FUNCTIONAL DEPENDENCY THEORY

Relational Database Design _Part 3a

Outline

Features of Good Relational Design

Atomic Domains and First Normal Form

Functional Dependencies

Normal Forms

Functional Dependency Theory

Decomposition Using Functional Dependencies

Algorithms for Decomposition using Functional Dependencies

Decomposition Using Multivalued Dependencies

More Normal Form

Functional-Dependency Theory

We now consider the formal theory that tells us which functional dependencies are implied logically by a given set of functional dependencies.

Closure of a Set of Functional Dependencies

Given a set *F* set of functional dependencies, there are certain other functional dependencies that are logically implied by *F*.

- For e.g.: If $A \rightarrow B$ and $B \rightarrow C$, then we can infer that $A \rightarrow \mathrm{C}$

The set of **all** functional dependencies logically implied by *F* is the **closure** of *F*.

We denote the *closure* of *F* by $\boldsymbol{F^+}$.

Closure of a Set of Functional Dependencies

We can find F^+, the closure of F, by repeatedly applying **Armstrong's Axioms:**

- if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ **(reflexivity)**
- if $\alpha \rightarrow \beta$, then $\gamma\,\alpha \rightarrow \gamma\,\beta$ **(augmentation)**
- if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$ **(transitivity)**

These rules are

- **sound** (generate only functional dependencies that actually hold), and
- **complete** (generate all functional dependencies that hold).

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^+

- $\boldsymbol{A \rightarrow H}$
  - by transitivity from $A \rightarrow B$ *and* $B \rightarrow H$
- $\boldsymbol{AG \rightarrow I}$
  - by augmenting $A \rightarrow C$ with G, to get $AG \rightarrow CG$ and then transitivity with $CG \rightarrow I$
- $\boldsymbol{CG \rightarrow HI}$
  - 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+

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

Closure of Functional Dependencies (Cont.)

Additional rules:

- If $\alpha \rightarrow \beta$ holds *and* $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds **(union)**
- If $\alpha \rightarrow \beta\gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds **(decomposition)**
- If $\alpha \rightarrow \beta$ holds *and* $\gamma\beta \rightarrow \delta$ holds, then $\alpha\gamma \rightarrow \delta$ holds **(pseudotransitivity)**

The above rules can be inferred from Armstrong's axioms.

Closure of Attribute Sets

Given a set of attributes α, define the ***closure*** of α **under** *F* (denoted by α^+) as the set of attributes that are functionally determined by α under *F*

Algorithm to compute α^+, the closure of α under *F*

result := α;
while (changes to *result*) **do**
  for each $\beta \rightarrow \gamma$ **in** *F* **do**
    begin
      if $\beta \subseteq$ *result* **then** *result* := *result* $\cup\ \gamma$
    end

Example of Attribute Set Closure

$R = (A, B, C, G, H, I)$

$F = \{A \rightarrow B$
$A \rightarrow C$
$CG \rightarrow H$
$CG \rightarrow I$
$B \rightarrow H\}$

$(AG)^+$

1. $result = AG$
2. $result = ABCG$ $(A \rightarrow C$ and $A \rightarrow B)$
3. $result = ABCGH$ $(CG \rightarrow H$ and $CG \subseteq AGBC)$
4. $result = ABCGHI$ $(CG \rightarrow I$ and $CG \subseteq AGBCH)$

Is *AG* a candidate key?

1. Is AG a super key?
   1. Does $AG \rightarrow R$? == Is $(AG)^+ \supseteq R$
2. Is any subset of AG a superkey?
   1. Does $A \rightarrow R$? == Is $(A)^+ \supseteq R$
   2. Does $G \rightarrow R$? == Is $(G)^+ \supseteq R$

Uses of Attribute Closure

There are several uses of the attribute closure algorithm:

Testing for superkey:

- To test if α is a superkey, we compute α^{+}, and check if α^{+} contains all attributes of *R*.

Testing functional dependencies

- To check if a functional dependency $\alpha \rightarrow \beta$ holds (or, in other words, is in F^{+}), just check if $\beta \subseteq \alpha^{+}$.
- That is, we compute α^{+} by using attribute closure, and then check if it contains β.
- Is a simple and cheap test, and very useful

Computing closure of F

- For each $\gamma \subseteq R$, we find the closure γ^{+}, and for each $S \subseteq \gamma^{+}$, we output a functional dependency $\gamma \rightarrow S$.