Database Management Systems

Dr. Santhosh Manikonda
Department of CSE
School of Engineering

Malla Reddy University, Hyderabad

Syllabus
UNIT – III
Module 2

- Basic of Functional Dependencies and Normalization for Relational Databases:
- Functional Dependencies
- Normal Forms Based Primary Keys
- General Definitions of Second and Third Normal Forms
- Boyce-Codd Normal Forms
- Multivalued Dependency
- Fourth Normal Form.

Inference Rules for Functional Dependencies

- **Reflexivity**: If $Y \subseteq X$ then FD: $X \rightarrow Y$
  - This is called a trivial dependency
  - *Ex: sname, address $\rightarrow$ address*
- **Augmentation**: If FD: $X \rightarrow Y$ then FD: $XW \rightarrow YW$
  - Ex: *cid $\rightarrow$ cname then cid, sid $\rightarrow$ cname,sid*
- **Transitivity**: If FD: $X \rightarrow Y$ and FD: $Y \rightarrow Z$ then FD: $X \rightarrow Z$
  - sid $\rightarrow$ cid and cid $\rightarrow$cname, then sid $\rightarrow$ cname

Inference Rules for Functional Dependencies

- Union: If FD: $X \rightarrow Y$ and FD: $X \rightarrow Z$ then FD: $X \rightarrow YZ$
- Pseudo-transitivity: If FD: $X \rightarrow Y$ and FD: $WY \rightarrow Z$ then FD $XW \rightarrow Z$
- Decomposition: If FD $X \rightarrow YZ$ then FD: $X \rightarrow Y$, $X \rightarrow Z$
- Composition: If FD $X \rightarrow Y$ and FD: $Z \rightarrow W$ then $XZ \rightarrow YW$

Closure set – Prime, Non-Prime attribute

- R = {A, B, C, D}
- FDs: {$A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow A$}
- Find all possible Candidate Keys

- Prime Attribute: Attributes that are used in the formation of Candidate Key id are known as Prime Attributes
- Non - Prime Attribute: Attributes that are NOT used in the formation of Candidate Key are known as Non - Prime Attributes
- Find all Prime and Non-Prime Attributes.

Example problems

- R = {A, B, C, D, E}
- FDs: {$A \rightarrow B$, $BC \rightarrow D$, $E \rightarrow C$, $D \rightarrow A$}
- Find all possible Candidate Keys
- Find all Prime and Non-Prime Attributes.

First Normal Form (1NF)

- Rules
- The table should not contain any multivalued attributes
- The table should not contain any composite attribute – each cell should contain atomic values
- *Every column must have only one type of data*
- *Two columns cannot have same name*
- *Order of rows/columns is not important*

SID	Sname	Course
1001	Ram	Python, Java
1002	Raj	Java
1003	Rahul	Python, DBMS

Not in 1NF

First Normal Form (1NF)

- Converting the table to 1NF

SID	Sname	Course
1001	Ram	Python, Java
1002	Raj	Java
1003	Rahul	Python, DBMS

Not in 1NF

SID	Sname	Course
1001	Ram	Python
1001	Ram	Java
1002	Raj	Java
1003	Rahul	Python
1003	Rahul	DBMS

In 1NF

First Normal Form (1NF)

- Converting the table to 1NF

SID	Sname	Course
1001	Ram	Python, Java
1002	Raj	Java
1003	Rahul	Python, DBMS

Not in 1NF

SID	Sname	Course 1	Course 2
1001	Ram	Python	Java
1002	Raj	Java	NULL
1003	Rahul	Python	DBMS

In 1NF

First Normal Form (1NF)

- Converting the table to 1NF

SID	Sname	Course
1001	Ram	Python, Java
1002	Raj	Java
1003	Rahul	Python, DBMS

Not in 1NF

R{SID, Sname, Course}

SID	Sname
1001	Ram
1002	Raj
1003	Rahul

SID	Course
1001	Python
1001	Java
1002	Java
1003	Python
1003	DBMS

In 1NF

R1{SID, Sname,}

R2{SID, Course}

Second Normal Form (2NF)

- Rules
- The table must be in First Normal From(1NF)
- There are no Partial Dependencies in the relation
  - Partial Dependency: Subset of Candidate Key determines Non – Prime Attributes
  - $\{\subseteq CK\} \rightarrow \{\text{Non – Prime Attributes}\}$
- Super key :Set of attributes whose closure contains all attributes of a relation
- Candidate key: If no proper subset of super key is a super key, then that SK is a CK

- R{A, B, C, D, E, F}
- FDs: $\{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E\}$

Second Normal Form (2NF)

- Rules
- The table must be in First Normal From(1NF)
- There are no Partial Dependencies in the relation
  - Partial Dependency: Subset of Candidate Key determines Non – Prime Attributes
  - $\{\subseteq CK\} \rightarrow \{\text{Non – Prime Attributes}\}$
- Super key :Set of attributes whose closure contains all attributes of a relation
- Candidate key: If no proper subset of super key is a super key, then that SK is a CK

- R{A, B, C, D, E, F}
- FDs: {A→B, B→C, C→D, D→E}
- CK = { AF }
- Prime Attributes: PA = {A, F}
- Non – Prime Attributes: NPA = {B, C, D, E}
- A → B – Partial Dependency, therefore relation R is NOT in 2NF

Second Normal Form (2NF) - Example

- R{A, B, C, D}
- FDs: {AB→CD, C→A, D→B}
- CK = { }
- Prime Attributes: PA = { }
- Non – Prime Attributes: NPA = { }

Second Normal Form (2NF) - Example

- R{A, B, C, D}
- FDs: {AB→CD, C→A, D→B}
- CK = { AB, BC, CD, AD }
- Prime Attributes: PA = {A, B, C, D}
- Non – Prime Attributes: NPA = {Φ}

- NO Partial Dependencies, therefore relation R is in 2NF

Second Normal Form (2NF) - Example

- R{A, B, C, D}
- FDs: {A→B, B→C, C→D}
- CK = {   }
- Prime Attributes: PA = {   }
- Non – Prime Attributes: NPA = {   }

Second Normal Form (2NF) - Example

- R{A, B, C, D}
- FDs: {A→B, B→C, C→D}
- CK = {A}
- Prime Attributes: PA = {A}
- Non – Prime Attributes: NPA = {B, C, D}

- Proper subset of CK is a null set.
- NO Partial Dependencies, therefore relation R is in 2NF

Third Normal Form (3NF)

- Rules
- The table must be in Second Normal From(2NF)
- There are NO Transitive Dependencies in the relation (for NPAs)
  - Transitive dependency: Non – Prime attributes determine other Non – Prime attributes
  - {Non – Prime Attribute} $\rightarrow$ {Non – Prime Attribute}

- R = {A, B, C, D}
- FDs: {A$\rightarrow$B, B$\rightarrow$C, C$\rightarrow$D}
- CK = { }
- PA = { }
- NPA = { }

Third Normal Form (3NF)

- Rules
- The table must be in Second Normal From(2NF)
- There are NO Transitive Dependencies in the relation (for NPAs)
  - Transitive dependency: Non – Prime attributes determine other Non – Prime attributes
  - {Non – Prime Attribute} → {Non – Prime Attribute}

- R = {A, B, C, D}
- FDs: {A→B, B→C, C→D}
- CK = {A}
- PA = {A}
- NPA = {B, C, D}
- B→C and C→D (Transitive Dependencies) – Therefore relation R is not in 3NF

Third Normal Form (3NF) - Example

- R = {A, B, C, D, E, F}
- FDs: {AB→CDEF, BD→F}
- CK = { }
- PA = { }
- NPA = { }

Third Normal Form (3NF) - Example

- R = {A, B, C, D, E, F}
- FDs: {AB→CDEF, BD→F}
- CK = { AB}
- PA = { A, B }
- NPA = { C, D, E, F}
- BD→F (Transitive Dependency) – Therefore relation R is not in 3NF

Boyce Codd Normal Form (BCNF)

- Rules
- The table must be in Third Normal From(3NF)
- For each non-trivial FD X→ Y, X must be a super key

- R = {A, B, C}
- FDs: {A→B, B→C, C→A}
- CK = { }
- PA = { }
- NPA = { }

Boyce Codd Normal Form (BCNF)

- Rules
- The table must be in Third Normal From(3NF)
- For each non-trivial FD X→ Y, X must be a super key

- R = {A, B, C}
- FDs: {A→B, B→C, C→A}
- CK = {A, B, C}
- PA = {A, B, C}
- NPA = {Φ}
- A→B, B→C, C→A, all LHS attributes are SK, therefore relation R is in BCNF

Boyce Codd Normal Form (BCNF)

- R = {A, B, C, D, E}
- FDs: {A→BCDE, BC→ACE, D→E}
- CK = { }
- PA = { }
- NPA = { }

Boyce Codd Normal Form (BCNF)

- R = {A, B, C, D, E}
- FDs: {A→BCDE, BC→ACE, D→E}
- CK = {A, BC}
- PA = {A, B, C}
- NPA = {D, E}

- A→BCDE, BC→ACE, all LHS attributes are SK
- D → E, LHS is not a SK, therefore, relation R is not in BCNF

Boyce Codd Normal Form (BCNF)

- R = {A, B, C, D, E}
- FDs: {AB→CDE, D→A}
- CK = { }
- PA = { }
- NPA = { }

Boyce Codd Normal Form (BCNF)

- R = {A, B, C, D, E}
- FDs: {AB→CDE, D→A}
- CK = {AB, BD}
- PA = {A, B, D}
- NPA = {C, E}

- AB→CDE, all LHS attributes are SK
- D → A, LHS is not a SK, therefore, relation R is not in BCNF

Fourth Normal Form (4NF)

- Rules
- The table must be in BCNF
- The relation should not have any Multi-valued Dependency
  - For a dependency A → B, if for a single value of A, multiple value of B exists, then the relation may have multi-valued dependency
  - *Also, a table should have at-least 3 columns for it to have a multi-valued dependency*
  - *And, for a relation R{A,B,C}, if there is a multi-valued dependency between, A and B, then B and C should be independent of each other*

Fourth Normal Form (4NF)

s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
2	C#	Cricket
2	Php	Hockey

- R = {s_id, course, hobby}
- FDs: {s_id→→course, s_id→→hobby}

s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
1	Science	Hockey
1	Maths	Cricket

- R1 = {s_id, course}
- R2 = {s_id, hobby}

Fourth Normal Form (4NF)

R

Car_model	Mfg_year	Color
H001	2017	Metallic
H001	2017	Green
H005	2018	Metallic

R1

Car_model	Mfg_year
H001	2017
H001	2017
H005	2018

R2

Car_model	Color
H001	Metallic
H001	Green
H005	Metallic

Lossless Decomposition

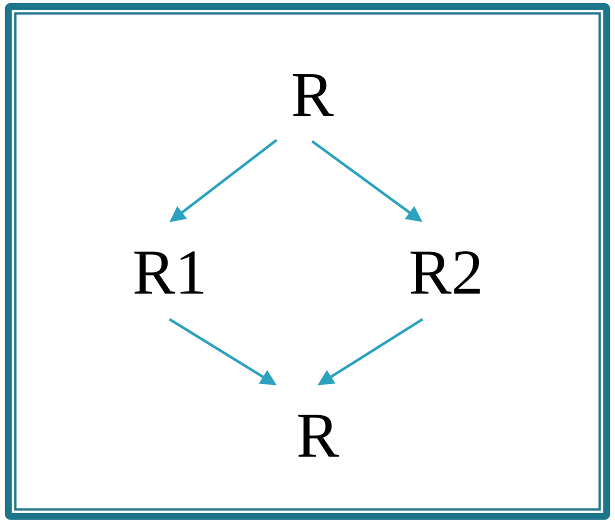
- This type of decomposition ensures no extra tuples are generated and not tuples are lost.
- If a relation (R) is decomposed into two relations R1 and R2, it will be lossless if:

1) attributes{R1} ∪ attributes{R2} = attributes {R}
2) attributes {R1} ∩ attributes {R2} ≠ φ
3) attributes {R1} ∩ attributes {R2} → attributes {R1}

   (meaning: common attribute must be candidate key)

   or

   attributes {R1} ∩ attributes {R2} → attributes {R2}



Lossless Decomposition

R

A	B	C	D
1	a	p	x
2	b	q	y

R1

A	B
1	a
2	b

R2

C	D
1	a
2	b

R1 ⋈ R2

A	B	C	D
1	a	p	x
1	a	q	y
2	b	p	x
2	b	q	y

Lossless Decomposition

R

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

R → R1{A, B}
R2{B, C}

R1

A	B
1	2
2	2
3	3

R2

B	C
2	1
2	2
3	2

Lossless Decomposition

R

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

R1

A	B
1	2
2	2
3	3

R2

B	C
2	1
2	2
3	2

```
SELECT R2.C FROM R2 NATURAL JOIN R1 WHERE R1.A=1;
```

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

Lossless Decomposition

```
SELECT R2.C FROM R2 NATURAL JOIN R1 WHERE R1.A=1;
```

R

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

In original table there is only one row for A=1

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

In the table after decomposition and join operation there are two rows for A = 1

This is a lossy decomposition join

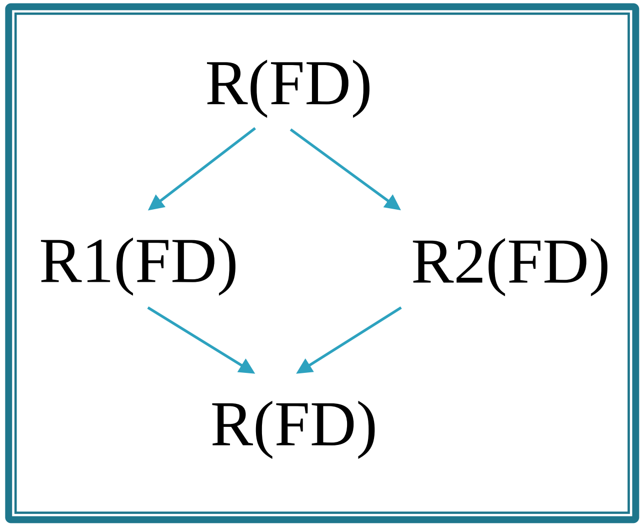
Dependency preserving decomposition

- If a relation R, having functional dependency (FD) set F, is decomposed into R1 and R2 having FD set F1 and F2, then:

$$FD1 \subseteq FD^{+}$$

$$FD2 \subseteq FD^{+}$$

$$(FD1 \cup FD2)^{+} \subseteq FD^{+}$$



Dependency preserving decomposition - Example

1. R{ABC}
   FD:
   $A \rightarrow B$
   $B \rightarrow C$
   $C \rightarrow A$

R1{AB}	R2{BC}
FD1:	FD2:
$A \rightarrow B$	$C \rightarrow B$
$B \rightarrow A$	

$(FD1 \cup FD2)^+ \subseteq FD^+$

Dependency preserving decomposition - Example

1. R{ABCD}
   FD:
   A→B
   B→C
   C→D
   D→B

R1{AB}
~~A→A~~
~~B→B~~
A→B
~~B→A~~
Check A^+ and B^+

R2{BC}
FD2:
B→C
C→B

R2{BD}
FD2:
B→D
D→B

$(FD1 \cup FD2)^+ \subseteq FD^+$

A→B
B→C
C→B ----- $C^+ = CBD$
B→D
D→B

A→B
B→C
C→D
D→B