Introduction to Database Systems

Normalization

Lee Mong Li

National University of Singapore

Recap: Functional Dependencies

Definition: X → Y

- If two tuples agree on the values of the attributes in X, then they agree on the values of the attributes in Y.
- **Example:** {position} → {salary}

If two employees have the same position, they also have the same salary.

Types of FDs:

- Trivial $\mathbf{Y \subset X}$
- Non-trivial $\mathbf{Y \not\subset X}$
- Completely non-trivial $\mathbf{Y \cap X = \varnothing}$
- **Example:** Given a relation R(A,B,C,D,E) with

$$F = \{AB \to CE, CD \to D, ACE \to BC, D \to E\}$$

How many FDs of each type are there?

Armstrong's Axioms

Let X, Y, Z be subsets of the scheme of relation R

- **Reflexivity:** If $Y \subset X$, then $X \rightarrow Y$
- **Augmentation:** If $X \rightarrow Y$, then $X \cup Z \rightarrow Y \cup Z$
- **Transitivity:** If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

Example: Given a relation R(A,B,C,D,E) with

F = { AB → ABCE, C → BC }

Show that ACD is a **superkey** using the axioms.

In other words, show that ACD → ABCDE

Armstrong's Axioms

Given a relation R(A,B,C,D,E) with

F = { AB → ABCE, C → BC }

Show that ACD → ABCDE

1. C → BC (Given)
2. AC → ABC (Augmentation with (1))
3. AB → ABCE (Given)
4. ABC → ABCE (Augmentation with (3))
5. AC → ABCE (Transitivity with (2) & (4))
6. ACD → ABCDE (Augmentation with (5))

Attribute Closure

For a set X of attributes, we call the **closure of X (*w.r.t. a set of FDs*)**, denoted by X^+, the maximum set of attributes such that $X \rightarrow X^+$.

Example: Given a relation R(A, B, C, D, E) with

$$F = \{A \rightarrow B,\ \ A \rightarrow C,\ \ B \rightarrow D\}$$

Is A in A+? Is B in A+? Is C in A+? Is D in A+? Is E in A+?

Algorithm:

- Start with the given set X.
- Find and add new attributes which can be determined by any subset of X, until no more attributes can be added.

*Useful for (a) finding **candidate keys** and (b) proving **equivalence of sets of FDs***

Candidate Keys

Consider the relation R(A,B,C,D,E) with the FDs:

$AB \rightarrow CDE$

$AC \rightarrow BDE$

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

Find all **candidate keys**

Candidate Keys

$AB \rightarrow CDE$, $AC \rightarrow BDE$, $B \rightarrow C$, $C \rightarrow B$, $C \rightarrow D$, $B \rightarrow E$

$A^+ =$ A
$B^+ =$ BCDE
$C^+ =$ BCDE
$D^+ =$ D
$E^+ =$ E

$AB^+ =$ <u>ABCDE</u>
$AC^+ =$ <u>ABCDE</u>
$AD^+ =$ AD
$AE^+ =$ AE
$BC^+ =$ BCDE
$BD^+ =$ BCDE
$BE^+ =$ BCDE
$CD^+ =$ BCDE
$CE^+ =$ BCDE
$DE^+ =$ DE

$ADE^+ =$ ADE
$BCD^+ =$ BCDE
$BCE^+ =$ BCDE
$BDE^+ =$ BCDE
$CDE^+ =$ BCDE

$BCDE^+ =$ BCDE

Equivalence of 2 Sets of FDs

Algorithm based on attribute closure

Intuition:

- Given two sets of FDs F and G
- If we have in F, A→B and A^+ = ABC, then we know that A→A, A→B and A→C is in F^+
- And if we have in G, A^+ = AB, then we know that there is some FD (i.e., A→C) in F^+ but not in G^+
- Hence, F and G are not equivalent.

Minimal Cover

A set of FDs F is a minimal cover for a set of FDs G if and only if

- Every FD in F is of the form X $\rightarrow$ A where X is a set of attributes, A is a **single attribute** and X has **no redundant attributes**
- There are **no redundant FDs** in F
- F is equivalent to G, that is, **$F^+ = G^+$**

Minimal Cover

Consider the relation R(A,B,C,D,E) with FDs:

$AB \rightarrow CDE$

$AC \rightarrow BDE$

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

Find a **minimal cover** for the FDs

Minimal Cover Step (1) *Single Attribute on RHS of FD*

$AB \rightarrow CDE$	$AB \rightarrow C$
	$AB \rightarrow D$
	$AB \rightarrow E$
$AC \rightarrow BDE$	$AC \rightarrow B$
	$AC \rightarrow D$
	$AC \rightarrow E$
$B \rightarrow C$	$B \rightarrow C$
$C \rightarrow B$	$C \rightarrow B$
$C \rightarrow D$	$C \rightarrow D$
$B \rightarrow E$	$B \rightarrow E$

Minimal Cover Step (2) *Remove Redundant Attributes*

$AB \rightarrow C$	$AB \rightarrow C$
$AB \rightarrow D$	$B \rightarrow D$
$AB \rightarrow E$	$AB \rightarrow E$
$AC \rightarrow B$	$AC \rightarrow B$
$AC \rightarrow D$	$AC \rightarrow D$
$AC \rightarrow E$	$AC \rightarrow E$
$B \rightarrow C$	$B \rightarrow C$
$C \rightarrow B$	$C \rightarrow B$
$C \rightarrow D$	$C \rightarrow D$
$B \rightarrow E$	$B \rightarrow E$

Can we replace $AB \rightarrow D$ with $B \rightarrow D$ or $A \rightarrow D$?

Yes!

$B \rightarrow C$ and $C \rightarrow D$ by transitivity give $B \rightarrow D$

Minimal Cover: Step (2)

$AB \rightarrow C$

$AB \rightarrow D$

$AB \rightarrow E$

$AC \rightarrow B$

$AC \rightarrow D$

$AC \rightarrow E$

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

$B \rightarrow C$

$B \rightarrow D$

$B \rightarrow E$

$C \rightarrow B$

$C \rightarrow D$

$C \rightarrow E$

Minimal Cover Step (3)

Remove Redundant FDs

B→C

B→D

B→E

C→B

C→D

C→E

Can we eliminate B→C ?

Compute B+ without B→C

B+=BDE

Does not contain C

No!

Minimal Cover Step (3)

Remove Redundant FDs

B→C	B→C
B→D	B→E
B→E	C→B
C→B	C→D
C→D	C→E
C→E	

Can we eliminate B→D ?

Compute B+ without B→D

B+=BCDE

B+ contains D

Yes!

Minimal Cover Step (3)

Remove Redundant FDs

$B \rightarrow C$	$B \rightarrow C$
$B \rightarrow E$	$C \rightarrow B$
$C \rightarrow B$	$C \rightarrow D$
$C \rightarrow D$	$C \rightarrow E$
$C \rightarrow E$	

Can we eliminate $B \rightarrow E$?

Compute B+ without $B \rightarrow E$

B+=BCDE

B+ contains E

Yes!

Minimal Cover

Extended Minimal Cover Step (4)

$$B \rightarrow C \qquad\qquad B \rightarrow C$$

$$\left.\begin{array}{l} C \rightarrow B \\ C \rightarrow D \\ C \rightarrow E \end{array}\right\} \quad C \rightarrow BDE$$

Remark: Minimal Cover Computation

- $F = \{ABCD \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$
- A minimal cover for F is $\{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$
- Can we remove redundant FDs *before* redundant attributes?

Initialize G = {ABCD→E, E→D, A→B, AC→D}

1. Remove redundant FDs

ABCD→E is non-redundant since ABCD+ = ABCD wrt G – {ABCD →E}

E→D is non-redundant since E+ = E wrt G – {E→D}

A→B is non-redundant since A+ = A wrt G – {A →B}

AC→D is non-redundant since AC+ = AC wrt G – {AC →D}

2. Remove redundant attributes

A in ABCD→E is non-redundant since BCD+ (wrt G) = BCD

B in ABCD→E is redundant since ACD+ (wrt G) = ABCDE

G = {ACD→E, E→D, A→B, AC→D}

C in ACD→E is non-redundant since AD+ (wrt G) = ABD

D in ACD→E is redundant since AC+ (wrt G) = ABCDE

G = {AC→E, E→D, A→B, AC→D}

A in AC→D is non-redundant since C+ (wrt G) = C

C in AC→D is non-redundant since A+ (wrt G) = AB

Output ***G = {AC→E, E→D, A→B, AC→D} which is not a minimal cover***

Normalization

Process of Decomposition

- Break down a relation into smaller relations such that all attributes are still present in at least one of the smaller relations.

Lossless / Lossy

- Whether the original relation can be recovered by combining the smaller relations.

Dependency preserving

- Whether all the dependencies still exist in the smaller relations.

Decomposition

Given a relation R(A,B,C) with {A} → {B,C}, which of the following are decompositions?

R1(A), R2(B)	No
R1(A,C), R2(B)	Yes
R1(A,B), R2(B,C)	Yes
R1(A,B), R2(A,C)	Yes
R1(A,C), R2(B,C)	Yes

Decomposition

Given a relation R(A,B,C) with {A} → {B,C}, which are lossless decompositions?

R1(A), R2(B)	No
R1(A,C), R2(B)	No
R1(A,B), R2(B,C)	No
R1(A,B), R2(A,C)	
R1(A,C), R2(B,C)	

A	B	C
1	Bill	Gates
2	Bill	Clinton

A	B
1	Bill
2	Bill

B	C
Bill	Gates
Bill	Clinton

A	B	C
1	Bill	Gates
1	Bill	Clinton
2	Bill	Gates
2	Bill	Clinton

Decomposition

Given a relation R(A,B,C) with {A} → {B,C}, which are lossless decompositions?

R1(A), R2(B)	No
R1(A,C), R2(B)	No
R1(A,B), R2(B,C)	No
R1(A,B), R2(A,C)	Yes
R1(A,C), R2(B,C)	

A	B	C
1	Bill	Gates
2	Bill	Clinton

A	B
1	Bill
2	Bill

A	C
1	Gates
2	Clinton

A	B	C
1	Bill	Gates
2	Bill	Clinton

Decomposition

Given a relation R(A,B,C) with {A} → {B,C}, which are lossless decompositions?

R1(A), R2(B)	No
R1(A,C), R2(B)	No
R1(A,B), R2(B,C)	No
R1(A,B), R2(A,C)	Yes
R1(A,C), R2(B,C)	No

A	B	C
1	Steve	Jobs
2	New	Jobs

A	C
1	Jobs
2	Jobs

B	C
Steve	Jobs
New	Jobs

A	B	C
1	Steve	Jobs
1	New	Jobs
2	Steve	Jobs
2	New	Jobs

Decomposition

Given a relation R(A,B,C) with {A} → {B,C}, which are dependency preserving decompositions?

R1(A), R2(B)	**No**	
R1(A,C), R2(B)	**No**	
R1(A,B), R2(B,C)	**No**	
R1(A,B), R2(A,C)	**Yes**	R1(A,B) **A → B** R1(A,C) **A → C**
R1(A,C), R2(B,C)	**No**	

Normal Forms

Boyce-Codd Normal Form (BCNF)

- For any FD $X \rightarrow Y$ on a relation R, either the FD is trivial (i.e., $Y \subset X$) or X is a superkey of R

Third Normal Form (3NF)

- For any FD $X \rightarrow Y$ on a relation R, either the FD is trivial (i.e., $Y \subset X$), or X is a superkey, or Y is part of some candidate key

Example:

- Given a relation R(A, B, C, D, E, G) with FDs

$$F = \{ AB \rightarrow CDEG, \quad E \rightarrow G \}$$

- Candidate key: AB
- $E \rightarrow G$ violates both BCNF and 3NF

Decomposition into BCNF

Given a set of relations and a set of FDs

1. **Find FD $X \rightarrow Y$ on a relation R that violates BCNF property**
2. **Decompose R into $(R - X^+ + X)$ and X^+ (and project the FDs)**
3. **Repeat Steps 1 & 2 until all relations are in BCNF**

Note: It is sometimes useful to find the minimal cover first.

Example: R(A, B, C, D, E, G) with $F = \{ AB \rightarrow CDEG,\ E \rightarrow G \}$

- **$E \rightarrow G$ violates BCNF**
- **BCNF decomposition into:**

  R1(E, G) with $F1 = \{ E \rightarrow G \}$

  R2(A, B, C, D, E) with $F2 = \{ AB \rightarrow CDE \}$

Decomposition into 3NF

Given a set of relations and a set of FDs

1. **Find the minimal cover G.**
2. **For every FD X → Y in G,**
   If $X \cup Y$ is not in the existing relations, then add a relation R with $X \cup Y$ (and project the FDs)
3. **If there is no relation containing the key, add a relation with the key.**

Note 1: The given relations are ignored.
Note 2: The extended minimal cover can be used, too.

Example: R(A, B, C, D, E, G) with F = { AB → CDEG, E → G }

- **E → G violates 3NF**
- **G = { AB → C, AB → D, AB →E, E → G}**
- **Extended minimal cover G' = { AB → CDE, E → G }**
- **3NF synthesis: { R1(<u>A,B</u>,C,D,E), R2(<u>E</u>,G) }**

BCNF Decomposition

Consider the relation R(A,B,C,D,E) with FDs:

$AB \rightarrow CDE$

$AC \rightarrow BDE$

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

Give a BCNF decomposition of R.

BCNF Decomposition

$AB \rightarrow CDE$

$AC \rightarrow BDE$

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

$\equiv$

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

??? Not in BCNF

Candidate keys: AB, AC

BCNF Decomposition

Decompose with $B \rightarrow C$:

$B+ = BCDE$

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

project

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

Candidate keys

R_1: AB

R_2: B, C

BCNF Decomposition

R1(A,B)	$B \rightarrow C$	Super key
R2(B,C,D,E)	$C \rightarrow B$	Super key
	$C \rightarrow D$	Super Key
Candidate keys	$B \rightarrow E$	Super Key

R1: AB

R2: B, C

Is it a dependency preserving decomposition?

3NF Decomposition

Consider the relation R(A,B,C,D,E) with FDs:

$AB \rightarrow CDE$

$AC \rightarrow BDE$

$B \rightarrow C$

$C \rightarrow B$

$C \rightarrow D$

$B \rightarrow E$

Give a 3NF decomposition of R.

3NF Decomposition

$AB \rightarrow CDE$
$AC \rightarrow BDE$
$B \rightarrow C$
$C \rightarrow B$
$C \rightarrow D$
$B \rightarrow E$

$\equiv$

$B \rightarrow C$
$C \rightarrow B$
$C \rightarrow D$
$B \rightarrow E$

C part of candidate key
B part of candidate key
??? NOT in 3NF

Candidate keys: AB, AC

3NF Decomposition (Synthesis)

1. Find a minimal cover of the FDs

   { $B \rightarrow C$, $C \rightarrow B$, $C \rightarrow D$, $B \rightarrow E$ }

2. Create a relation for each FD

   $R_1(B, C)$ $R_2(C, D)$ $R_3(B, E)$

3. If no relation contains a key, create one

   $R_1(B, C)$ $R_3(B, E)$

   $R_2(C, D)$ $R_4(A, B)$

Candidate keys

AB, AC

3NF Decomposition (Synthesis): Better

1. Find an extended minimal cover of the FDs

   Minimal cover $\{B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}$

   Extended minimal cover is $\{B \rightarrow CE, C \rightarrow BD\}$

2. Create a relation for each FD

   $R_1(B,C,E)$ $R_2(C,B,D)$

3. If no relation contains a key, create one

   $R_1(B,C,E)$ $R_2(C,B,D)$

   $R_3(A, B)$

Candidate keys

AB, AC