Database Management System (DBMS)

Lecture-30

Dharmendra Kumar

October 31, 2020

Unit-3

Relational Database Design

Functional dependency

Consider a relation schema R, and let $\alpha \subseteq R$ and $\beta \subseteq R$. The functional dependency $\alpha \rightarrow \beta$ holds on relation schema R if, in any legal relation r(R), for all pairs of tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, then $t_1[\beta] = t_2[\beta]$ must also satisfy with in r(R).

Super key: A subset α of a relation schema R is said to be super key of R if $\alpha \rightarrow R$ holds.

Candidate key: A subset α of a relation schema R is said to be super key of R if

(1) α should be super key of R i.e. $\alpha \rightarrow R$.

(2) There should not exist any proper subset K of α such that K $\rightarrow$ R.

Example: Consider the following relation :-

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_2	c_2	d_3
a_3	b_3	c_2	d_4

Find out which functional dependencies are satisfied.

Solution: Observe that A$\rightarrow$C is satisfied. There are two tuples that have an A value of a_1. These tuples have the same C value namely, c_1. Similarly, the two tuples with an A value of a_2 have the same C value, c_2. There are no other pairs of distinct tuples that have the same A value. The functional dependency C $\rightarrow$ A is not satisfied, however. To see that it is not, consider the tuples $t_1 = (a_2, b_3, c_2, d_3)$ and $t_2 = (a_3, b_3, c_2, d_4)$. These two tuples have the same C values, c_2, but they have different A values, a_2 and a_3, respectively. Thus, we have found a pair of tuples t_1 and t_2 such that $t_1[C] = t_2[C]$, but $t_1[A] \neq t_2[A]$.
Some other functional dependencies which satisfied are the following:-

AB $\rightarrow$ C, D $\rightarrow$ B, BC $\rightarrow$ A, CD $\rightarrow$ A, CD $\rightarrow$ B, AD $\rightarrow$ B, AD $\rightarrow$ C.

Trivial functional dependency

A functional dependency $\alpha \rightarrow \beta$ is said to be trivial if $\beta \subseteq \alpha$.
Some trivial functional dependencies are the following:- ABC $\rightarrow$ C, CD $\rightarrow$ C, A $\rightarrow$ A.

Closure of a Set of Functional Dependencies

Consider F is a set of functional dependencies defined on relation schema R.
Closure of F is the set of all the functional dependencies which are logically implied(or derived) from F. It is denoted by F^+.

Armstrong's axioms

Following three rules are said to be Armstrong's axioms.

- **Reflexivity rule:** If α is a set of attributes and $\beta \subseteq \alpha$, then $\alpha \to \beta$ holds.
- **Augmentation rule:** If $\alpha \to \beta$ holds and γ is a set of attributes, then $\gamma\alpha \to \gamma\beta$ holds.
- **Transitivity rule:** If $\alpha \to \beta$ holds and $\beta \to \gamma$ holds, then $\alpha \to \gamma$ holds.

Some additional rules are the following:-

- **Union rule:** If $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds.
- **Decomposition rule:** If $\alpha \rightarrow \beta\gamma$ holds then $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$ holds.
- **Pseudo transitivity rule:** If $\alpha \rightarrow \beta$ holds and $\gamma\beta \rightarrow \delta$ holds, then $\gamma\alpha \rightarrow \delta$ holds.

Example: Consider relation schema R = (A, B, C, G, H, I) and the set F of functional dependencies $A \rightarrow B$, $A \rightarrow C$, $CG \rightarrow H$, $CG \rightarrow I$, $B \rightarrow H$. We list several members of F+ here:

- $A \rightarrow H$. Since $A \rightarrow B$ and $B \rightarrow H$ hold, we apply the transitivity rule.
- $CG \rightarrow HI$. Since $CG \rightarrow H$ and $CG \rightarrow I$, the union rule implies that $CG \rightarrow HI$.
- $AG \rightarrow I$. Since $A \rightarrow C$ and $CG \rightarrow I$, the pseudo transitivity rule implies that $AG \rightarrow I$ holds.

Note: The left-hand and right-hand sides of a functional dependency are both subsets of R. Since a set of size n has 2^n subsets, therefore there are a total of $2 \times 2^n = 2^{n+1}$ possible functional dependencies, where n is the number of attributes in R.

Relational Database Design

Algorithm to compute F^+ using Armstrong's axioms

In this algorithm, the input will be F and R. It is computed by following algorithm:-

Input: F and R

Output: F^+

$F^+ \leftarrow F$

repeat

  for *for each functional dependency f in* F^+ **do**

    apply reflexivity and augmentation rules on f

    add the resulting functional dependencies to F^+

  end

  for *each pair of functional dependencies* f_1 *and* f_2 *in* F^+ **do**

    if f_1 *and* f_2 *can be combined using transitivity rule* **then**

      add the resulting functional dependency to F^+

    end

  end

until F^+ *does not change any further*;

Algorithm 1: A procedure to compute F+