Database Management System (DBMS)

Lecture-31

Dharmendra Kumar

October 31, 2020

Closure of attribute sets

Consider relation schema R and a set of functional dependencies F.
Let $\alpha \subseteq$ R.
The closure of α is the set of all the attributes of R which are logically determined by α under a set F. It is denoted by α^+.

The closure of α is computed by following algorithm:-

Input: α and F

Output: α^+ = result

```
result ← α
while changes to result do
    for each functional dependency β → γ in F do
        if β ⊆ result then
            result ← result ∪ γ
        end
    end
end
```

Algorithm 1: An algorithm to compute α^+, the closure of α under F

Example: Consider relation schema R = (A, B, C, G, H, I) and the set F of functional dependencies A $\rightarrow$ B, A $\rightarrow$ C, CG $\rightarrow$ H, CG $\rightarrow$ I, B $\rightarrow$ H. Compute the closure of {A,G}, {C,G} and {A}.

Solution:

$\{A, G\}^+ = \{\text{A,G}\}$
$= \{\text{A,B,C,G}\}$
$= \{\text{A,B,C,G,H,I}\}$

Therefore, $\{A, G\}^+ = \{\text{A,B,C,G,H,I}\}$

$\{C, G\}^+ = \{\text{C,G}\}$
$= \{\text{C,G,H,I}\}$

Therefore, $\{C, G\}^+ = \{\text{C,G,H,I}\}$

$\{A\}^+ = \{\text{A}\}$
$= \{\text{A,B,C}\}$
$= \{\text{A,B,C,H}\}$

Therefore, $\{A\}^+ = \{\text{A,B,C,H}\}$

Uses or applications of attribute closure

There are several uses of the attribute closure:

- To test if α is a superkey, we compute α^+, and check if α^+ contains all attributes of R.
- We can check if a functional dependency $\alpha \to \beta$ holds by checking if $\beta \subseteq \alpha^+$.
- It gives us an alternative way to compute F^+: For each $\gamma \subseteq R$, we find the closure γ^+, and for each $S \subseteq \gamma^+$, we output a functional dependency $\gamma \to S$.

Canonical Cover

Before defining canonical cover, first we are going to define some concepts related with it.

Extraneous attribute: Consider a set F of functional dependencies and the functional dependency $\alpha \rightarrow \beta$ in F.

- Attribute A is said to be extraneous in α if A $\in \alpha$, and F logically implies (F - $\{\alpha \rightarrow \beta\}$) $\cup$ $\{(\alpha - A) \rightarrow \beta\}$.
- Attribute A is said to be extraneous in β if A $\in \beta$, and the set of functional dependencies (F - $\{\alpha \rightarrow \beta\}$) $\cup$ $\{\alpha \rightarrow (\beta - A)\}$ logically implies F.

Example: Consider F = { AB $\rightarrow$ C and A $\rightarrow$ C}. Find extraneous attributes in F.

Solution: Consider the functional dependency AB $\rightarrow$ C. In this dependency, right hand side contains single attribute, therefore right hand side has no extraneous attribute.

Now, consider left hand side. Now, we check A is extraneous attribute or not.

Eliminate A from FD, AB $\rightarrow$ C. We get B $\rightarrow$ C. Clearly, B $\rightarrow$ C can not be derived from F, therefore A is not extraneous attribute.

Now, we check B is extraneous attribute or not.

Eliminate B from FD, AB $\rightarrow$ C. We get A $\rightarrow$ C. Clearly, A $\rightarrow$ C is derived from F, therefore B is an extraneous attribute.

Consider the functional dependency A $\rightarrow$ C. In this dependency, left hand and right hand side contains single attribute, therefore this FD has no extraneous attribute.

Example: Consider F = {AB $\rightarrow$ CD and A $\rightarrow$ C}. Find extraneous attributes in F.

Solution: Consider the functional dependency AB $\rightarrow$ CD. In this FD, both sides may contain extraneous attributes.

Now, consider left hand side. Now, we check A is extraneous attribute or not.

Eliminate A from FD, AB $\rightarrow$ CD. We get B $\rightarrow$ CD. Clearly, B $\rightarrow$ CD can not be derived from F, therefore A is not extraneous attribute.

Now, we check B is extraneous attribute or not.

Eliminate B from FD, AB $\rightarrow$ CD. We get A $\rightarrow$ CD. Clearly, A $\rightarrow$ CD can not be derived from F, therefore B is not extraneous attribute.

Now, we check C is extraneous attribute or not.

Eliminate C from FD, AB $\rightarrow$ CD. We get AB $\rightarrow$ D. Clearly, Set F' = {AB $\rightarrow$ D, A $\rightarrow$ C} derives set F, therefore C is an extraneous attribute.

Now, we check D is extraneous attribute or not.
Eliminate D from FD, $AB \rightarrow CD$. We get $AB \rightarrow C$. Clearly, Set F' $= \{AB \rightarrow C, A \rightarrow C\}$ can not derive set F, therefore D is not an extraneous attribute.
Consider the functional dependency $A \rightarrow C$. In this dependency, left hand and right hand side contains single attribute, therefore this FD has no extraneous attribute.

Redundant functional dependency

A functional dependency $\alpha \rightarrow \beta$ in F is said to be redundant if after eliminating $\alpha \rightarrow \beta$ from F, we get a set of functional dependency F' equivalent to F. That is, $F^+ = F'^+$.

Example: Consider F = {A → B, B → C, and A → C}. In this set F, FD A → C} is redundant because it is derived from A → B and B → C using transitivity rule.

Canonical Cover:

Canonical cover is defined for a set F of functional dependencies. Canonical cover of F is the minimal set of functional dependencies equivalent to F that is canonical cover is a set of functional dependencies equivalent to F which does not contain any extraneous attribute and redundant FD. It is denoted by F_c.

A canonical cover for a set of functional dependencies F can be computed by following algorithm.

Input: F

Output: F_c

$F_c \leftarrow F$

repeat

Use the union rule to replace any dependencies in F_c of the form $\alpha_1 \rightarrow \beta_1$ and $\alpha_1 \rightarrow \beta_2$ with $\alpha_1 \rightarrow \beta_1\beta_2$.

Find a functional dependency $\alpha \rightarrow \beta$ in F_c with an extraneous attribute either in α or in β.

/* Note: the test for extraneous attributes is done using F_c, not F */

If an extraneous attribute is found, delete it from $\alpha \rightarrow \beta$.

until F_c *does not change any further*;

Algorithm 2: Computing canonical cover

Example: Consider the following set F of functional dependencies on relation schema R = (A,B,C):

$A \to BC$, $B \to C$, $A \to B$, $AB \to C$

Compute the canonical cover for F.

Solution:

- There are two functional dependencies with the same set of attributes on the left side of the arrow:
  $A \to BC$, $A \to B$
  We combine these functional dependencies using union rule into $A \to BC$.
- A is extraneous in $AB \to C$ because F logically implies $(F - \{AB \to C\}) \cup \{B \to C\}$. This assertion is true because $B \to C$ is already in our set of functional dependencies.
- C is extraneous in $A \to BC$, since $A \to BC$ is logically implied by $A \to B$ and $B \to C$.

Thus, canonical cover of F is

$F_c = \{A \to B, B \to C\}$.

Note: A canonical cover might not be unique.

Relational Database Design

Example: Consider the following set F of functional dependencies on relation schema R = (A,B,C):

A $\rightarrow$ BC, B $\rightarrow$ AC, C $\rightarrow$ AB

Compute the canonical cover for F.

Solution: If we apply the extraneity test to A BC, we find that both B and C are extraneous under F. However, it is incorrect to delete both. The algorithm for finding the canonical cover picks one of the two, and deletes it. Then,

1. If C is deleted, we get the set F' = {A $\rightarrow$ B, B $\rightarrow$ AC, and C $\rightarrow$ AB}. Now, B is not extraneous in the right hand side of A $\rightarrow$ B under F'. Continuing the algorithm, we find A and B are extraneous in the right hand side of C $\rightarrow$ AB, leading to two canonical covers
   F_c = {A $\rightarrow$ B, B $\rightarrow$ C, and C $\rightarrow$ A}, and
   F_c = {A $\rightarrow$ B, B $\rightarrow$ AC, and C $\rightarrow$ B}.
2. If B is deleted, we get the set F' = {A $\rightarrow$ C, B $\rightarrow$ AC, and C $\rightarrow$ AB}. This case is symmetrical to the previous case, leading to the canonical covers
   F_c = {A $\rightarrow$ C, C $\rightarrow$ B, and B $\rightarrow$ A}, and
   F_c = {A $\rightarrow$ C, B $\rightarrow$ C, and C $\rightarrow$ AB}.