NORMALIZATION

Informal Design Guidelines for Relational Db

Guideline-1 : Semantics of relation Attr

↳ all the tuples should be related to entitytype / relationship.

eg

entity type

relationship

emp

eno	ename	dno
101	a.	10.

dept

dno	dname
10	
20	

wo {empno, pno}. emp{eno, ...} project{pno,-}

C1 2 col

eno	pno.
101	P01
102	P01
103	P02

T_1

C2 eno ename dno pno 4 col.

T_2

$T_1 < T_2$

Guideline 2 : schema design should not suffer

< insertion, deletion & updation anomaly

eg. Parent

R1

regno	name	age
101	~~x~~ x1	20

R2

regno	name
101	~~x~~ x1
101	~~x~~ x1

col

R3

regno,	name,	marks.
101	~~x~~ x1	
101	~~x~~ x1	

col

Guideline 3 : relation must have few NULL values

eg more null val → form a new table with PK.

emp {eno, mgrssn}.

eno	mgrssn.
101	103
102	103
103	null

cases

↳ invalid / not applicable

↳ value unknown

↳ value not exist

Guideline 4 : lossless join

a) losslessness (X)

b) preserve functional dependency

fd : empno → name

R

R_1 (eno, ename) (fd)

R_2 (eno) (fd ✗)

(a) (X)

$R_1 \cup R_2 = R$. (lossless join)

$R_1 \cup R_2 \neq R$ (lossy join)

Functional Dependency

→ tool to analyze whether the design is good / bad.

→ avoid redundancy of data.

⇓

Normalization

⇓

fd's.

→ performance will be good → no. of scans/search are reduced → response time is good. → cost of query is low.

Formal definition of FD.

emp

R

	eno	ename
t_1	101	x
	102	y
	103	~~z~~
t_2	101	x

$R = \{eno, ename\}$

$X = \{eno\}$

$Y = \{ename\}$

$X, Y \subseteq R$.

t_1, t_2 → two tuples

if $X \to Y$ holds true on R, if $t_1[X] = t_2[X]$ are same then $t_1[Y] = t_2[Y]$

eg

fd: $eno \to ename$, $t_1[eno] = t_2[eno]$ are same then $t_1[ename] = t_2[ename]$

pictorial representation of FD's

emp(eno, ename, desig, sal, pno, pname).

fd1   fd2   fd3

eno → ename , eno → design , eno → pno.

fd1: eno → ename, desig, pno

fd2: pno → pname

fd3: desig → sal.

Armstrong rules.

↳ derive set of FD's from the existing FD's

Rule 1: Reflexivity rule: X → Y if X is super set of Y

eg $\overset{X}{AC}$ → subset of X (A) & (C)

AC → A

AC → C.

Rule 2: Augmentation rule.

If X → Y, then XZ → YZ where Z is an attr added on both sides.

Rule 3 : Transitivity rule

If $X \rightarrow Y$ & $Y \rightarrow Z$ then $X \rightarrow Z$

eg

if eno $\rightarrow$ dno & dno $\rightarrow$ dname

then

eno $\rightarrow$ dname

Rule 4 : Union rule.

If $X \rightarrow Y$ & $X \rightarrow Z$ then $X \rightarrow YZ$.

If L.H.S of both FD's are same, then we combine the R.H.S attributes

Rule 5 : Decomposition rule.

If $X \rightarrow YZ$ then $X \rightarrow Y$ & $X \rightarrow Z$.

Rule 6 : pseudotransitivity rule.

If $X \rightarrow Y$ & $YZ \rightarrow V$ then $XZ \rightarrow V$

Rule 7 : composition rule.

If $X \rightarrow Y$ & $V \rightarrow W$ then $XV \rightarrow YW$.

both L.H.S & R.H.S are combined

Trivial FD (I) $X \rightarrow Y$, X is superset of Y

(II) $X \rightarrow X$

eg eno $\rightarrow$ eno [$X \rightarrow X$]

eno, pno $\rightarrow$ eno [$X \rightarrow Y$] where $Y \subset X$.

Eg to derive FD from given FDs.

R (A, B, C, E, G, H, I }.

$\mathcal{F}$ = {A → B, A → C, CG → H, CG → I, B → H}.

Derived FD's from $\mathcal{F}$

(1) A → A, B → B, C → C → X → X. trivial FD's.

AB → A, AB → B. Reflexivity rule.

AC → A, AC → C. X → Y where Y ⊂ X

Trivial FD's → reflexivity rule/ Identity rule. X → X

X → Y where Y is a subset of X

(2) Augmentation rule

A → C. ⇒ AB → CB.

(3) Transitive rule.

(A → B, B → H ⇒ A → H.

Closure of set of attributes

Assume, schema R

set of FD that holds on R.

eg $\mathcal{F}$: $X \to Y$

$X^+ = \{\ \}$ → maximal super set

Algorithm for X^+

Given: R & $\mathcal{F}$. To find X^+

1) $X^+ = X$

2) Repeat

    for each FD $Y \to Z \in \mathcal{F}$

        if X^+ contains Y then

            $X^+ = X + Z$

    until X^+ does not change.

3) End

fd1 : $A \to B$ fd2 : $A \to C$

LHS   RHS

$CG \to H$ ✗

$CG \to I$ ✗

$B \to H$

Eg: $R\{A, B, C, G, H, I\}$

$\mathcal{F} = \{A \to B,\ A \to C,\ CG \to I,\ B \to H,\ CG \to H\}$

$A^+ = \{A\}$ → based on step ① in Algo.

$A^+ = \{A\ B\ C\ H\}$ → based on step ② in Algo.

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

$CG^+ = \{C\ G\ H\ I\}$

① To find key attribute

$\boxed{\text{if } X \to Y, \text{ if } X^+ = R, \text{ then } X \text{ is a key attribute}}$

From the above eg,

$A^+ \neq R$, $B^+ \neq R$, $CG^+ \neq R$

As the given FD's closure set are not equal to R, we have to derive the new FD's from existing FD. So,

Apply Armstrong rule,

Existing FD's : $A \to \cancel{C}$ $\cancel{C}G \to H$

Apply pseudotransitivity rule,

Derived FD's : $\boxed{AG \to H}$

$AG^+ = \{A\ G\ B\ C\ H\ I\} = R$

Now, AG is the key attribute of R

Extra Points / Inference.

CK = {empno, {empno, pno}, {adharno, pno}}

composite key

PK is minimal set

PK = empno.

PK is minimal storage.

PK = composite key & applicable to all entity

✓ $AC^+ = R$, So AC is a key attr

✓ $BC^+ = R$, BC is prime attr

$A \to B$, augmentation rule

$AC \to BC$.

✓ $AC^+ = R$, AC is also prime attr

CS
CK {comp}
Primary key

CK = {{AC}, {BC}, {AC}, ~~{A}~~}

composite key

candidate set → possible prime attr.

primary/key attr → minimal CK → {A}

less storage → indexing - MM

all composite key → applicable to all entity

CK = {{pno, eno}, {eno, ano}, {eno, dlno}}

② Infer FD from a set of given FD ($\mathcal{F}$)

Given: relation R & FD's $\mathcal{F}$

Assume FD $X \to Y$ does not belong to $\mathcal{F}$

To find:

whether $X \to Y$ holds on R (or)

can we infer $X \to Y$ from $\mathcal{F}$

Soln

1) compute X^+ — $X \to Y$

2) If X^+ contains Y then

$X \to Y$ can be inferred from $\mathcal{F}$

eg

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

To find $A \to CD$ can be inferred from $\mathcal{F}$

Soln 1) $A^+ = \{A\ C\ D\}$

2) R.H.S of $A \to CD$ is present in A^+,

so $A \to CD$ can be inferred from $\mathcal{F}$

$A \to C,\ AC \to D$ [pseudotransitive rule].

$A \to D$.

$A \to C \quad A \to D$ [union rule]

$\therefore A \to CD$ [If this derived FD can be inferred from $\mathcal{F}$, then it is correct]

③ F_1 Covers F_2

Given F_1 & F_2 [FD set]

def 1: F_1 covers F_2.

every FD in F_2 can be inferred from F_1

def 2: Equivalent.

- if F_1 covers F_2 &
  F_2 covers F_1
  then
  F_1 & F_2 are equivalent

eg

$R = \{A, B, C, D\}$ ✓

$F_1 = \{A \to B, B \to C, AC \to D\}$ ✓

$F_2 = \{A \to B, B \to C, A \to D\}$ ✓

1) F_1 covers F_2:

consider each FD in F_2 & check whether it can be inferred from F_1

$A \to B$, $A^+ = \{A\ B\ C\ D\}$ inferred from F_1

$B \to C$, $B^+ = \{B\ C\}$ $\therefore$ inferred from F_1

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

Since all Fd's in F_2 are inferred from F_1, F_1 covers F_2

2) F_2 covers F_1

$A \to B$, $A^+ = \{A\ B\ C\ D\}$ inferred from F_2

$B \to C$, $B^+ = \{B\ C\}$, $B \to C$ is "

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

As all Fd's in F_1 are inferred from F_2,

F_2 covers F_1

As F_1 covers F_2 and

F_2 covers F_1

$\therefore F_1$ & F_2 are equivalent

eg2 $R(A, B, C)$

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

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

1) F_1 covers F_2

$A \to B$, $A^+ = \{A\ B\ C\}$, inferred from F_1

$B \to C$, $B^+ = \{B\}$, not inferred.

2) $\mathcal{F}_2$ covers $\mathcal{F}_1$

$A \to B$, $A^+ = \{A\ B\ C\}$ inferred from $\mathcal{F}_2$

$A \to C$ $A^+ = \{A\ B\ C\}$ "

As $\mathcal{F}_1$ does not cover $\mathcal{F}_2$ $(B \to C)$, we say that $\mathcal{F}_1$ & $\mathcal{F}_2$ are not equivalent.

4) Canonical form of FD

def: R.H.S of FD should contain one attribute

eg R(A B C D E)

$AB \to C$ (it is in canonical form)

$E \to D$ (")

$A \to BC$ (not in canonical form)

Rules Given: R, FD Set $\mathcal{F}$ & G → MINIMAL COVER

G minimal covers $\mathcal{F}$.

1) All FD in G should be in canonical

2) Extraneous attr in G should not be present

3) No FD can be removed from G.

4) G is equivalent to F

remove the redundant FD's.

eg

Given: $R(A\,B\,C)$

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

To find: minimal cover of F (G)

soln

step1: canonical form.

$A \to BC$ is not in canonical form,

we decompose as, $A \to B$ & $A \to C$

$\therefore$ $\cancel{A \to B}$ $A \to C$ $B \to C$ $A \to B$ $AB \to C$

step 2: remove redundant FD's.

$\cancel{A \to C}$ | $B \to C$ $A \to B$ $\cancel{AB \to C}$

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

$B \to C$, $B^+ = \{B\}$, R.H.S of FD is not present

$A \to B$, $A^+ = \{A\}$, "

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

$B \to C$ & $A \to B$.

For each FD

take the closure of L.H.S of the FD

if the R.H.S of the FD is present in the closure

then remove the FD

else

dont remove the FD.

Step 3: L.H.S of the FD's must be a single attr

$\underline{B} \to C$ & $\underline{A} \to B$.

Suppose $AB \to D$

$AB \to A^+ \to$ if B, remove A.

$AB \to B^+ \to$ if A is present remove B.

$B \to D$ & repeat step 2

$G = B \to C$ & $A \to B$.

$\cancel{A \to C}$ $B \to C$ $A \to B$ $AB \to C$ | $A^+ = \{ABC\}$

$B \to C$ $A \to B$ $AB \to C$ | $B^+ = \{B\}$

$B \to C$ $A \to B$ $AB \to C$ | $A^+ = \{A\}$

$B \to C$ $A \to B$ $\cancel{AB \to C}$ | $AB^+ = \{ABC\}$

O/P

$B \to C$ & $A \to B$ & $AB \to D$

Step 3

$AB \to D \Rightarrow B \to D$

$G = \underbrace{B \to C \quad A \to B}$ + $\underbrace{B \to D}$

repeat step 2

No redundant FD are present, then G is the minimal cover of F

Normalization

↳ decompose the relation

↳ minimize redundancy

↳ preserve functional dependency

Types

1NF, 2NF, 3NF & BCNF

higher the NF ⇒ minimize redundancy
⇒ decomposition is high.

---

cost of query → time taken to execute the query → response time → operations involved query

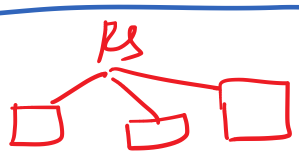
response time = ($t - t_1$)

t : query given to the interpreter

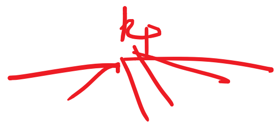
t_1 : first row of result displayed on screen

Join operations → decomposition of table ↑ → higher NF

Rs → 3 relation (3NF)



Rp → 5 relation (6NF)



	redundancy	strength	decomposition	cost
(3NF) Rs	high	weak	low	low
Rp (6NF)	low	strong	high	high

Rs: Rs1 3, Rs2 3, Rp3 2

Rp: Rp1 2, Rp2 2, …, Rp6

8

Q

final inference

balance b/w the NF/decomposition

low NF ⇒ ↑ redundancy ⇒ updation storage.

high NF ⇒ Join is high ⇒ cost ↑

Types of FD

1. Fully FD (FFD)
2. Partial FD (PFD)
3. Transitive FD (TFD)

Invoice (Ino, Lno, qty, Idate)

① **FFD** : non-key attribute depends on full key attribute

(LHS)

eg Ino, Lno → qty } FFD

(full key) (non-key)

Ino → qty } not a FFD

(part of key attr) (non-key)

② **PFD** : non-key attribute depends on part of key attribute

eg Ino → qty } PFD.

(part of key) (non-key)

applicable when u have composite key

eno → ename ⟹ FFD.

③ **TFD** : non-key attr depend on another non-key attr

eg qty → Idate ⟹ TFD.

(non-key) (non-key)

qty → Ino , Ino → Idate

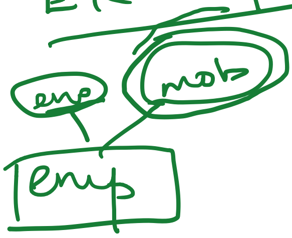
∴ qty → Idate ⟹ TFD.

1 NF

↳ values stored in a relation must be
- * single valued ✓
- * atomic ✓

↳ multivalued attr in db:

ER diagrams



emp(no, name, ...)

⇒ emp_mob(eno, mob)

eg
~~emp_per~~
R

emp no	name	deg
101	x	BA, MA, ...

as 'deg' is a MV attr, it is not in 1NF

R
- R_{11} (R - MV)
- R_{12} (PK(R), MV)

emp_per (R)
- emp1 (eno, name)
- emp2 (eno, deg)

emp_per (eno, name, {{deg}}, {{mobno}})
(R)
MV1, MV2

R
- (R - (MV1, MV2)) R_{11}
- (PK(R), MV1) R_{12}
- (PK(R), MV2) R_{13}

R_{11} (eno, ename)
R_{12} (eno, deg)
R_{13} (eno, mobno)

2NF

A relation is said to be in 2NF
(1) it must be in 1NF
2) should not have <u>PFD</u>

<u>eg</u>. R
Invoice(<u>Ino, Lno</u>, Pno, qty ,Idate).

FD: $\underset{A}{Ino} \longrightarrow \underset{B}{Idate}$ (PFD)

Since PFD holds true on a gn relation, it is not in $\boxed{2NF}$

As it is not in 2NF, decompose the relation
A → B. holds true on R, then decompose R as,

R → R1 (A B), R2 (R − B)

Invoice → Invoice_1 ($\underset{A}{\underline{Ino}}$, $\underset{B}{Idate}$), Invoice_2

Invoice_2 → (<u>Ino, Lno</u>, Pno, qty)

<u>Key attribute</u> must be <u>preserved</u>

Here, full key attr (Ino, Lno) is present in the decomposed relation (Invoice_2) ∴ it is preserved.

3NF

↳ it must be in 2NF

↳ it should not have TFD

eg Invoice(Ino, Lno, Pno, qty, Idate)

FD: $\overset{A}{\text{pno}} \rightarrow \overset{B}{\text{qty}}$ (TFD)

(NK) (NK)

R

AB R-B

Invoice

I_1 I_2

(pno, qty) (Ino, lno, pno, Idate)

1) Eg

$R = \{A, B, C, D, E, F\}$

FD1 : $A \to FC$

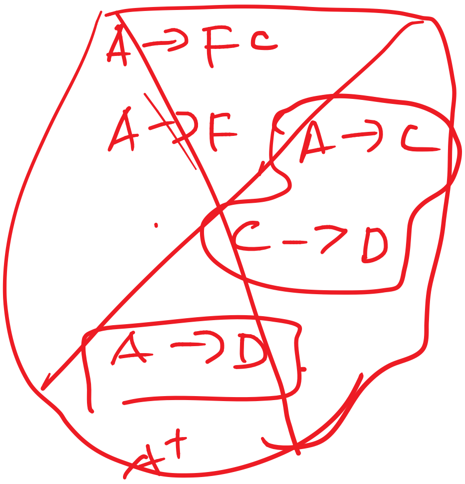
FD2 : $B \to E$

FD3 : $C \to D$.

Find the PK & normalize to 3NF

(i) To find the PK.

Closure Algorithm



$A^+ = \{A\ F\ C\ D\} \neq R$

$B^+ = \{B\ E\} \neq R$.

$C^+ = \{C\ D\} \neq R$.

By augmentation rule, add B on both sides

$AB \to B\ FC$

$AB^+ = \{A\ B\ F\ C\ E\ D\} = R$ ✓

$\therefore$ AB is PK of R.

(ii) 1NF Simple/single & atomic.

All the attr in R is atomic, becoz it is taken care in design phase

All the MV attr are mapped into single valued attr during ER-table conversion

(iii) 2NF : definition (2pts) 1NF no PFD

$A \rightarrow FC$ & $B \rightarrow E$

both the FD's are PFD's.

$R_1(A\,F\,C)$ $\quad$ $R_2(B\,E)$.

$R = \{A\ B\ \cancel{C}\ D\ \cancel{E}\ \cancel{F}\}$. x

$R_3 = \{A\ B\ D\}$.

∴ The normalized relation after 2NF are

$$\boxed{\begin{array}{l} R_1(AFC) \\ R_2(BE) \\ R_3 = (ABD) \end{array}}$$

(iv) 3NF } - defn (2 pts)

FD3: $C \to D$ is a TFD

FD1: $A \to FC$

by decomposition rule, $A \to FC$ can be written as $A \to F$ & $A \to C$.

$A \to C$ & $C \to D$ can be written as

$A \to D$ (transitive rule)

This transitive rule holds true on

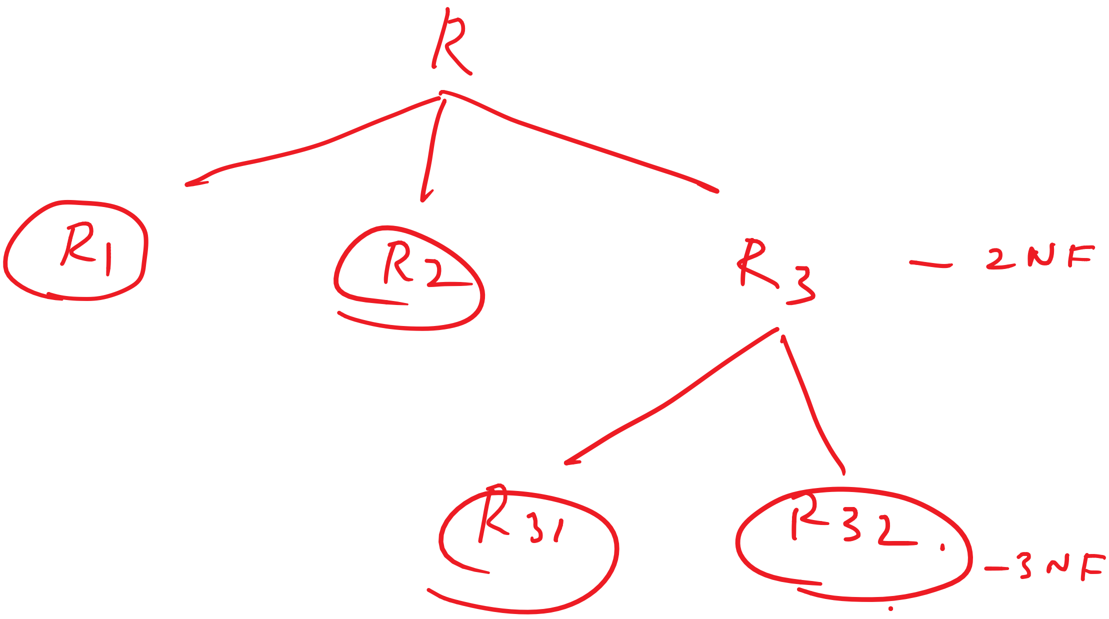
$R_3(A\ B\ D)$

R_{31} (CD)

R_{32} (consider $C \to D$) (AB)

∴ o/p is

$$\boxed{\begin{aligned} R_{31} &= \{CD\} \\ R_{32} &= \{AB\} \end{aligned}}$$



∴ The final relations are

$$\boxed{\begin{array}{l} R_1\ (A\ F\ C) \\ R_2\ (B\ E) \\ R_{31}\ (C\ D) \\ R_{32}\ (A\ B) \end{array}}$$

Practise

(1)

$F = \{A \to B \quad C \to B \quad D \to ABC \quad AC \to D\}$.

1) minimal cover of F - (G)

2) Verify F & G are equivalent FD's.

(2) Car_sale (refer last slide in ppt)