Multivalued Dependencies

Fourth Normal Form

A New Form of Redundancy

- Multivalued dependencies (MVD’s) express a condition among tuples of a relation that exists when the relation is trying to represent more than one many-many relationship.
- Then certain attributes become independent of one another, and their values must appear in all combinations.

Example

Drinkers(name, addr, phones, beersLiked)
- A drinker’s phones are independent of the beers they like.
- Thus, each of a drinker’s phones appears with each of the beers they like in all combinations.
- This repetition is unlike redundancy due to FD’s, of which name->addr is the only one.

Tuples Implied by Independence

Given name, addr ->-> phones is operative if we have the tuples above the dashed line:

<table>
<thead>
<tr>
<th>name</th>
<th>addr</th>
<th>phones</th>
<th>beersLiked</th>
</tr>
</thead>
<tbody>
<tr>
<td>sue</td>
<td>a</td>
<td>p1</td>
<td>b1</td>
</tr>
<tr>
<td>sue</td>
<td>a</td>
<td>p2</td>
<td>b2</td>
</tr>
<tr>
<td>sue</td>
<td>a</td>
<td>p2</td>
<td>b1</td>
</tr>
<tr>
<td>sue</td>
<td>a</td>
<td>p1</td>
<td>b2</td>
</tr>
</tbody>
</table>

then the tuples below the dashed line must also be in the relation.

Definition of MVD

- A multivalued dependency (MVD) X ->-> Y is an assertion that if two tuples of a relation agree on all the attributes of X, then their components in the set of attributes Y may be swapped, and the result will be two tuples that are also in the relation.

Example

The name-addr-phones-beersLiked example illustrated the MVD name->->phones and the MVD name ->-> beersLiked.
**MVD Rules**

- Every FD is an MVD.
- If \( X \rightarrow Y \), then swapping \( Y \)'s between two tuples that agree on \( X \) doesn't change the tuples.
- Therefore, the "new" tuples are surely in the relation, and we know \( X \rightarrow Y \).
- **Complementation**: If \( X \rightarrow Y \) and \( Z \) is all the other attributes, then \( X \rightarrow \neg Y \).

**Example**

- Consider a drinkers relation: Drinkers(name, areaCode, phone, beersLiked, manf)
- A drinker can have several phones, with the number divided between areaCode and phone (last 7 digits).
- A drinker can like several beers, each with its own manufacturer.

**Example Data**

Here is possible data satisfying these MVD's:

<table>
<thead>
<tr>
<th>name</th>
<th>areaCode</th>
<th>phone</th>
<th>beersLiked</th>
<th>manf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sue</td>
<td>650</td>
<td>555-1111</td>
<td>Bud</td>
<td>A.B.</td>
</tr>
<tr>
<td>Sue</td>
<td>650</td>
<td>555-1111</td>
<td>WickedAle</td>
<td>Pete's</td>
</tr>
<tr>
<td>Sue</td>
<td>415</td>
<td>555-9999</td>
<td>Bud</td>
<td>A.B.</td>
</tr>
<tr>
<td>Sue</td>
<td>415</td>
<td>555-9999</td>
<td>WickedAle</td>
<td>Pete's</td>
</tr>
</tbody>
</table>

But we cannot swap area codes or phones my themselves. That is, neither name \( \rightarrow \) areaCode nor name \( \rightarrow \) phone holds for this relation.
Fourth Normal Form

• The redundancy that comes from MVD's is not removable by putting the database schema in BCNF.
• There is a stronger normal form, called 4NF, that (intuitively) treats MVD's as FD's when it comes to decomposition, but not when determining keys of the relation.

4NF Definition

• A relation $R$ is in 4NF if whenever $X -> Y$ is a nontrivial MVD, then $X$ is a superkey.
  • "Nontrivial means that:
    1. $Y$ is not a subset of $X$, and
    2. $X$ and $Y$ are not, together, all the attributes.
• Note that the definition of "superkey" still depends on FD's only.

BCNF Versus 4NF

• Remember that every FD $X \rightarrow Y$ is also an MVD, $X \rightarrow \rightarrow Y$.
• Thus, if $R$ is in 4NF, it is certainly in BCNF.
  • Because any BCNF violation is a 4NF violation.
• But $R$ could be in BCNF and not 4NF, because MVD's are "invisible" to BCNF.

Decomposition and 4NF

• If $X \rightarrow \rightarrow Y$ is a 4NF violation for relation $R$, we can decompose $R$ using the same technique as for BCNF.
  1. $XY$ is one of the decomposed relations.
  2. All but $Y - X$ is the other.

Example

Drinkers(name, addr, phones, beersLiked)

FD: name -> addr
MVD's: name ->> phones
    name ->>> beersLiked
• Key is {name, phones, beersLiked}.
• All dependencies violate 4NF.

Example, Continued

• Decompose using name -> addr:
  1. Drinkers1(name, addr)
    • In 4NF, only dependency is name -> addr.
  2. Drinkers2(name, phones, beersLiked)
    • Not in 4NF. MVD's name ->> phones and name ->>> beersLiked apply. No FD's, so all three attributes form the key.
Example: Decompose Drinkers2

Either MVD name -&gt; phones or name -&gt; beersLiked tells us to decompose to:

- Drinkers3(name, phones)
- Drinkers4(name, beersLiked)