Database Theory for Relational Databases:

Multivalued Dependencies
4th Normal Form

CENG315 INFORMATION MANAGEMENT

Relational Design by Decomposition

- "Mega relations" + properties of data
- The system decomposes those on based on the properties that are specified
- The final set of decomposed relations satisfy normal form
  - Normal forms are "good" relations
    - No anomalies
    - No lost information
- The properties themselves are defined either as
  a) Functional dependencies $\rightarrow$ Boyce-Code Normal Form (BCNF)
  b) Multivalued dependencies $\rightarrow$ Fourth Normal Form

Example

- Apply (SSN, cName, hobby)
- FD’s?
- Keys?
- BCNF?
- Good design?

Example

- Apply (SSN, cName, hobby)
- FD's? No.
- Keys?
- BCNF?
- Good design?

Example

- Apply (SSN, cName, hobby)
- FD's? No.
- Keys? All attributes of the relation.
- BCNF?
- Good design?

Example

- Apply (SSN, cName, hobby)
- FD’s? No.
- Keys? All attributes of the relation.
- BCNF? Yes.
- Good design?

Example

- Apply (SSN, cName, hobby)
- FD's? No.
- Keys? All attributes of the relation.
- BCNF? Yes.
- Good design? No.

Definition of Multivalued Dependency

- A *multivalued dependency (MVD)* on R, $X \rightarrow\rightarrow Y$, says that if two tuples of R agree on all the attributes of X, then their components in Y may be swapped, and the result will be two tuples that are also in the relation.
- i.e., for each value of X, the values of Y are independent of the values of R-X-Y.
- MVD's can cause redundancy in a relation.

Example: MVD

Customers (name, addr, phones, drinksLiked)

- A customer may have several phone numbers and may like several drinks.
- Those are independent facts about the customer.
- A customer’s phones are independent of the drinks they like.
  - name $\rightarrow\rightarrow$ phones and name $\rightarrow\rightarrow$ drinksLiked
- Thus, each of a customer’s phones appears with each of the drinks they like in all combinations.
- This repetition is unlike FD redundancy.
  - name $\rightarrow$ addr is the only FD.

Tuples Implied by name →→ phones

- If we have tuples:

name	addr	phones	drinksLiked
sue	a	p1	d1
sue	a	p2	d2

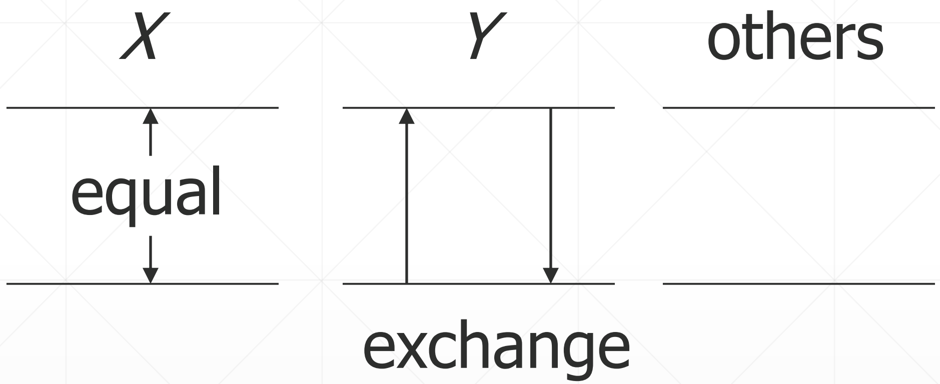
Tuples Implied by name →→ phones

- If we have tuples:

name	addr	phones	drinksLiked
sue	a	p1	d1
sue	a	p2	d2
sue	a	p2	d1
sue	a	p1	d2

- Then these tuples must also be in the relation.

Picture of MVD $X \to\to Y$



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 $X \rightarrow\rightarrow Y$.
- *Complementation*: If $X \rightarrow\rightarrow Y$, and Z is all the other attributes, then $X \rightarrow\rightarrow Z$.

Splitting Doesn't Hold

- Like FD's, we cannot generally split the left side of an MVD.
- But unlike FD's, we cannot split the right side either --- sometimes you have to leave several attributes on the right side.

Example: Multiattribute Right Sides

Customers (name, areaCode, phone, drinksLiked, manf)

- A customer can have several phones, with the number divided between areaCode and phone (last 7 digits).
- A customer can like several drinks, each with its own manufacturer.

Example: Multiattribute Right Sides (Cont.)

Customers (name, areaCode, phone, drinksLiked, manf)

- Since the areaCode-phone combinations for a customer are independent of the drinksLiked-manf combinations, we expect that the following MVD's hold:

name →→ areaCode phone

name →→ drinksLiked manf

Example Data

- Here is possible data satisfying these MVD's:

name	areaCode	phone	drinksLiked	manf
Sue	650	555-1111	Coke	C.C.
Sue	650	555-1111	IceTea	Lipton
Sue	415	55-9999	Coke	C.C.
Sue	415	55-9999	IceTea	Lipton

- But we cannot swap area codes or phones by themselves.
  - That is, neither name →→ areaCode nor name →→ phone holds for this relation.

Fourth Normal Form (4NF)

- The separation of independent facts is what 4NF is about.
- 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 $X \rightarrow\rightarrow Y$ is a nontrivial MVD, then X is a superkey.
  - *Nontrivial MVD* 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 (after conversion to an MVD).
- 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. (R-(Y-X))

4NF Decomposition Algorithm

- Input: Relation R, FD's for R, MVD's for R
- Output: Decomposition of R into 4NF relations with "lossless join"
- Compute keys for R
- Repeat until all relations are in 4NF:
  - Pick any R' with nontrivial $X \twoheadrightarrow Y$ that violates 4NF
  - Decompose R' into $R_1(X, Y)$ and $R_2(X, rest)$
  - Compute FD's and MVD's for R_1 and R_2
  - Compute keys for R_1 and R_2

Example: 4NF Decomposition

Apply (SSN, cName, hobby)

MVD's: $SSN \rightarrow\rightarrow cName$

$SSN \rightarrow\rightarrow hobby$

- Key is {SSN, cName, hobby}.
- All dependencies violate 4NF.

Example: 4NF Decomposition (Cont.)

- Decompose using SSN →→ cName:

1. Apply1 (<u>SSN</u>, <u>cName</u>)
   - No FD's
   - No MVD's
   - In 4NF
2. Apply2 (<u>SSN</u>, <u>hobby</u>)
   - No FD's
   - No MVD's
   - In 4NF

Example: 4NF Decomposition (Cont.)

Customers (name, addr, phones, drinksLiked)

FD: name → addr

MVD's: name →→ phones

name →→ drinksLiked

- Key is {name, phones, drinksLiked}.
- All dependencies violate 4NF.

Example: 4NF Decomposition (Cont.)

- Decompose using name → addr:

1. Customers1 (<u>name</u>, addr)
   - In 4NF; only dependency is name → addr.
2. Customers2 (<u>name</u>, <u>phones</u>, <u>drinksLiked</u>)
   - Not in 4NF. MVD's name →→ phones and name →→ drinksLiked apply. No FD's, so all three attributes form the key.

Example: 4NF Decomposition (Cont.)

- Either MVD name →→ phones or name →→ drinksLiked tells us to decompose to:
  - Customers3 (<u>name</u>, <u>phones</u>)
  - Customers4 (<u>name</u>, <u>drinksLiked</u>)

References

- Jeffrey D. Ullman and Jennifer Widom, “A First Course in Database Systems”, 3rd Ed., 2007.
  - http://infolab.stanford.edu/~ullman/fcdb/aut07/slides/mvds.pdf
- Stanford DB Class