Database Management System (DBMS)

Lecture-35

Dharmendra Kumar

November 11, 2020

Relational Database Design

Algorithm to check if a decomposition is lossless

Input: A relation schema R($A_1, A_2, ..., A_n$) and a decomposition D = $\{R_1, R_2, ..., R_m\}$ and a set F of functional dependencies.

1. Create a matrix S with one row i for each relation R_i in D, and one column j for each attribute a_j in R.
2. Set S(i,j) = a_j , if relation schema R_i contains attribute A_j otherwise set S(i,j) = b_{ij}.

Relational Database Design

3.

change $\leftarrow$ *true*

while *(change* $=$ *true)* **do**

  for *each functional dependency* $\alpha \rightarrow \beta$ *in* F **do**

    if *(row i and j exists such that the same symbol appears in each column corresponding to attribute in* α*)* **then**

      if *one of the symbols in the* β *column is* a_r **then**

        Make other symbol to be a_r

      end

      else if *the symbols are* b_{pk} *and* b_{qk} **then**

        Make both of them b_{pk}

      end

    end

    else

      change $=$ false

    end

  end

end

4. If there exists a row in which all the symbols are a's, then the decomposition has the lossless. Otherwise decomposition has lossy.

Example:

Consider the following schema:-

R=(SSN, Ename, Pnumber, Pname, Plocation, Hours)

F= { SSN $\rightarrow$ Ename, Pnumber $\rightarrow$ Pname Plocation, SSN pnumber $\rightarrow$ Hours }

R_1 = (SSN, Ename)

R_2 = (Pnumber, Pname, Plocation)

R_3 = (SSN, Pnumber, Hours)

Find out decomposition of R in R_1, R_2, R_3 is lossless or lossy.

Solution:

First construct matrix. It will be order of 3×6.

The initialization table will be the following:-

	SSN	Ename	Pnumber	Pname	Plocation	Hours
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	b_{32}	a_3	b_{34}	b_{35}	a_6

After the first iteration , the table will be the following:-

	SSN	Ename	Pnumber	Pname	Plocation	Hours
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	a_2	a_3	a_4	a_5	a_6

After the second iteration, the table will not changed. Therefore, the above table is the final table.

Since in this table, row 3 contains only a's symbol, therefore this decomposition is lossless.

Example:

Consider the following schema:-

R=(A, B, C, D, E)

F= { AB $\rightarrow$ CD, A $\rightarrow$ E, C $\rightarrow$ D }

R_1 = (A, B, C)

R_2 = (B, C, D)

R_3 = (C, D, E)

Is the decomposition of R in R_1, R_2, R_3 is lossless or lossy?

Solution:

First construct matrix. It will be order of 3×5.

The initialization table will be the following:-

	A	B	C	D	E
R_1	a_1	a_2	a_3	b_{14}	b_{15}
R_2	b_{21}	a_2	a_3	a_4	b_{25}
R_3	b_{31}	b_{32}	a_3	a_4	a_5

After the first iteration , the table will be the following:-

	A	B	C	D	E
R_1	a_1	a_2	a_3	a_4	b_{15}
R_2	b_{21}	a_2	a_3	a_4	b_{25}
R_3	b_{31}	b_{32}	a_3	a_4	a_5

After the second iteration, the table will not changed. Therefore, the above table is the final table.

Since in this table, no row contains only a's symbol, therefore this decomposition is lossy.

Multivalued dependency

Let R be a relation schema and let $\alpha \subseteq R$ and $\beta \subseteq R$. The multivalued dependency $\alpha \rightarrow\rightarrow \beta$ holds on R if in any legal relation r(R), for all pairs of t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, there exists two tuples t_3 and t_4 in r such that

$$t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$
$$t_3[\beta] = t_1[\beta]$$
$$t_4[\beta] = t_1[\beta]$$
$$t_3[R - \beta] = t_2[R - \beta]$$
$$t_4[R - \beta] = t_1[R - \beta]$$

Trivial multivalued dependency

A multivalued dependency $\alpha \rightarrow\rightarrow \beta$ is said to be trivial if $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$.

Note: If $\alpha \rightarrow \beta$, then $\alpha \rightarrow\rightarrow \beta$. That is, every functional dependency is also a multivalued dependency.

Example: Consider the following relation schema.

loan-number	customer-name	customer-street	customer-city
L-23	Smith	North	Rye
L-23	Smith	Main	Manchester
L-93	Curry	Lake	Horseneck

In this table, multivalued dependency

customer-name $\rightarrow\!\!\rightarrow$ customer-street customer-city

holds.