Other Kinds of Dependencies

CS157A
Chris Pollett
Nov. 30, 2005.

Outline

- Multivalued Dependencies and 4NF
- Join Dependencies and 5NF

Multivalued Dependencies and 4NF

- FDs are the most common kinds of dependencies considered on the data; however, there are some other dependencies which are sometimes useful.
- For example, consider

EMP(ENAME, PNAME, DNAME)

Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

An employee may work on several projects and have several dependents. Projects and dependents are independent of each other. so for each fixed project we must ensure each possible dependent of an employee occurs. This is called a multivalued dependency.

Another Way to Think About FDs AND MVDs

- Suppose we had an FD A-->BC on a table T with columns ABC.
- One was to think of this dependency is as two maps F_1: dom(A) --> dom(B) and F_2:dom(A) --> dom(C), such that every row in T can be written as (a, F_1(a), F_2(a)) for some a∈dom(A).
- One way to generalize this would be to allow our functions to map to subsets of dom(B) and dom(C) rather than to just one value. Hence, our functions would become multivalued.
- So now F_1(a) might output $\{b_1,\ldots, b_m\}$ and F_2(a) might output $\{c_1,\ldots, c_n\}$.
- To store all the information with associated with *a* in the original table, which since we are in 1NF cannot store multivalued data, we need for each possible *i* and *j*, a row (a, b_i, c_j).

Formal Definition of Multivalued Dependency

Definition: A *multivalued dependency (MVD)* X-->> Y holds on R means that if two distinct tuples t1 and t2 exist in r such that t1[X]=t2[X] then two tuples t3 and t4 should also exist in r with the following properties: (let Z denote R-(X ∪Y))

- t3[X]=t4[X]=t1[X]=t2[X]
- t3[Y]=t1[Y] and t4[Y]=t2[Y]
- t3[Z]=t2[Z] and t4[Z]=t1[Z]

- Notice if X-->>Y then also X-->>Z
- Notice any FD is an MVD.
- If Y is a subset of X then the MVD is trivial.
- MVDs also have rules of inference like Armstrong's axioms (see book).

4NF

Definition: A relation schema R is in 4NF with respect to a set of FDs and/or MVDs F if, for every nontrivial multivalued dependency X-->>Y in F^+, X is superkey for R.

Binary Test for Multivalued Lossless Joins

- The binary test for LJP we had for functional dependencies can be modified to multivalued dependencies:
  - Lossless Join Test:

    The schemas R_1 and R_2 form a lossless join decomposition of R with respect to a set F of FDs and MVDs iff:

$$(R_1 \cap R_2) \twoheadrightarrow (R_1 - R_2)$$

or

$$(R_1 \cap R_2) \twoheadrightarrow (R_2 - R_1)$$

Lossless Join Decomposition into 4NF

Input: A universal relation R and a set of MVDs F on the attributes of R

Set D:={R};

While there is a relation Q in D that is not in 4NF

{

1.choose a relation Q in D not in 4NF

2.find a MVD X-->>Y in Q that violates 4NF

3.replace Q in D by the two relations (Q-Y) and $(X \cup Y)$

}

Since 4NF is essentially BCNF rewritten for MVDs, this algorithm works for the same reason our BCNF algorithm worked.

Join Dependencies

- Our lossless join conditions for FDs and MVDs tell us when we can replace a schema R with two schema R1 and R2 and still get a lossless join.
- Sometimes there may be no way to split R into just two relation R1 and R2 and still get the LJP to hold.
- Nevertheless, one might be able to split R into $R_1,\ldots R_n$ where n>2 and get LJP to hold.
- We say a join dependency, $JD(R_1,\ldots R_n)$, holds among $R_1,\ldots R_n$, a decomposition of R, if for every legal r of R:

$$\pi_{R_1}(r) * \pi_{R_2}(r) * \cdots * \pi_{R_n}(r) = r$$

- It turns our any MVD can be written a s JD.
- A JD is called **trivial** if one of the R_i equals R.

5NF

Definition: A relation schema is in **5NF** (or **project-join normal form**) with respect to a set F of FDs, MVDs, and JD if, for every nontrivial join dependency JD(R_1, … R_n) in F^+, every R_i is a superkey of R.

The book gives an example of a table SUPPLY(SNAME, PARTNAME, PROJNAME).

for a supplier who supplies parts to a project. The data in the book could be decomposed into three tables R1(SNAME, PARTNAME), R2(SNAME, PROJNAME), and R3(PARTNAME, PROJNAME) with the LJP. However, decomposition into any two tables would not preserve this property.