poora aa rha hai ki nhi,

since ABD^+ me poora a rha hai

So ABD is only candidate key.

Instructor (id, dept_name, city)

Always do decomposition based on first violating functional dependency.

4NF → multivalued dependency ($\twoheadrightarrow$)

ID $\twoheadrightarrow$ department

then Each ID may correspond to more than one department.

Formal definition - $\alpha \twoheadrightarrow \beta$

iff

$\forall\, t_1, t_2 \in r$ if, $t_1[\alpha] = t_2[\alpha]$ then $\exists\, t_3, t_4 \in r$

such that -

① $t_3[\alpha] = t_4[\alpha] = t_1[\alpha] = t_2[\alpha]$

② $t_3[\beta] = t_1[\beta]$ and $t_4[\beta] = t_2[\beta]$

③ $t_3[R-\beta-\alpha] = t_2[R-\beta-\alpha]$ and $t_4[R-\beta-\alpha] = t_1[R-\beta-\alpha]$

	α	β	$R-\beta-\alpha$
t_1	a	b_1	c_1
t_2	a	b_2	c_2
	⋮	⋮	⋮
t_3	a	b_1	c_2
t_4	a	b_2	c_1

Date

Every FD is also multivalued FD.

multivalued FD is trivial if $\Rightarrow$ $\alpha \twoheadrightarrow \beta$

1. $\alpha \cup \beta = R$

or

2. $\beta \subseteq \alpha$

all FD & multivalued FD's in R

R is in 4NF wrt D iff $\forall$ mvd $\alpha \twoheadrightarrow \beta \in D$ —

1. $\alpha \twoheadrightarrow \beta$ is trivial

or

2. α is superkey.