Database Management Systems

Module 29: Relational Database Design/9: MVD and 4NF

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

Module Recap

- Using the specification for a Library Information System, we have illustrated how a schema can be designed and then refined for finalization

Module Objectives

- To understand multi-valued dependencies arising out of attributes that can have multiple values
- To define Fourth Normal Form and learn the decomposition algorithm to 4NF

Module Outline

- Multivalued Dependencies
- Decomposition to 4NF

MVD

Multivalued Dependency

MVD: Multivalued Dependency

- Persons(Man, Phones, Dog_Like)

Person :			Meaning of the tuples
Man(M)	**Phones(P)**	**Dogs_Like(D)**	Man M have phones P, and likes the dogs D.
M1	P1/P2	D1/D2	M1 have phones P1 and P2, and likes the dogs D1 and D2.
M2	P3	D2	M2 have phones P3, and likes the dog D2.
Key : MPD			

There are no non trivial FDs because all attributes are combined forming Candidate Key, that is, MDP. In the above relation, two multivalued dependencies exists:

- Man $\twoheadrightarrow$ Phones
- Man $\twoheadrightarrow$ Dogs_Like

A man's phone are independent of the dogs they like. But after converting the above relation in Single Valued Attribute, each of a man's phones appears with each of the dogs they like in all combinations.

Post 1NF Normalization

Man(M)	Phones(P)	Dogs_Likes(D)
M1	P1	D1
M1	P2	D2
M2	P3	D2
M1	P1	D2
M1	P2	D1

Source: http://www.edugrabs.com/multivalued-dependency-mvd/

MVD (2)

- If two or more independent relations are kept in a single relation, then Multivalued Dependency is possible. For example, Let there are two relations :
  - **Student(SID, Sname)** *where* **(SID $\rightarrow$ Sname)**
  - **Course(CID, Cname)** *where* **(CID $\rightarrow$ Cname)**
- There is no relation defined between Student and Course. If we kept them in a single relation named **Student_Course**, then MVD will exists because of m:n Cardinality
- If two or more MVDs exist in a relation, then while converting into SVAs, MVD exists.

Student:

SID	Sname
S1	A
S2	B

Course:

CID	Cname
C1	C
C2	B

SID	Sname	CID	Cname
S1	A	C1	C
S1	A	C2	B
S2	B	C1	C
S2	B	C2	B

2 MVDs exist:
1. SID $\rightarrow\rightarrow$ CID
2. SID $\rightarrow\rightarrow$ Cname

Source: http://www.edugrabs.com/multivalued-dependency-mvd/

MVD (3)

- Suppose we record names of children, and phone numbers for instructors:
  - *inst_child(ID, child_name)*
  - *inst_phone(ID, phone_number)*
- If we were to combine these schema to get
  - *inst_info(ID, child_name, phone_number)*
  - Example data:
    (99999, David, 512-555-1234)
    (99999, David, 512-555-4321)
    (99999, William, 512-555-1234)
    (99999, William, 512-555-4321)
- This relation is in BCNF
  - Why?

MVD: Definition

- Let R be a relation schema and let $\alpha \subseteq$ R and $\beta \subseteq$ R. The **multivalued dependency**

  $\alpha \twoheadrightarrow \beta$

  holds on R if in any legal relation $r(R)$, for all pairs for tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2\ [\alpha]$, there exist tuples t_3 and t_4 in r such that:

$$
\begin{aligned}
&t_1[\alpha] = t_2\ [\alpha] = t_3\ [\alpha] = t_4\ [\alpha]\\
&t_3[\beta] = t_1\ [\beta]\\
&t_3[R-\beta] = t_2[R-\beta]\\
&t_4\ [\beta] = t_2[\beta]\\
&t_4[R-\beta] = t_1[R-\beta]
\end{aligned}
$$

Example: A relation of university courses, the books recommended for the course, and the lecturers who will be teaching the course:

- **course** $\twoheadrightarrow$ **book**
- **course** $\twoheadrightarrow$ **lecturer**

Test: course $\twoheadrightarrow$ book

Course	Book	Lecturer	Tuples
AHA	Silberschatz	John D	t1
AHA	Nederpelt	William M	t2
AHA	Silberschatz	William M	t3
AHA	Nederpelt	John D	t4
AHA	Silberschatz	Christian G	
AHA	Nederpelt	Christian G	
OSO	Silberschatz	John D	
OSO	Silberschatz	William M	

MVD: Example

- Let R be a relation schema with a set of attributes that are partitioned into 3 nonempty subsets.
  Y, Z, W
- We say that Y $\twoheadrightarrow$ Z (Y **multidetermines** Z) if and only if for all possible relations $r\ (R\ )$
  $< y_1, z_1, w_1 >\in r$ and $< y_1, z_2, w_2 >\in r$
  then
  $< y_1, z_1, w_2 >\in$ r and $< y_1, z_2, w_1 >\in r$
- Note that since the behavior of Z and W are identical it follows that
  $Y \twoheadrightarrow Z$ if $Y \twoheadrightarrow W$

MVD: Example (2)

- In our example:
  $ID \twoheadrightarrow child\_name$
  $ID \twoheadrightarrow phone\_number$
- The above formal definition is supposed to formalize the notion that given a particular value of $Y(ID)$ it has associated with it a set of values of Z ($child\_name$) and a set of values of W ($phone\_number$), and these two sets are in some sense independent of each other.
- Note:
  - If $Y \rightarrow Z$ then $Y \twoheadrightarrow Z$
  - Indeed we have (in above notation) $Z_1 = Z_2$
    The claim follows.

MVD: Use

- We use multivalued dependencies in two ways:
  a) To test relations to **determine** whether they are legal under a given set of functional and multivalued dependencies
  b) To specify **constraints** on the set of legal relations. We shall thus concern ourselves only with relations that satisfy a given set of functional and multivalued dependencies.
- If a relation r fails to satisfy a given multivalued dependency, we can construct a relations r' that does satisfy the multivalued dependency by adding tuples to r.

MVD: Theory

	Name	Rule
C-	Complementation	If $X \twoheadrightarrow Y$, then $X \twoheadrightarrow (R - (X \cup Y))$.
A-	Augmentation	If $X \twoheadrightarrow Y$ and $W \supseteq Z$, then $WX \twoheadrightarrow YZ$.
T-	Transitivity	If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then $X \twoheadrightarrow (Z - Y)$.
	Replication	If $X \rightarrow Y$, then $X \twoheadrightarrow Y$ but the reverse is not true.
	Coalescence	If $X \twoheadrightarrow Y$ and there is a W such that $W \cap Y$ is empty, $W \rightarrow Z$ and $Y \supseteq Z$, then $X \rightarrow Z$.

- A MVD $\mathbf{X} \twoheadrightarrow \mathbf{Y}$ in $\mathbf{R}$ is called a trivial MVD is
  - $\mathbf{Y}$ is a subset of $\mathbf{X}$ $(\mathbf{X} \supseteq \mathbf{Y})$ or
  - $\mathbf{X} \cup \mathbf{Y} = \mathbf{R}$. Otherwise, it is a non trivial MVD and we have to repeat values redundantly in the tuples.

MVD: Theory (2)

- From the definition of multivalued dependency, we can derive the following rule:
  - If $\alpha \rightarrow \beta$, then $\alpha \twoheadrightarrow \beta$

  That is, every functional dependency is also a multivalued dependency
- The **closure** D^+ of D is the set of all functional and multivalued dependencies logically implied by D.
  - We can compute D^+ from D, using the formal definitions of functional dependencies and multivalued dependencies.
  - We can manage with such reasoning for very simple multivalued dependencies, which seem to be most common in practice
  - For complex dependencies, it is better to reason about sets of dependencies using a system of inference rules

Decomposition to 4NF

Fourth Normal Form

- A relation schema R is in **4NF** with respect to a set D of functional and multivalued dependencies if for all multivalued dependencies in D^+ of the form $\alpha \twoheadrightarrow \beta$, where $\alpha \subseteq$ R and $\beta \subseteq$ R, at least one of the following hold:
  - $\alpha \twoheadrightarrow \beta$ is trivial (that is, $\beta \subseteq \alpha$ or $\alpha \cup \beta =$ R)
  - α is a superkey for schema R
- If a relation is in 4NF it is in BCNF

Restriction of Multivalued Dependencies

- The restriction of D to R_i is the set D_i consisting of
  - All functional dependencies in D^+ that include only attributes of R_i
  - All multivalued dependencies of the form
    $\alpha \twoheadrightarrow (\beta \cap R_i)$
    where $\alpha \subseteq R_i$ and $\alpha \twoheadrightarrow \beta$ is in D^+

4NF Decomposition Algorithm

a) For all dependencies A $\twoheadrightarrow$ B in D^+, check if A is a superkey
   - By using attribute closure

b) If not, then
   - Choose a dependency in F+ that breaks the 4NF rules, say A $\twoheadrightarrow$ B
   - Create R1 = A B
   - Create R2 = (R – (B – A))
   - Note that: R1 $\cap$ R2 = A and A $\twoheadrightarrow$ AB (= R1), so this is lossless decomposition

c) Repeat for R1, and R2
   - By defining $D1^+$ to be all dependencies in F that contain only attributes in R1
   - Similarly $D2^+$

4NF Decomposition Algorithm

result: $= \{R\}$;
done := false;
compute D^+;
Let D_i denote the restriction of D^+ to R_i
while (**not** *done*)
  if (there is a schema R_i in *result* that is not in 4NF) **then**
    begin
      let $\alpha \twoheadrightarrow \beta$ be a nontrivial multivalued dependency that holds
        on R_i such that $\alpha \rightarrow R_i$ is not in D_i, and $\alpha \cap \beta = \phi$;
      $result := (result - R_i) \cup (R_i - \beta) \cup (\alpha, \beta)$;
    end
  else *done*:= true;

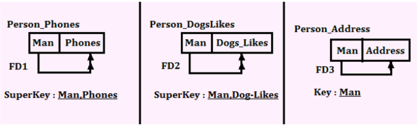
Note: each R_i is in 4NF, and decomposition is lossless-join

Example of 4NF Decomposition

- Example:
  - **Person_Modify(Man(M), Phones(P), Dog_Likes(D), Address(A))**
  - FDs:
    - FD1 : Man $\twoheadrightarrow$ Phones
    - FD2 : Man $\twoheadrightarrow$ Dogs_Like
    - FD3 : Man $\rightarrow$ Address
  - Key = MPD
  - All dependencies violate 4NF

Man(M)	Phones(P)	Dogs_Likes(D)	Address(A)
M1	P1	D1	49-ABC,Bhiwani(HR.)
M1	P2	D2	49-ABC,Bhiwani(HR.)
M2	P3	D2	36-XYZ,Rohtak(HR.)
M1	P1	D2	49-ABC,Bhiwani(HR.)
M1	P2	D1	49-ABC,Bhiwani(HR.)

Post Normalization



In the above relations for both the MVD's – ***'X' is Man***, which is again not the super key, but as ***X*** $\cup$ ***Y*** = ***R*** i.e. (Man & Phones) together make the relation.
So, the above MVD's are trivial and in FD 3, Address is functionally dependent on Man, where **Man** is the key in **Person_Address**, hence all the three relations are in 4NF.

Example of 4NF Decomposition

- R =(A, B, C, G, H, I)
  F = A $\twoheadrightarrow$ B
  B $\twoheadrightarrow$ HI
  CG $\twoheadrightarrow$ H
- R is not in 4NF since $A \twoheadrightarrow B$ and A is not a superkey for R
- Decomposition
  a) R_1 = (A, B) (R_1 is in 4NF)
  b) R_2 = (A, C, G, H, I) (R_2 is not in 4NF, decompose into R_3 and R_4)
  c) R_3 = (C, G, H) (R_3 is in 4NF)
  d) R_4 = (A, C, G, I) (R_4 is not in 4NF, decompose into R_5 and R_6)
    - $A \twoheadrightarrow B$ and $B \twoheadrightarrow$ HI $\rightarrow$ A $\twoheadrightarrow$ HI, (MVD transitivity), and
    - and hence A $\twoheadrightarrow$ I (MVD restriction to R_4)

  e) R_5= (A, I) (R_5 is in 4NF)
  f) R_6 = (A, C, G) (R_6 is in 4NF)

Module Summary

- Understood multi-valued dependencies to handle attributes that can have multiple values
- Learnt Fourth Normal Form and decomposition to 4NF

Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.
Edited and new slides are marked with “PPD”.