Functional Dependencies Tutorial

SWEN 304

Trimester 2, 2017

Lecturer: Dr Hui Ma

Engineering and Computer Science

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Slides by: Pavle Mogin & Hui Ma

Outline

- Inferring FDs satisfied by *Faculty* relation
- Eliminating redundant functional dependencies
  - Closure of a set of attributes
  - Finding a minimal cover
- Relation Schema keys as a consequence of functional dependencies
  - A Key Finding Algorithm
  - Inferring additional keys

Universal Relation "Faculty"

StId	*StName*	*NoPts*	*CourId*	*CoName*	*Grd*	*LecId*	*LeName*
007	James	80	M114	Math	A+	*777*	Mark
131	Susan	18	C102	Java	B-	*101*	Ewan
007	James	80	C102	Java	A	*101*	Ewan
555	Susan	18	M114	Math	B+	*999*	Vladimir
007	James	80	C103	Algorithm	A+	*99*	Peter
131	Susan	18	M214	Math	ω	*333*	Peter
555	Susan	18	C201	C++	ω	*222*	Robert
007	James	80	C201	C++	A+	*222*	Robert
010	John	0	C101	Inet	ω	*820*	Ray

FDs of the Faculty Relation Schema

- Suppose the rules of behavior in UoD dictate the following functional dependencies are valid

$$
\begin{aligned}
F = \{ & StId \rightarrow StName + NoPts, \\
& CourId \rightarrow CoName, \\
& LeId \rightarrow LeName, \\
& LeId \rightarrow CourId, \\
& StId + CourId \rightarrow Grade, \\
& StId + CourId \rightarrow LeId \}
\end{aligned}
$$

- From the relation, one can infer that the following FDs are not satisfied

$$
\left.
\begin{aligned}
& StName \rightarrow StId, \\
& CourId \rightarrow LeId, \\
& LeId \rightarrow StId, \\
& StId \rightarrow Grade, \ldots
\end{aligned}
\right\} \notin F
$$

Redundant Functional Dependencies

$U = \{A, B, C, D\}$

- $F = \{A \to B, B \to C, C \to D, AB \to D, A \to C, A \to D, BC \to D\}$
- $F_1 = \{A \to B, B \to C, C \to D\}$

- Use inference rules to show that F can be replaced by F_1
  - This way works fine for small sets of FDs (and F_1 is known)
  - The other way would be to compute closures (and F_1 is known)
  - The best way is to look directly for the minimal cover of F

Inference Rules

- Given U, F, and $X, Y, Z, W \subseteq U$

1. (Reflexivity) $Y \subseteq X \vDash X \rightarrow Y$ (trivial FD)
2. (Augmentation) $X \rightarrow Y \wedge W \subseteq Z \vDash XZ \rightarrow YW$ (partial FD)
3. (Transitivity) $X \rightarrow Y \wedge Y \rightarrow Z \vDash X \rightarrow Z$ (transitive FD)
4. (Decomposition) $X \rightarrow YZ \vDash X \rightarrow Y \wedge X \rightarrow Z$
5. (Union) $X \rightarrow Y \wedge X \rightarrow Z \vDash X \rightarrow YZ$
6. (Pseudo transitivity) $X \rightarrow Y \wedge WY \rightarrow Z \vDash WX \rightarrow Z$

(if $W = \emptyset$, pseudo transitivity turns into transitivity)

- Inference rules 1, 2 and 3 are known as **Armstrong's inference rules**

Computing the Closure of F

- $F_1 = \{A \to B,\ B \to C,\ C \to D\}$
- $F_1{}^+ = \{\emptyset \to \emptyset,\ A \to \emptyset,\ A \to A,\ A \to B,\ A \to C,\ A \to D,\ B \to \emptyset,\ B \to B,\ B \to C,\ B \to D,\ C \to \emptyset,\ C \to C,\ C \to D,\ D \to \emptyset,\ D \to D,\ AB \to \emptyset,\ AB \to A,\ AB \to B,\ AB \to C,\ AB \to D,\ AC \to \emptyset,\ AC \to A,\ AC \to B,\ AC \to C,\ AC \to D,\ AD \to \emptyset,\ AD \to A,\ AD \to B,\ AD \to C,\ AD \to D,\ BC \to \emptyset,\ BC \to B,\ BC \to C,\ BC \to D,\ BD \to \emptyset,\ BD \to D,\ BD \to C,\ BD \to D,\ CD \to \emptyset,\ CD \to C,\ CD \to D,\ ABC \to \emptyset,\ ABC \to A,\ ABC \to B,\ ABC \to C,\ ABC \to C,\ ABC \to D,\ ABD \to \emptyset,\ ABD \to A,\ ABD \to B,\ ABD \to C,\ ABC \to D,\ BCD \to \emptyset,\ BCD \to B,\ BCD \to C,\ BCD \to D,\ ABCD \to \emptyset,\ ABCD \to A,\ ABCD \to B,\ ABCD \to C,\ ABCD \to D\}$

Closure of a Set of Attributes

- Given U, F and $X \subseteq U$
- Closure of X with regard to F, defined as

$$X_F^+ = \{A \in U \mid X \to A \in F^+\}$$

is used in finding the minimal cover of F

Attribute Closure Algorithm

```
X+ := X;                        // according to reflexivity
oldX+ = ∅
while (oldX+ ⊂ X+) {
        oldX+ = X+
        for (each FD Y → Z ∈ F) {
                if (Y ⊆ X+) {
                X+ = X+ ∪ Z;  //according to
                  // augmentation & transitivity
                }
        }
}
```

Exercise 1: Computing the Closure of X

- $F = \{D \rightarrow E, BC \rightarrow D, A \rightarrow C, A \rightarrow D\}$, compute closure of
  - $A^+ =$
  - $B^+ =$
  - $C^+ =$
  - $D^+ =$
  - $E^+ =$
  - *AB, AC, AD, AE, BC, BD, BE, CD, CE, DE, ABC, ABD….*

Exercise 1: Computing the Closure of X

- $F = \{D \to E, BC \to D, A \to C, A \to D\}$
  - $A^+ = ACDE$
  - $B^+ = B$
  - $C^+ = C$
  - $D^+ = DE$
  - $E^+ = E$
  - $(BC)^+ = BCDE$
  - $(AB)^+ = ABCDE$

- Note: we need to compute closures for all subsets of attributes of relation *R* to determine keys for *R*. Here there are 31 subsets.

Minimal Cover of a Set of FDs F

- A set of FDs *G* is a minimal cover of a set *F* if:
  1. each FD in *G* has a single attribute on its right hand side
  2. *G* is left reduced (no one FD in *G* has any superfluous attribute on its left hand side, (a left reduced FD = total FD, a not reduced FD = partial FD))

     $$(\forall X \to A \in G)(\forall B \in X)((X - B) \to A \notin G^+)$$

  3. *G* is non redundant (doesn't contain any trivial or pseudo transitive FD)

     $$(\forall X \to A \in G)((G - \{X \to A\})^+ \subset G^+),$$

  4. $F^+ = G^+$

Finding a Minimal Cover Algorithm

1. Set $G := F$
2. Replace each FD $X \rightarrow \{A_1, A_2, \ldots, A_n\}$ in G with the following n FDs $X \rightarrow A_1$, $X \rightarrow A_2, \ldots, X \rightarrow A_n$
3. Do left reduction

   for each FD $X \rightarrow A$ in G do

   for each B in X do

   if $A \in (X - B)^+_G$ then

   $G := (G - \{X \rightarrow A\}) \cup \{(X - B) \rightarrow A\}$

4. Eliminate redundant FDs

   for each FD $X \rightarrow A$ in G do

   if $A \in (X)^+_{G - \{X \rightarrow A\}}$ then $G := G - \{X \rightarrow A\}$

Computing a Minimal Cover Example 1

- $U = \{A, B, C, D, E\}$
- $F = \{A \to B, AC \to B, A \to A, AD \to CE, B \to DE\}$
- After step 2 of the algorithm

  $G = \{A \to B, AC \to B, A \to A, AD \to C, AD \to E, B \to D, B \to E\}$
- After step 3 of the algorithm

  $G = \{A \to B, A \to A, A \to C, A \to E, B \to D, B \to E\}$
- After step 4 of the algorithm

  $G = \{A \to B, A \to C, B \to D, B \to E\}$

Exercise 2: Computing a Minimal Cover

- Given:

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

$$F = \{A \to B,\ A \to C,\ B \to C,\ C \to B\}$$

- Compute the two possible minimal covers of F
- In the fourth step of the minimal cover algorithm:
  - first consider whether FD $A \to B$ is redundant
  - then consider whether FD $A \to C$ is redundant
- In the second attempt
  - consider FD $A \to C$ first, and
  - then FD $A \to B$

A Key Finding Algorithm

$X := R$ (* X is initialized as a super key*)

for each A in X do

if $R \subseteq (X - A)^{+}_{F}$ then

$X := X - A$

- Example
  - $R = \{A, B, C\}$, $F = \{A \rightarrow B, B \rightarrow C\}$
  - $X = ABC$
  - $(X - A)^{+}_{F} = BC$ (* The superkey is still $X = ABC$ *)
  - $(X - B)^{+}_{F} = ABC$ (* The superkey is now $X = AC$ *)
  - $(X - C)^{+}_{F} = ABC$ (* The superkey is now $X = A$ *)
  - $K(R) = A$

Exercise 3: Finding Keys

- $R = \{A, B, C\}$, $F = \{A \to B, B \to C, C \to A\}$
- $R = \{D, E, F\}$, $F = \{D \to E, E \to D, D \to F\}$
- $R = \{G, H, I\}$, $F = \{G \to H, G \to I\}$
- $R = \{C, E, J\}$, $F = \{CE \to J\}$
- $R = \{C, E, G\}$, $F = \{\}$
- $R = \{I, L\}$, $F = \{I \to L\}$

Inferring Additional Keys

- Let $X = \{A_1, \ldots, A_i, \ldots, A_k\}$ be a relation schema (R, F) key, where $X \subseteq R$,
  - If there is $W \rightarrow Z \in F$ $(Z \not\subseteq W$, $Z \subseteq X$ and $W \not\subseteq X)$
  - Then $Y = (X - Z) \cup W$ is also a relation schema (R, F) key,
- Example:

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

  $X = AB$ is a key of (R, F)
  - since $D \rightarrow B \in F$

    $Y = AD$ is another key of (R, F)
  - since $C \rightarrow D \in F$

    $Z = AC$ is a key of (R, F), as well

Exercise 4: Finding Keys

$$R = \{StdId,\ StName,\ NoPts,\ CourId,\ CoName,$$
$$LecId,\ LeName,\ Grade\ \}$$

$$F = \{StdId \rightarrow StName + NoPts,$$
$$CourId \rightarrow CoName,$$
$$LecId \rightarrow LeName,$$
$$LecId \rightarrow CourId,$$
$$StdId + CourId \rightarrow Grade,$$
$$StdId + CourId \rightarrow LecId\ \}$$

- $K_1(Faculty) = StdId + CourId$
- $K_2(Faculty) = StdId + LeId$