Normalization of DBMS

Dr. Munesh Singh

Indian Institute of Information Technology
Design and Manufacturing,
Kancheepuram
Chennai-600127

March 7, 2019

Normalization

- The normalization process was first proposed by Codd in 1972
- This process takes a relation schema through a series of tests to certify whether it satisfies a certain normal form.

How many normal forms?

- Initially, Codd proposed three normal forms, which he called first, second, and third normal form.
- A stronger definition of third normal form is called Boyce-Codd normal form, was proposed later by Boyce Codd
- Later fourth and fifth normal form were proposed

Why normalization?

- It is the process of analyzing the given relation shcemea based on heir functional dependencies and primary keys to achieve the desirable properties of
  - Minimizing redundancy
  - Minimizing the insertion, deletion and update

First Normal Form?

1NF

- First normal form states that, the domain of an attribute must include only atomic values.
- Atomic?

Person

PID	fname	lname	Phone
p1	n1	l1	21,22,23
p2	n2	l2	24
p3	n3	l3	25
p4	n4	l4	26

Phone

Phoneid	Phone	PID
ph1	21	p1
ph2	22	p1
ph3	23	p1
ph4	24	p2
ph5	25	p3
ph6	26	p4

Second Normal Form in DBMS

2NF

- Second normal form (2NF) is a normal form used in database normalization
- 2NF was originally defined by E.F. Codd
- A table that is in first normal form (1NF) must meet additional criteria if it is to qualify for second normal form
- Second normal form states that it should meet all the rules for 1NF and there must be no partial dependences of any of the columns on the primary key

Second Normal Form in DBMS

2NF Example

- **Table: Student**
  Columns:**Student_id**,Name, Email, Phone, **Batch_id**,start_date, course, batch_size, batch_time, faculty
- **NOTE:**
- If table has only one primary key, then no need to look for 2NF.
- 2NF condition occurs when primary key is composite

Second Normal Form in DBMS

2NF Example

Student

SID	fname	email	Phone
s1	Amit	e1	21
s2	Amit	e2	24
s3	Raju	e3	25
s4	Ram	e4	26

Student

SID	BatchID	Name	Email	Phone	SD	Course	BS	BT	Faculty

Student

SID	Name	Email	Phone

Relation

SID	BatchID

Batch

BatchID	SD	Course	BS	BT	Faculty

Third Normal Form in DBMS

3NF

- Third normal form (3NF) is a normal form used in database normalization
- 3NF was originally defined by E.F.Codd
- Codd's definition states that a table is in 3NF if and only if both of the following conditions hold
  - The relation R (table) is in second normal form (2NF)
  - Every non-prime attributes of R is non-transitively dependent on every key of R.

3NF Example

Faculty

PK

FID	Name	Dept	DeptHead
101	Ram	CS	Sonu
102	Amit	ME	Monu
103	Arun	EE	Ramesh

FID---->Dept x->y

DeptHead----> Dept y->z **Transitive coluser**

DeptHead---->FID x->z

Faculty

PK

FID	Name	Dept
101	Ram	CS
102	Amit	ME
103	Arun	EE

Department

Dept	DeptHead
CS	Sonu
ME	Monu
EE	Ramesh

Functional Dependency

- It is an association between two attributes of the same table (relation)
- $x --> y$ if it can take only one value for a given value of the attributes upon which it is functionally dependent

Functional Dependency

Examples

R1 A->B

A	B
1	6
2	7
3	8
4	9

R2 A->B ?

A	B
1	6
2	7
3	8
2	9

R

A	B	C
1	2	3
2	2	3
3	3	3

Which FD correct?

A->B
B->C
BC->A
AC->B

R

X	Y	Z
1	4	3
1	5	3
4	6	3
3	2	2

Which FD correct?

(a) XY->Z, Z->Y
(b) XZ->X, Y->Z
(c) YZ->X, Z->X
(d) XZ->Y, Y->Z

Classification of Functional Dependency

Normalization

- **Trivial F.D** Function dependency $X- > Y$ is trivial if and only if $Y \subseteq X$ e.g AB-¿BC
- **Non-trivial F.D** If there is at-least one attribute in RHS that is not part of the LHS e.g AB-¿BC
- **Fully Functional Dependency** Given R and F.D $X- > Y$, then Y is full functional dependent on X if there is no Z, where Z is proper subset of X such that $Z- > Y$ e.g $AB- > C$ and $A- > C$ (not fully functional dependent)

Classification of Functional Dependency

Normalization

- **Partial Dependency** Given a relation R with F.D f defined on the attributes of R and K as a candidate key, if X is a proper subset of K and if $X- > A$, then A is said to be partially dependent on K
- **Transitive Dependency** $X- > Y$, $Y- > A$, then A is transitively depends on X.

Inference Rules/Closure Propoerties/Armstrong's Axioms

Normalization

- **Reflexivity** if $Y \subseteq X$, then $X - > Y$ e.g AB-¿A
- **Augmentation** if $X - > Y$, then $XZY - > YZ$.
- **Transitivity** if $X - > Y$ and $Y - > Z$, then $X - > Z$.
- **Union** if $X - > Y$ and $X - > Z$, then $X - > YZ$.
- **Decompositions** if $X - > YZ$, then $X - > Y$, $X - > Z$.

Closure Set of FD's (F^+)

Normalization

- F^+ is the set of all FDs that can be determined using the fiven set of functional dependency (F)
  Total No of possible FDs in R(n attributes)$=2^{2^n}$
- **Closure Set of Attributes (X^+)** is the set of all attributes that can be determined using the given set X(attributes)
  $A->B, B->C$
  A^+=(ABC), B^+=(BC)
- **Application of Attribute Closure:** if (X^+) contains all attributes of a relation then X is called Super-key of R.
  Eg. R(ABCDE)
  $\{A->BC, CD->E, B-D, E->A\}$ Find candidate keys of R

Closure Set of FD's (F^+)

Examples

- **Eg. R(ABCDE)**
  $\{AB- > C, CD- > E, DE- > B\}$ Is AB a candidate key? if not is ABD?

Finding Additional FD's

- **Eg. R(ABCDE)**
  $\{A- > BC, B- > CD, D- > AB\}$ Find whether $AD- > C$ is possible or not?
- Find the closure of (AD) using given functional dependency, if c comes then it satisfied
- **Eg. R(ABCDE)**
  $\{A- > B, A- > C, CD- > E, B- > D, E- > A\}$ which of the following FD is not implies by the above set?
- $CD- > AC$, $BD- > CD$, $BC- > CD$, $AC- > BC$

Equivalent FDs

Examples

- How to check for equivalent FDs:
    1. F=$\{A->C, AC->D, E->AD, E->H\}$
    2. G=$\{A->DC, E->A\}$
       Check whether both FD,s are equivalent or not?
    3. Checking for FDs in G relation. $A->DC$ $(A)^+$=ACD
    4. Checking for FDs in G relation $E->A$ $(E)^+$=EADH
    5. Checking for FDs in F relation $A->C$ $(A)^+$=AC
    6. Checking for FDs in F relation $AC->D$ $(AC)^+$=ACD
    7. Checking for FDs in F relation $E->A$ $(E)^+$=ADC
    8. Checking for FDs in F relation $E->H$ $(E)^+$=
    9. These functional dependency are not equivalent

Decomposition

Examples

- A relation 'R' can be decomposed into a collection of relation scheme to eliminate some of the anomalies in original relation R.
  1. Lossless Join Decomposition: Let R is a relation and has set of FDs 'F' over R. The decomposition of R into R1 and R2 is lossless w.r.t if R1$\bowtie$R2=R
  2. Lossy Decomposition: Contains extra tuples.
- Note: R1,R2 if (R1$\bigcup$R2) forms a superkey in either R1 or R2, then it is lossless.
- Solved Examples on Decomposition
  1. R(ABC), decomposed in R1(AB) and R2(BC) check whether lossless or lossy. Given FD R$\{A->B\}$
  2. R1$\bigcup$R2 $=$ B here B is not a key in either R1 or R2, so it is lossy decomposition

Decomposition

Examples

- Solved Examples on Decomposition

1. R(ABC), decomposed in R1(AB) and R2(AC) check whether lossless or lossy. Given FD R$\{A->B\}$
2. R(ABC), decomposed in R1(ABC) and R2(CD) check whether lossless or lossy. Given FD R$\{A->B, A->C, C->D\}$
3. R(ABC), decomposed in R1(ABC),R2(BCD), R3(CDE) check whether lossless or lossy. Given FD R$\{AB->CD, A->E, C->D\}$

if a col has two alpha and it derive to other attribute,
where one alpha is given we change it
This goes till row has all alpha,then lossless decompostion

	A	B	C	D
R1	α	α	α	α
R2			α	α

How to find candidate key in a relation

- Candidate Keys
  1. R(ABCD), Find all candidate keys of relation R. Given FDs $\{BC->A, AD->B, CD->B, AC->D\}$
  2. Look for the attributes which is not present in RHS
  3. Find closure of attributes/set of attributes which are not present in RHS if closure contains all attributes then it will be the only key
  4. Find combination of remaining attributes with attributes not present in RHS
  5. For above FDs closures $(AC)^+$,$(BC)^+$,$(CD)^+$

Normalization 2NF & 3NF

- Based on the concept of full functional dependency (FFD)
- Relation 'R' is in 2NF, if every non-prime attribute of R is fully functional dependent on the key of R
- No no-prime attribute should be determine by the part of candidate key

- A relation scheme (R) is in 3NF if it satisfies 2NF and no non-prime attribute of R is transitively dependent on key of 'R'
- A relation is in 3NF if F.D $X->Y$ satisfy any one of the following condition
  1. $X->Y$ is a trivial F.D i.e Y⊂X AB$->$B
  2. if $X->Y$, then X is a superkey AD$->$C
  3. if $X->Y$, then (Y-X) is a prime (Attribute in Y but not in X)

Keys Attributes

Examples

- **Prime Attributes**
  Attributes which are part of C.K
- **Non-Prime Attributes**
  Are not part of C.K
- Some examples:
  1. R(ABCD)$\{AB->C, B->D\}$ Find Normal Form?
  2. R(ABC)$\{A->B, B->C\}$ Find Normal Form?
  3. R(ABCD)$\{AB->C, A->D\}$ Decompose into 2NF?
  4. R(ABCDE)$\{AB->C, A->D, B->E\}$ Decompose into 2NF?

Keys Attributes

Examples

- Some examples:
  1. R(ABCD)$\{A- > B, B- > C\}$ Find Normal Form?
  2. R(ABC)$\{AB- > C, B- > D, D- > E\}$ Decompose it into 3NF?
  3. R(ABCD)$\{AB- > C, C- > A\}$ Which Normal Form?
  4. R(ABCDE)$\{AB- > C, A- > D, B- > E\}$ Decompose into 2NF?

BCNF Normalization

BCNF

- Extension of 3NF on strict terms
- A relation is in BCNF if at least one of the condition holds:
- Every BCNF is in 3NF but vice-versa is not true
  - X− >Y is a trivial F.D
  - X− >Y, then X is a superkey
- Some examples:
  1. R(ABC)$\{AB->C, C->A\}$ R is in 3NF or BCNF?
  2. R(ABCDEFGHIJ)$\{AB->C, A->DE, B->F, F->GH, D->IJ\}$ Decompose it into BCNF

Practise of Normalization

- Some examples:
  1. R(ABCDEF)$\{AB->C, C->DE, E->F, F->A\}$ check highest normal form?
  2. R(ABCDEF)$\{AB->C, C->D, C->E, E->F, F->A\}$ check the highest normal form
  3. Registration($\underline{\text{rollno}}$,course), FD=$\{rollno->course\}$
  4. Registration($\underline{\text{rollno}}$,$\underline{\text{course}}$,email), FD=$\{\underline{rollno, course}->email, email->rollno\}$
  5. Registration($\underline{\text{rollno}}$,$\underline{\text{course}}$,marks,grade), FD=$\{\underline{rollno, course}->marks, grade, marks->grade\}$
  6. Registration($\underline{\text{rollno}}$,$\underline{\text{course}}$,credit), FD=$\{\underline{rollno, course}->credit, courseid->credit\}$

Practise of Normalization

- Some examples:
  1. R(ABCDEF)$\{C- > F, E- > A, EC- > D, A- > B\}$ check highest normal form?
  2. R(ABCDE)$\{A- > B, BC- > D, E- > C, D- > A\}$ check the highest normal form
  3. R(ABCDEPG)$\{AB- > CD, DE- > P, C- > E, P- > C, B- > G\}$ check the highest normal form
  4. R(ABCDEFGH)$\{CH- > G, A- > BC, B- > CFH, E- > A, F- > G\}$ check the highest normal form
  5. R(ABCDEF)$\{AB- > CD, CD- > EF, BC- > DEF, D- > B, CE- > F\}$ check the highest normal form
  6. Student Performance(name,courseNo,RollNo,grade) $\{name, courseNo- > grade, rollNo, courseNo- > grade, name- > rollNo, rollNo- > name\}$