CSCI4333 Database Design & Implement

Lecture Twenty-four: Normalization

Instructor: Dr. Yifeng Gao

Reasoning About FDs

- Given some FDs, we can usually infer additional FDs:
  - $ssn \rightarrow did,\ did \rightarrow lot$ implies $ssn \rightarrow lot$
  - $A \rightarrow BC$ implies $A \rightarrow B$
- An FD f is <u>*logically implied by*</u> a set of FDs F if f holds whenever all FDs in F hold.
  - F^+ = *closure of F* is the set of all FDs that are implied by F.

Reasoning about FDs

- How do we get all the FDs that are logically implied by a given set of FDs?
- Armstrong's Axioms (X, Y, Z are sets of attributes):
  - *Reflexivity*:
    - If $X \supseteq Y$, then $X \rightarrow Y$
  - *Augmentation*:
    - If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z
  - *Transitivity*:
    - If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

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

Armstrong's axioms

- Armstrong' s axioms are *sound* and *complete* inference rules for FDs!
  - Sound: all the derived FDs (by using the axioms) are those logically implied by the given set
  - Complete: all the logically implied (by the given set) FDs can be derived by using the axioms.

Example of using Armstrong's Axioms

- Couple of additional rules (that follow from AA):
  - *Union*: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
  - *Decomposition*: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
- Derive the above two by using Armstrong's axioms!

Derive Union

- Show that

$$\text{If } X \rightarrow Y \ \text{ and } \ X \rightarrow Z, \ \text{ then } \ X \rightarrow YZ$$

– *Reflexivity*:
  - If $X \supseteq Y$, then $X \rightarrow Y$
– *Augmentation*:
  - If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z
– *Transitivity*:
  - If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

Derive Union

- Show that

$$\text{If } X \rightarrow Y \text{ and } X \rightarrow Z, \text{ then } X \rightarrow YZ$$

- $X \rightarrow YX$ (augment) // Append X on both sides of X->Y
- $YX \rightarrow YZ$ (augment) // Append Y on both sides of X->Z
- Thus $X \rightarrow YZ$ (transitive)

Derivation Example

- $R = (A, B, C, G, H, I)$
  $F = \{A \rightarrow B;\ A \rightarrow C;\ CG \rightarrow H;\ CG \rightarrow I;\ B \rightarrow H\}$
- some members of F^+ ~~(how to derive them?)~~
  - $A \rightarrow H$
    - by transitivity from $A \rightarrow B$ *and* $B \rightarrow H$
  - $AG \rightarrow I$
    - by augmenting $A \rightarrow C$ with G, to get $AG \rightarrow CG$
  - $CG \rightarrow HI$
    - by augmenting $CG \rightarrow I$ to infer $CG \rightarrow CGI$,
    - and augmenting of $CG \rightarrow H$ to infer $CGI \rightarrow HI$,
    - and then transitivity

Procedure for Computing F^+

- $F^+ = $ *closure of F* is the set of all FDs that are implied by *F*.
- To compute the closure of a set of functional dependencies F:

$F^+ = F$
repeat
    for each functional dependency f in F^+
        apply reflexivity and augmentation rules on f
        add the resulting functional dependencies to F^+
    for each pair of functional dependencies f_1 and f_2 in F^+
        if f_1 and f_2 can be combined using transitivity
            then add the resulting functional dependency to F^+
until F^+ does not change any further

NOTE: We shall see an alternative procedure for this task later

Example on Computing F+

- $F = \{A \rightarrow B,\ B \rightarrow C,\ C D \rightarrow E\}$
- Step 1: For each f in F, apply reflexivity rule
  - We get: $CD \rightarrow C$; $CD \rightarrow D$
  - Add them to F:
    - $F = \{A \rightarrow B,\ B \rightarrow C,\ C D \rightarrow E; CD \rightarrow C; CD \rightarrow D\}$
- Step 2: For each f in F, apply augmentation rule
  - From $A \rightarrow B$ we get: $A \rightarrow AB$; $AC \rightarrow BC$; $AD \rightarrow BD$; $ABC \rightarrow BC$; $ABD \rightarrow BD$; $ACD \rightarrow BCD$
  - From $B \rightarrow C$ we get: ….
- Step 3: Apply transitivity on pairs of f's
- Keep repeating… You get the idea

Attribute Closure

- A short cut: Attribute Closure
  - Compute *attribute closure* of X (denoted X^+) wrt F

  Intuition of Attribute Closure:

  Given an attribute set X, $X^+ = Z$ if all the FDs implied by X: $\{X \rightarrow Y\}$ satisfies the condition: $Y \subseteq Z$

- An efficient check for $X \rightarrow Y$:
  - Given an attribute closure X^+, if $Y \subseteq X^+$, then **FD is hold**

- Does $F = \{A \rightarrow B,\ B \rightarrow C,\ C\,D \rightarrow E\}$ imply $A \rightarrow E$?
  - i.e, is $A \rightarrow E$ in the closure F^+? Equivalently, is E in A^+?

Computing X^+

- Input F (a set of FDs), and X (a set of attributes)
- Output: Result=X^+ (under F)
- Method:
  - Step 1: Result :=X;
  - Step 2: Take $Y \rightarrow Z$ in F, **and** Y is in Result, do:

    Result := Result $\cup$ Z
  - Repeat step 2 until Result cannot be changed and then output Result.
- Easy! Hard to make mistake!

Example of Attribute Closure X^+

- Does $F = \{A \rightarrow B,\ B \rightarrow C,\ C\,D \rightarrow E\ \}$ imply $A \rightarrow E$?
  - i.e, is $A \rightarrow E$ in the closure F^+? Equivalently, is E in A^+?

Step 1: Result = A

Step 2: Consider $A \rightarrow B$, Result = AB

Consider $B \rightarrow C$, Result = ABC

Consider $CD \rightarrow E$, CD is not in ABC, so stop

Ans: $A^+ = \{ABC\}$

E is NOT in A^+, so $A \rightarrow E$ is NOT in F^+

Example of computing X^+

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

What is A^+ ? (i.e. what is the attribute closure for A?)

Answer: $A^+ = ABCD$

Computing F^+

- Given $F=\{ A \rightarrow B, B \rightarrow C\}$. Compute F^+ (with attributes A, B, C).

Step 2: Compute the attribute closures for all attribute/combination of attributes

Step 1: Construct an empty matrix, with all Possible combinations of attributes in the rows And columns

	A	B	C	AB	AC	BC	ABC
A							
B							
C							
AB							
AC							
BC							
ABC							

Attribute closure
$A^+=?$
$B^+=?$
$C^+=?$
$AB^+=?$
$AC^+=?$
$BC^+=?$
$ABC^+=?$

Step 3: Fill in the matrix using the results from Step 2

Computing F^+

- Given $F=\{A \rightarrow B, B \rightarrow C\}$. Compute F^+ (with attributes A, B, C).

Step 1: Construct an empty matrix, with all Possible combinations of attributes in the rows And columns

	A	B	C	AB	AC	BC	ABC
A	✓	✓	✓	✓	✓	✓	✓
B							
C							
:							

Step 2: Compute the attribute closures for all attribute/combination of attributes

Attribute closure
A^+=**ABC**
B^+=?
C^+=?
AB^+=?
AC^+=?
BC^+=?
ABC^+=?

Step 3: Fill in the matrix using the results from Step 2.

We have A^+=ABC. Now fill in the row for A. Consider the first column. Is A part of A^+? Yes, so check it. Is B part of A^+? Yes, so check it… and so on.

Computing F^+

- Given $F=\{ A \to B, B \to C\}$. Compute F^+ (with attributes A, B, C).

Step 1: Construct an empty matrix, with all Possible combinations of attributes in the rows And columns

	A	B	C	AB	AC	BC	ABC
A	√	√	√	√	√	√	√
B							
C							
:							

Step 2: Compute the attribute closures for all attribute/combination of attributes

Attribute closure
$A^+=$**ABC**
$B^+=?$
$C^+=?$
$AB^+=?$
$AC^+=?$
$BC^+=?$
$ABC^+=?$

Step 3: Fill in the matrix using the results from Step 2.
We have $A^+=ABC$. Now fill in the row for A. Consider the first column. Is A part of A^+? Yes, so check it.
Is B part of A^+? Yes, so check it… and so on.

Computing F^+

- Given $F=\{ A \rightarrow B, B \rightarrow C\}$. Compute F^+ (with attributes A, B, C).

	A	B	C	AB	AC	BC	ABC
A	√	√	√	√	√	√	√
B		√	√			√	
C			√				
AB	√	√	√	√	√	√	√
AC	√	√	√	√	√	√	√
BC		√	√			√	
ABC	√	√	√	√	√	√	√

Attribute closure
$A^+=ABC$
$B^+=BC$
$C^+=C$
$AB^+=ABC$
$AC^+=ABC$
$BC^+=BC$
$ABC^+=ABC$

- An entry with √ means FD (the row) $\rightarrow$ (the column) is in F^+.
- An entry gets √ when (the column) is in (the row)$^+$

Computing F^+

Step 4: Derive rules.

$A \to BC$

	A	B	C	AB	AC	BC	ABC
A	√	√	√	√	√	√	√
B		√	√			√	
C			√				
AB	√	√	√	√	√	√	√
AC	√	√	√	√	√	√	√
BC		√	√			√	
ABC	√	√	√	√	√	√	√

Attribute closure
$A^+=ABC$
$B^+=BC$
$C^+=C$
$AB^+=ABC$
$AC^+=ABC$
$BC^+=BC$
$ABC^+=ABC$

- An entry with √ means FD (the row) → (the column) is in F^+.
- An entry gets √ when (the column) is in (the row)$^+$

Decomposition Example

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

=

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

$\bowtie$

R	W
8	10
5	7

Decomposition Example 2

Student_ID	Name	Dcode	Cno	Grade
123-22-3666	Attishoo	INFS	501	A
231-31-5368	Guldu	CS	102	B
131-24-3650	Smethurst	INFS	614	B
434-26-3751	Guldu	INFS	614	A
434-26-3751	Guldu	INFS	612	C

$\neq$

Name	Dcode	Cno	Grade
Attishoo	INFS	501	A
Guldu	CS	102	B
Smethurst	INFS	614	B
Guldu	INFS	614	A
Guldu	INFS	612	C

$\bowtie$

Student_ID	Name
123-22-3666	Attishoo
231-31-5368	Guldu
131-24-3650	Smethurst
434-26-3751	Guldu