Mod-II

Anamalies in relational database

- Insertion anamaolies
  (Insert some info results in constraints)
- Deletion anamolies
  (we delete some & cause deletion of other undesirable info)
- Updation anamolies
  (we try to update but lead to inconsistency)

Normalisation :-

Process by which that reduce data redundancy from the database and eliminates anomalies from the database

→ Decompose unsatisfactory large relation by breaking into smaller form

→ It is a step by step process & each step is known as NORMAL FORM

→ It is reversible process

1NF, 2NF, 3NF, BCNF } Functional dependency

4NF ⇒ Multivalued dependency

5NF ⇒ Join dependency

⊙ Functional dependency :-

In a relation X & Y are two subset of sets of attributes;

Y is functional dependent on X if a given value of X uniquely determines the value of Y

$$X \rightarrow Y$$

X	Y
a	1
b	22
c	65
a	1

$X \rightarrow Y$

&

if $t_1[x] = t_2[x]$

then $t_1[y] = t_2[y]$

X	Y
a	2
b	2
a	2
b	2
c	2

a → 2
b → 2
c → 2

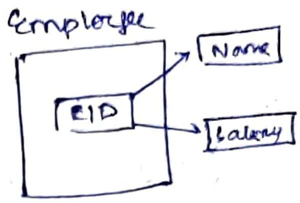
Yes
X → Y

For same value of x, value of y must be same
For diff value of x, value of y may be same

X (EID) → Y (Name, Salary)

means:

x (EID) → y (name)
x (EID) → y (salary)



Classification of FD's :-

i) Trivial

X → Y is trivial if & only if $Y \subseteq X$.

$\overset{x}{AB} \rightarrow \overset{y}{B}$ $B \subseteq AB$

also $x \rightarrow x$
$y \rightarrow y$

ii) Non-trivial

If atleast one attribute in RHS that is not part of LHS

$AB \rightarrow BC$

$A \rightarrow BCD$
$A \rightarrow B$ $A \rightarrow C$ $A \rightarrow D$ $A \rightarrow A$

(iii) Fully functional dependency :-

F.D: $X \rightarrow Y$ y is fully dep. on X if there is no Z where Z is proper subset of X such that
$Z \rightarrow Y$

Ex: F.D $AB \rightarrow C$
$A \rightarrow C$ ✗

iv) Partial Dependency

F.D → F
C.K → R

R(ABCD)
C.K → AB
F.D = {A → C}   Partial

v) Transitive Dependency : Dependency b/w non-prime key attribute

$X \rightarrow Y$, $Y \rightarrow A$ then $X \rightarrow A$

1NF

→ Attribute of table cannot hold multiple values
→ Should have atomic values

Na	Ph
X	66
Y	77, 88

↓ 1NF

X	66
Y	77
Y	98

X	74A, Delhi
Y	64B, Odisha

↓ 1NF

X	74A	Delhi
Y	64B	Odisha

Composite attribute

	Da	WEP
X	21/10	3
Y	10/11	2

↓

X	21/10
Y	10/11

Derived attribute

Prime attribute:

For a given relation $R = \{A_1 \ldots A_n\}$ an attribute is prime attribute if it is part of candidate key.

Non-prime Attribute

An key that is not a part of candidate key

Closure set of FDs:

F^* is set of all FDs that can be determined using given set of FD

Total no. of possible FDs in $R = 2^{2n}$ (n attribute)

ex: $R(A,B)$ FDs

$2^{2\times2} = 2^4 = 16$

Attribute Closure:-

① Finding keys of relation:

If x^+ contain all attributes then X is called superkey of R

$R(ABCDE)$
$\{A \to BC$
$CD \to E$
$B \to D$
$E \to A\}$

$A^+ = \{ABCDE\}$ → C.K
$B^+ = \{BD\}$
$C^+ = \{CEABD\}$
$D^+ = \{DEABC\}$
$E^+ = \{EABCD\}$

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

Additional FD's

R(ABCD)

{ A → BC
B → CD
D → AB }

AD → C is possible or not

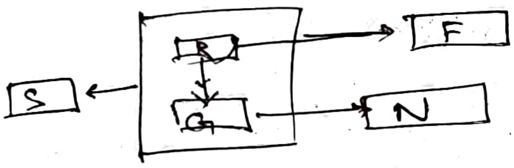
$AD^+ = \{ADBC\}$

AD → C is possible

⊛ Full functional Dependency

Let A be non prime key attribute then A completely dependent upon all prime key attribute

A ⇒ FFD



Prime: R, G
Nonprime: ~~G~~, N, F
G is F.F.D
R, G → S

⊛ Partial Dependency:-

If A non prime key attrible not dependent on all but some primary key attribute.
It is called as partial dependency.

In above ex. F, N are partial de.
R → F   G → N

ABCD → E
& AB → E } then it P.D.

ABCD → F } F.F.D

⊛ "2NF"

A relation is said to be in 2NF if
(i) 1NF
(ii) Non prime key attributes are fully functional dependent upon primary key
(No partial F.D.)
⇓
Non-prime → non prime ✗

$R \rightarrow ABCDE$

F: { $AB \rightarrow C$
$A \rightarrow D$
$B \rightarrow E$ }

$(AB)^+ = \{ABCDE\}$

P.K = {A, B}

N.P.K = {C, D, E}

$AB \rightarrow C$ (FFD) ✓ 2NF

$A \rightarrow D$ (P.D) ×

$B \rightarrow E$ (P.D) ×

$R_1 = (AB \rightarrow C)$

$R_2 = (A \rightarrow D)$

$R_3 = (B \rightarrow E)$

⊛ Armstrong's axioms

1) Reflexivity: If $Y \subseteq X$ then $X \rightarrow Y$ which also implifies $X \rightarrow X$

2) Augmentation:

If $X \rightarrow Y$ then $XZ \rightarrow YZ$ where Z is an set of attribute

3) Transivity:

If $X \rightarrow Y$ and $\underbrace{Y \rightarrow Z}_{\text{Non prime key relationship}}$ then $X \rightarrow Z$ hold

4) Union If $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$ holds

5) Decomposition:

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

6) Pseudotransitivity:

If $X \rightarrow Y$ and $YZ \rightarrow W$ then $XZ \rightarrow W$ holds

1. $R = \{ABCDEFG\}$

FD: $\{A \to B \quad BC \to DE$
$AEF \to G \quad E \to C \quad A \to E\}$

<u>Closure</u>
LHS को निकाल देना

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

$BC^+ = \{BCDE\}$

$AEF^+ = \{AEFGCBD\}$

$E^+ = \{EC\}$

2. Candidate keys :-
It is minimal set of attributes whose attribute closure is set of all attributes called as candidate key.

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

$FD = AB \to C \quad CD \to E \quad DE \to B$

$AD \to C, B, E$

$ADC^+ = \{ADCEB\}$
$ABD^+ = \{ABDCE\}$
$ADE^+ = \{ADEBC\}$

$C.K = \{ADC^+, ABD^+, ADE^+\}$

$R = \{ABCDEF\}$

$AB \to C$
$C \to D$
$D \to BE$
$E \to F$
$F \to A$

$A^+ = \{A\}$
$B^+ = \{B\}$
$C^+ = \{C, D, B, E, F, A\}$ ck
$D^+ = \{DBEFAC\}$ ck
$E^+ = \{EFA\}$
$F^+ = \{FA\}$

$AB^+ = \{ABCDEF\}$ ck
$AE^+ = \{AEF\}$
$AF^+ = \{AF\}$

$AEF^+ = \{AEF\}$ ×

$BE^+ = \{BEFACD\}$ ck
$BF^+ = \{BFACDE\}$ ck
$EF^+ = \{EFA\}$

$CK = \{C, D, AB, BE, BF\}$

Convert relation to 2NF

Q1)

R = {A, B, C, D}

$AB \to D$ {A, B} CK

$B \to C$ {C, D} N.P attr.

$AB \to D$ 2NF ✓ {AB^+} = {A.B.D.C}

$B \to C$ ✗

Decomposition

→ R_1 (AB, D) $AB \to D$

→ R_2 (B C) $B \to C$

Q2)

R = {A, B, C, D, E}

$AB \to C$

$B \to DE$

$A \to D$

AB ⇒ Primary key

CDE ⇒ N.P. attr.

$AB \to C$ ✗ R_1 (A B, C)

$A \to D$ ✗ R_2 (A, D)

$B \to DE$ ✗ R_3 (B, D E)

⊛ 3NF

A relation is said to be 3NF if

(i) Satisfies 2NF

(ii) Non primary key attribute must be non transitive dependence

To convert into 3NF

① General ② (Non prime) wala

Q R {A B C D E}

FD ⇒ $AB \to C$ (AB)$_{CK}$

$B \to D$

$D \to E$

$B \to D$

$D \to E$ $B \to E$

$AB \to C$ 3NF ✓ No transm

$B \to D$ ✗ 2NF P.F.D

$D \to E$ Non P.F.D

convert to 2NF

$R_1(A\ B, C)$ 2NF

R_2 (B, DE } 2NF

convert 3NF

R_1 (AB, C) 3NF

R_2 (B, DE)

R_2 (D, E) → R_3(B, D)] 3NN

2NF 3NF 2NF 3N

Property of Normalisation:-

Two additional prop. also must hold on the decompisition of relation:-

(i) Lossless Join

(ii) Dependency preservation

① Lossless Join:

It ensures that no data will be lost after decomposition that means we will recover the original relation from the decomposition relation

$R \to R_1, R_2, R_3, R_4$ $\quad$ $R_1, R_2, R_3, R_4 \to R$ $\quad$ $R_1 \bowtie R_2 \bowtie \dots R_n = R$

→ It contains equal or less no of tuples without losing data

Lossy Join (Extra tuples)

$R_1 \bowtie R_2 \supset R$ (Natural Join)

Lossless Join is a compulsory in Norm.

② Dependency Preservation Property:

It ensures function dependency keep alive after decomposition also

$R(F) \to R_1(F_1), R_2(F_2) \to (F_1 \cup F_2)^+ = F^+$

Lossless Join Property:

X	Y	Z
A	6	7
B	3	4
C	8	4

$R_1(XYZ)$

X	Z
A	7
B	4
C	4

$R_2(XZ)$

Z	Y
7	6
4	3
4	8

$R_3(ZY)$

X	Y	Z
A	6	7
B	3	4
B	8	4
C	3	4
C	8	4

Lossy Join

X	Y	Z
A	6	7
B	3	4
C	8	4

X	Y
A	6
B	3
C	8

Y	Z
6	7
3	4
8	4

X	Y	Z
A	6	7
B	3	4
C	8	4

$R_1(XYZ) \Rightarrow R_2(XY) \bowtie R_3(YZ) \Rightarrow R(XYZ)$

Lossless

Condition to Check :-

$R \Rightarrow R_1 \ \& \ R_2$

For Lossless Join

(i) Union of R_1 & R_2 must contain all the attribute of R

(ii) Must be common attribute

$R_1 \cap R_2 \neq \phi$

(iii) Intersection of R_1 & R_2 must be superkey of R_1 or R_2 or Both

Q) R(ABCD) $R_1(AB)$ & $R_2(CD)$ is Lossless or lossy?

Ans

$R_1(AB) \cup R_2(CD) = R(ABCD)$ ✓

$R_1 \cap R_2 = \phi$ Fails

Hence it is Lossy Join

Q) R (ABCDEF)

R_1 (ABC) $A \to CD$

R_2 (AFD) $B \to C$ Lossless or Lossy?

R_3 (EF) $F \to DE$

$F \to A$

A.)

R (ABCDEF)

R(ABC) R_2 (AFD) R_3(EF)

R_4 (AEFD)

(i) $R_1(ABC) \cup R_4(AEFD) = R(ABCEFD)$

(ii) $R_1(ABC) \cap R_4(AEFD) = A \neq \phi$

(iii) R_1(ABC) $B \to C$

$AB^+ = \{ABC\}_{ck}$

R_2(AEFD)

$F \to DE$

$F \to A$

$F^+ = \{FDEA\}_{ck}$

AB^+ F^+

A Not super key $\Longrightarrow$ "Either R_1 or R_2 mein sirf 'A' aata ya phir dono mein bhi ok hai"

LOSSY JOIN

Dependency Preservation Property :-

Q) R(ABC) $F = \{A \to B, B \to C, C \to A\}$

R_1 (AB) R_2 (BC) $\{F_1 \cup F_2\}^+ = F$

$A \to B$ $A^+ = \{ABC\}_{ck}$

$B \to C$ $B^+ = \{BCA\}_{ck}$ F^+

$C \to A$ $C^+ = \{ABC\}_{ck}$

$R_1(AB)$ $F_1 \begin{cases} A \to B \\ B \to A \end{cases}$ (transitivity)

$R_2(BC)$ $F_2 \begin{cases} B \to C \\ C \to B \end{cases}$ (transitivity)

$(F_1 \cup F_2)^+ = \{A \to B, B \to A, B \to C, C \to B\}^+ = F^+$

$A \to C$

$A^+ = \{ABC\}$ $C^+ = \{ABC\}$

$B^+ = \{ACB\}$

BCNF (Boyce Codd Normal Form)

→ Strict format of BCNF

→ A relation is said to be in BCNF if

(i) It is in 3NF

(ii) All determinants are candidate keys
(LHS)

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

A & B ⇒ deter. ⇒ Cand.

Every BCNF is also 3NF but vice versa is not true

Ex:

R(ABC)

$A \to B$ —— $B \to B$

$A \to C$

A & B are candidate keys

$A \to BC$

$B \to B$

$A^+ = \{ABC\}_{ck}$

$B^+ = \{BC\}$. ×

Lossless cw

BCNF

$R_1(AB)$, $R_2(BC)$

$R_1 \cup R_2 = R$

$R_1 \cap R_2 = C$

$R_1(AC)$ $R_2(BC)$

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

$A^+ = \{AC\}_{ck}$ $B^+ = \{BC\}_{ck}$

(A) (B)

(C) super key

Lossless Join

BCNF

Multivalued Functional Dependency:

for R(ABC), there is multivalued dependency of B on attribute A if and only if the set of B value associated a given value of A and is independent of the C value.

$A \to\to B$ (A multidetermines B values)

If $A \to\to B$ then $A \to\to C$

$B \to\to C$ ✓

$C \to\to B$ Not possible

If any two tuples t_1 & t_2 has same A value, then there must exist two other tuples t_3 & t_4 obey the rule

(i) t_3 & t_4 has same A values of t_1 & t_2.

(ii) $t_3 = t_1 = B$

(iii) $t_4 = t_2 = B$

(iv) The dependency $A \to\to B$ is called multivalued if $B \subseteq A$ or $A \cup B = R$

4NF:

$A \to t_1 = t_2 = t_3 = t_4 =$ Deepak

$B \to t_1 = t_3 = P_1$ - Redundancy.

$B \to t_2 = t_4 = P_2$

Prog	Proj
Kapil	P_3
Deepak	P_1
Sanjay	P_6
Deepak	P2
Rahul	P1
Sonia	P9

Proj	Mod
P_3	M01
P_1	M31
P_6	M91
P_2	M61
P_1	M91
P_1	M71
P2	M75
P9	M01

$A \to\to B$

(A&B) Superkey

$B \to\to C$

(B&C) superkey

5NF :

- A relation R is in 4NF & it decomposed into $R_1, R_2, \ldots R_N$ then R satisfies the Join dependency

$$*(R_1, R_2, R_3 \ldots R_n) = R$$

If & only If Joining of

R_1 to $R_n = R$

$$R_1 \bowtie R_2 \bowtie R \ldots \bowtie R_n = R$$

X#	Y#	Z#
x1	y1	z2
x1	y2	z1
x2	y1	z1
x1	y1	z1

XY

X#	Y#
x1	y1
x1	y2
x2	y1

YZ

Y#	Z#
y1	z2
y2	z1
y1	z1

ZX

Z	X
z2	x1
z1	x1
z1	x2

X#	Y#	Z#
x1	y1	z2
x1	y1	z1
x1	y2	z1
x2	y1	z2
x2	y1	z1

X#	Y#	Z#
x1	y1	Z2
x1	y1	z1
x1	y2	z1
x2	y1	z1

(w)

Equivalence functional dependency :

F:

$\cancel{A} \to B$

$B \to C$

$C \to A$

G:

$A\cancel{B} \to BC$

$B \to A$

$C \to A$

$A^+ = \{ABC\}$

$B^+ = \{BAC\}$

$C^+ = \{ABC\}$

$F \subseteq G$

$A^+ = \{ABC\}$

$B^+ = \{BCA\}$

$C^+ = \{CAB\}$

$G \subseteq F$

So $F = G$

$AB \to C$

$A \to C$

$B \to C$

$\}$ X (wrong) But

$A \to BC$

$A \to B$

$A \to C$

<u>Canonical form</u> : Redundancy

$R(W, X, YZ)$

$X \to W$

$WZ \to XY$

$Y \to WXZ$

convert into single one/decom.

$X \to W$ ✓

$\cancel{WZ \to X}$

$WZ \to Y$ ✓

$\cancel{Y \to W}$

$Y \to X$ ✓

$Y \to Z$ ✓

$X^+ = \{\cancel{X}W\}$

$X^+ = \{X\}$

$\}$ Dependency as dec.

$WZ^+ = \{WZXY\}$

$WZ^+ = \{XY\cancel{W}Z\}$

$WZ^+ = WZYX$

$WZ^+ = \{WZ\}$

$\}$ imp

$Y^+ = \{WXZ\}$

$Y^+ = \{XZ\cancel{W}\}$

$Y^+ = \{YXZ\}$

$Y^+ = \{YZ\}$

$\}$ less

$Y^+ = \{\cancel{Y}XW\}$ ✓

(without Z)

So,

$X \to W$

$WZ \to Y$

$Y \to X$

$Y \to Z$

$WZ^+ = \{WZY\cancel{X}\}$

$W^+ = \{W\}$

$Z^+ = \{Z\}$

Fm (Ans)

$X \to W$

$XZ \to Y$

$Y \to XZ$

Min

R(ABCD)

A → B
C → B
D → ABC
AC → D

A → B ✓
C → B ✓
D → A ✓
~~D → B~~ ✗
D → C ✓
AC → D ✓

Minimal cover
Set of functional dependency that is reduced version of given set of FD

A → B
$A^+ = \{AB\}$
$A^+ = \{A\}$ dep

C → B
$C^+ = \{CB\}$
$C^+ = \{C\}$

D → A
$D^+ = \{DABC\}$
$D^+ = \{DBC\}$ dep

D → B
$D^+ = \{DABC\}$
$D^+ = \{ACBD\}$ ✓

D → C
$D^+ = \{DACB\}$
$D^+ = \{DAB\}$ ✓

AC → D
$AC^+ = \{ACDB\}$ ✓
$AC^+ = \{ACB\}$

final
A → B
C → B
D → A
D → C
AC → D

AC → D
$AC^+ = \{ACDB\}$ Im
$AC^+ = \{ACB\}$
$A^+ = \{AB\}$
$C^+ = \{CB\}$
Both are req

Q) R(VWXYZ)

V → W
VW → X
Y → VXZ

V → W ✓
VW → X ✓
Y → V ✓
~~Y → X~~
Y → Z ✓

V → W
$V^+ = \{VWX\}$
$V^+ = \{V\}$

VW → X
$VW^+ = \{VWX\}$
$VW^+ = \{VW\}$

Y → V
$Y^+ = \{YVXZW\}$
$Y^+ = \{YXZ\}$ ✓

Y → X
$Y^+ = \{YXZWV\}$
$Y^+ = \{VZWXY\}$

Y → Z
$Y^+ = \{YZVWX\}$
$Y^+ = \{YVWX\}$

$VW \rightarrow X$

$\{VW\}^+ = \{VWX\}$

$V^+ = \{VWX\}$] w को हटाने से फर्क नहीं पड़ता

$W^+ = \{W\}$

$V \rightarrow W$
$VW \rightarrow X$
$Y \rightarrow V$
$Y \rightarrow Z$

$V \rightarrow X$

Final ans:

$V \rightarrow W$
$V \rightarrow X$
$Y \rightarrow V$
$Y \rightarrow Z$

$V \rightarrow WX$ *
$Y \rightarrow VZ$

Keys

$(Key)^+ = R$ (whole relation)

↓

unique value

Set of attributes which can uniquely identify a set of tuples/rows in a table

$R(ABCD)$

$A \rightarrow BC$

$A^+ = \{ABC\}$ ×

A se D नहीं निकाल skta

So A is not a key

$AD^+ \rightarrow \{ADBC\}$

AD is a key

$ABC \rightarrow D$
$AB \rightarrow CD$
$A^+ \rightarrow BCD$

ABC ✓
AB ✓ } all are superkeys
A ✓

A^+ } Candidate key (Efficient/minimal version)

↓

whose proper subset is not a superkey.

Super key
allkeys

$B \rightarrow ACD$
$ACD \rightarrow B$

$B^+ = \{ACD\}$
$ACD^+ = \{ACDB\}$] Both are candidate key

Primary key: Among ckkey one is choosen as Prim. k by data administrator to identify tuples

R(ABCD)

$AB \to C$

$C \to BD$

$D \to A$

$AB^+ = \{ABCD\}$
$C^+ = \{CBDA\}$] CK
$D^+ = \{DA\}$ ×

To find Candidate Key :-

1) R(ABCD)

$A \to B$

$B \to C$

$C \to A$

'D' essential attribute

$AD^+ \to ABCD$
$BD^+ \to BDCA$
$CD^+ \to CDAB$
] ⇒ Minimal C.K

2) $AB \to CD$

$D \to A$

B is essential attribute

$AB^+ = ABCD$

$D = AD$ ×

(AB , BD) ck

$DB^+ = BDAC$

3) R(ABCDEF)

$AB \to C$

$C \to D$

$B \to AE$

B, F are essential attribute

$BF = \{BFAEC\}$ CK

~~ABF~~

No need to check further

4) R(ABCD)

$AB \to CD$

$C \to A$

$D \to B$

$A^+ = A$

$B^+ = B$

$C^+ = CA$

$D^+ = DB$

$AB^+ = \{ABCD\}$ ck

$AC^+ = \{AC\}$

$AD^+ = \{ADBC\}$ ck

$BC^+ = \{BCAD\}$ ck

$BD^+ = \{BD\}$

$CD^+ = \{CDAB\}$ ck

A

{B}

{C}

{D}

#

AC^+, AD^+, CD^+, BC^+

5) R(ABCDE)

AB → CD

D → A

BC → DE

B is essential attribute

$AB^+ = \{ABCDE\}_{ck}$

$BC^+ = \{BCDEA\}_{ck}$

$BD^+ = \{BDACE\}_{ck}$

$BE^+ = \{BE\}$ ×

C.K

⇒ AB, BC, BD

6) R(WXYZ)

Z → W

Y → XZ

XW → Y

$X^+ = \{X\}$

$Y^+ = \{YXZW\}_{ck}$

$Z^+ = \{ZW\}$

$W^+ = \{W\}$

X

Z

W

$XZ^+ = \{XZW.Y\}_{ck}$

$XW^+ = \{XWYZ\}_{ck}$

$ZW^+ = \{ZW\}$ ×

CK = Y, XZ, XW

7) R(ABCDEF)

AB → C

DC → AE

E → F

B & D are essential attribute

$BD^+ = \{BD\}$ ×

$ABD^+ = \{ABDCEF\}_{ck}$

$CBD^+ = \{CBDAE\}_{ck}$

$EBD^+ = \{EBDF\}$ ×

$FBD^+ = \{FBD\}$ ×

$EFBD^+ = \{EFBD\}$ ×

C.K = ABD, CBD

8) R(ABCDEFGHIJ)

AB → C

AD → GH

BD → EF

A → I

H → J

A, B, D are essen

$ABD^+ = \{ABDCGHJIE\}$

ABD is CK

9) R(ABCDE)

BC → ADE

D → B

C is essential

$AC^+ = AC$

$BC^+ = \{BCADE\}_{ck}$

$CD^+ = \{CDBADE\}_{ck}$

$EC^+ = \{EC\}$

$AEC = \{AEC\}$ ×

C.K BC, CD

10) R(ABCDEF)

$AB \to C$
$C \to D$
$D \to BE$
$E \to F$
$F \to A$

$A^+ = \{A\}$
$B^+ = \{B\}$
$C^+ = \{CDBEFA\}_{ck}$
$D^+ = \{DBEFAC\}_{ck}$
$F^+ = \{FA\}$
$E^+ = \{EFA\}$

A
B
F
E

$AB^+ = \{ABCDEF\}_{ck}$
$AF^+ = \{AF\}$
$BF^+ = \{BFACDE\}_{ck}$

$AE^+ = \{AEF\}$
$BE^+ = \{BEFACD\}_{ck}$
$FE^+ = \{EFA\}$ ×
$AEF^+ = \{AEF\}$ ×

CK, {C, D, AB, BF, BE}

11) R(ABCDEFGH) {

$CH \to G$
$A \to BC$
$B \to CFH$
$E \to A$
$F \to EG$

D is essential
$AD = \{ADBCFHEG\}_{ck}$
$BD = \{$

Q) R(ABC)

F: $A \to BC$
$B \to C$
$A \to C$
$AB \to C$

A is essential attribute
$A^+ = \{ABC\}_{ck}$ ✗
So minimal cover is {A}

$A \to B$ ✓
~~$A \to C$~~
$B \to C$ ✓
~~$A \to C$~~
~~$AB \to C$~~

$A \to B$
$A^+ = \{ABC\}$
$A^+ = \{AC\}$ ×

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

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

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

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

$\begin{bmatrix} A \to B \\ B \to C \end{bmatrix}$

~~$A \to BC$~~

2NF:

R(ABCD)

$AB \to D$ $\quad$ $AB^+ = \{ABCD\}$

$B \to C$ $\quad$ $CK = \{AB\}$

- A & B are prime attribute
- C & D are non-prime attribute

$AB \to D \Rightarrow$ Fully F.D.

$B \to D \Rightarrow$ Partial F.D

If B becomes NULL then we can't compute D as a result the prop. of prime attribute is violated

2NF: 1NF & No Partial F.D

R(ABCD) $\quad$ R_1(ABD) $\quad$ R_2(BC)

R(ABC)

$B \to C$

$AB^+ = \{ABC\}$

AB is the candidate key

A, B are Prime Attr.

C is N.P. Attr.

there is partial dependency

R_1(AB) $\quad$ R_2(BC)

Convert to 2NF

(i) R(ABCDE)

$AB \to C$

$D \to E$

$ABD^+ = \{ABDCE\}_{CK}$

A, B, D are prime

C, E are N.P

$AB \to C$ (Partial FD)

$D \to E$ (Partial)

R_1(ABC) $\quad$ R_2(DE)

R_3(ABD)

2) R(ABCDE)

$A \to B$

$B \to E$

$C \to D$

$AC^+ = \{ACBED\}_{CK}$

A, C are prime

B, D, E are N.P.

$A \to B$ (P.D)

$B \to E$ (F.F.D)

$C \to D$ (P.D)

trans ithvith

R_1(ABE) $\quad$ R_2(CD)

R_3(AC)

3) R(ABCDEFGHIJ)

$AB \to C$
$AD \to GH$
$BD \to EF$
$A \to I$
$H \to J$

$ABD^+ = \{ABCDGHEFIJ\}$ ck

A, B, D are prime att.
C, E, F, G, H, I, J are non-prime attr.

$AB \to C$ (P.D)
$AD \to GH$ (P.D)
$AD \to G$ $AD \to H$

$BD \to EF$ (P.D)
$A \to I$ (P.D)

$H \to J$ (F.D)

$R_1(ABC)$
$R_2(ADGHJ)$
$R_3(BDEF)$
$R_4(AI)$
$R_5(ABD)$

3NF :- (i) 2NF
(ii) No transitive dependency

R(ABCD)
$AB \to C$ (F.F.D) $AB^+ = \{ABCD\}$ ck
$C \to D$ (-)   A, B are prime
C, D are N. Prime

The above one is in 2NF

Here there is transitive dependency (N.P → N.P)

$R_1(ABC)$ $R_2(CD)$] → 3NF

AB se 'C' mil जाएगा अगर 'C' null hua to h 'D' nhi nikal paye
and it violates the rule that AB ~~has~~ can key

Q(i) R(ABCDE)

$AB \to C$ (FFD) $AB^+ = \{ABCDE\}$ ck
$B \to D$ (P.D)
$D \to E$ (-)

A, B are Prime attr.
C, D, E are Non Prime

Not in 2NF

- $R_1(ABC)$
- $R_2(BDE)$
  - $R_{11}(BD)$
  - $R_{12}(DE)$
- ~~$R_3(AB)$~~

2) R(ABCDEFGHIJ)

$AB \to C$ (P.F.D)

$A \to DE$ (P.D)

$B \to F$

$F \to GH$

$D \to IJ$

$AB^+ = \{ABCDEFGHIJ\}$

A, B are prime

C, D, E, F, G, H, I, J are N.P

- $R_1(ADEIJ)$
  - $R_{11}(ADE)$
  - $R_{12}(DIJ)$
- $R_2(BFGH)$
  - $R_{21}(BF)$
  - $R_{22}(FGH)$
- $R_3(ABC)$
- ~~$R_4(AB)$~~

3) R(ABCDEFGHIJ)

$AB \to C$

$BD \to EF$ (P.F)

$AD \to GH$ (P.D)

$A \to I$ (P.D)

$H \to J$

~~$AB^+ \neq R$~~

$ABD^+ = \{ABDCEFGHIJ\}$

- $R_1(ABCI)$
  - $R_{11}(ABC)$
  - $R_{12}(AI)$
- $R_2(BDEF)$
- $R_3(ADGHJ)$
  - $R_{31}(ADGH)$
  - $R_{32}(HJ)$
- $R_4(ABD)$

BCNF:-

(i) 3NF
(ii) All determinants must be candidate key

2NF $\alpha \rightarrow \beta$
P ⊗ N.P

3NF $\alpha \rightarrow \beta$
NP ⊗ NP

BCNF
$\alpha \rightarrow \beta$
super key ⊗ P

BCNF $\alpha \longrightarrow \beta$
P/NP. P

R (ABC)

$AB \rightarrow C$ $AB^+ = \{ABC\}$ in 3NF
$C \rightarrow B$ $AC^+ = \{ACB\}$ but not in BCNF

$C.K = \{AB, AC\}$
A, B, C are prime attr.
No transitive dependency

$C \rightarrow B \Rightarrow$ (c is not a superkey)

- R_1 (CB)
- R_2 (AC)

Q) R(ABCDEFGHIJ)

$AB \rightarrow C$ (F.D)
$A \rightarrow DE$ (P.D) $AB^+ = \{ABCDEFIJGH\}$ ck
$B \rightarrow F$
A, B are prime.
$D \rightarrow IJ$
C, D, E, F, G, H, I, J are NP
1st NF ✓ $F \rightarrow GH$

Not in 2NF

- R_1 (ABC) 3NF
- R_2 (ADEIJ)
  - R_{21} (ADE)
  - R_{22} (DIJ)
  - 3NF
- R_3 (BFGH)
  - R_{31} (BF)
  - R_{32} (FGH)
  - 3NF

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

F: { AB → C
B → D (P.D)
A → E }

$AB^+ = \{ABCDE\}$ CK

A, B are P.A
C, D, E are N.P

R_1 (A B C E)

R_{11} (A B C)

R_{12} (A E)

R_2 (B D)

R_1 (A B C)

R_2 (B D)

R_3 (A E)

Q) R (A B C D E)

F: A → BC
CD → E
B → D
E → A

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

Lossless Join proved ~~Dec~~.

$R_1 \cup R_2$ = ABCDE

$R_1 \cap R_2$ = A

R_1 (A B C)
A → BC
$A^+ = \{ABC\}$

$R_2 = \{ADE\}$
E → A
$DE^+ = \{DEA\}$

A^+ DE^+

R_1(ABC)	R_2(ADE)
A → BC	A
	D
	E → A
AB → ABC	AD
	DE
	AE

A super key

Lossless hence the dec. is right

Dependency preser: Not mandatory

R : F

R_1 F_1 | R_2 F_2

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

Q) R(ABC)

A → B

B → C

C → A

R_1(AB) | R_2(BC)

F_1: A → B | F_2: B → C

B → A | C → B

(B→C C→A) | (C → A, A→B)

$C^+ = \{CAB\}$

$F_1 \cup F_2 = F$

(A → B) (B → C) (B → A, C → B)

Q2) R(ABCD)

AB → CD

D → A

BDA

CD

R(AD) | R(BCD)

F_1: ~~(A → D)~~ | BD → C

D → A

$AB^+ = \{AB\}$

D epen ✗

R(PQRST)

F: $PQ \to R$

$S \to T$

$PQS^+ = \{PQRST\}_{CR}$

P, Q, S are pr.

R T are NP

$PQ \to R$ (P.D)

$S \to T$ (P.D.

R_1 (PQR)

R_2 (ST)

R_3 (PQS)

R (ABC)

~~$A \to BC$~~

$B \to C$

$A \to C$

~~$AB \to C$~~

$A \to BC$

A^+ {ABC}

$A^+ = \{ABC\}$

$B \to C$

$B^+ = BC$

$B^+ = B$

$A \to C$

$A^+ = AC$

$A^+ = A$

$AB \to C$

$AB^+ = \{ABC\}$

$AB^+ = \{ABC\}$

Architecture:

Focuses on managing design and administration of software that manages the actual data in the application.

→ Affects the performance

1st architecture

- Direct access to the database by the user
- Client, server & database are present in a single machine
  - Simple
  - Less cost
  - Easy to implement

2nd arch

Client server model

User present at client level links to database at server level

- The server level is responsible for query processing
- The user has the interface for the application, prog. to run
  - Son
  - Easy deployment
  - Less expense
  - Scalable
  - Easy to access

Third

- Each level has its own respon.
- The client/user cannot access the database directly
- There is an intermediate layer b/w client & the database
- Exchange the partial processed data b/w user & client
  - Security
  - Data Integrity
  - Scalability.

View of data:

View of data can be reffered as the view the abstract data from the database

Abstraction is hiding irrelevant details from the user.

(i) Physical:

→ lowest level of abst.

→ Describes how the data base is being stored in secondary memory devices like disk, tapes.

→ Most of user are unaware of the level

→ Defines method to access the abstract data

(ii) Conceptual

→ Intermediate layer/level of data abstraction.

→ Describes what type of data is being stored in the database & their relation

→ The store of data in the form of table

→ Database administrator decides what data to keep.

View: Highest level of

→ Meant for end user level, Here the user get the data by querying the query

→ Hides details

→ External level.

Data Independence:

Change of data in one level does not affect the data in another level.

Logical

→ Character of change in the conceptual schema that does not affect the external schema

→ Separate

→ Difficult to implement as it depends completely on the Logical str. of data

ex: Adding on new attribute

Physical

→ Change in internal schema without effecting the external schema

→ Easy to implement

→ Separate internal schema from conceptual schema

→ Changing the storage type

Instance,

→ Shows the data stored in the database at a particular instance of time.

→ Change freq

→ Data instance change with insertion, updation & deletion

→ Extension of Schema

Schema

→ Structure of the database

→ Doesn't change freq

→ Same for the whole database

→ Intension

→ Does not rep. the datatype.

Data Integrity:

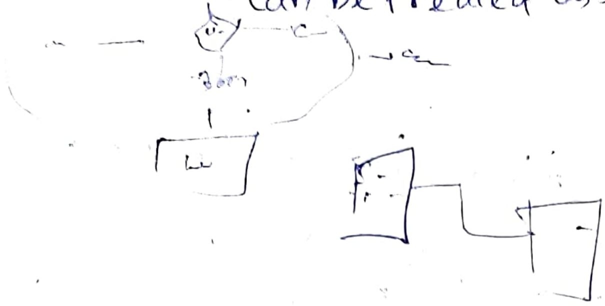
Set of predefined rules used to maintain the quality of information.

- Domain constraint (valid set of values)
- NOT NULL
- Entity Integrity constrain
- Key constraint
- Primary key constraint
- Reference Integrity constraint

Aggregation

→ Shows the relationship b/w two or more relationship set

→ Can be treated as single entity



Set Difference

R−S

All values of R are present but not in S

Natural Join

→ common at

→ Does not have any equicondn

→ Same as equijoin difference is in natural join the common attribute appears only once in the relational schema

View : Virtual set based on the result set of an SQL statement

Table & View :

(i) Table has physical existency, view is stored result set of SQL Query.

(ii) View is defined over the top of the table and does not hold data themselves.

(iii) View dependence on table but table is independent

Catalog info. is used to measure the cost of query process in the data base.
, Used to determine the selective factor for query process
, Based on the estimated cost the evaluation engine select the best plans & makes the execution & chooses the most efficient one.

Blocking Factor:

No. of tuples fit in one block

$$\frac{N_R}{B_R}$$

Selective Attribute of A

No. of tuples in A that satisfies the equality condⁿ of A

$$\frac{\text{No. of tuple}}{\text{Total}} \qquad \frac{N_R}{V(A, R)}$$

Measure of Query Cost:-

Execu- du
(i) de- ,
CPU
parallel or distr-- system

We chose disk access cost to compute cost que- over CPU cost as it is (state is slow)

Read pages from disk to memory
Write pages back to disk
So the transfer of no. of block to/from disk to seek is used for query cost.

(4) Join operation:- Used to join two or more data.
Most expensive as it is used to join two or more data.

(1) Nested

simple
iterates over each tuple in one relation & m-

$$b_R + (t_R \cdot b_S)$$

Index — Simple
use — reduce the risk of full scan.

$b_R + t_R$. cost of index lookup

Sort Merge
Efficient for large dataset & efficient when both relation are sorted.

$b_R \log(b_R) + b_S \log(b_S) + b_R + b_S$

Hash Join!
Efficient in equijoin with unsorted data set

$b_R + b_S$ + hash cost

Block-nested
↳ process relation based on the no. of blocks rather than per tuples

Heuristic based
simple, faster (Quick & efficient for small) Basic ops

Logical Optimisation
- Select
- Project
- Join reordering
- Selection

Physical
select the best
a) Index use.
speed retr
b) Join algorithm
c) Parallel Query exec

Semantic Query Optimisation
- eliminating redundant
- Exploiting constraint
- Const folding