

# **Chapter 3 Design Theory for Relational Databases**

# Contents

---

- Functional Dependencies
- Decompositions
- Normal Forms (BCNF, 3NF)
- Multivalued Dependencies (and 4NF)
- Reasoning About FD's + MVD's

# A New Form of Redundancy

- A relation is trying to represent more than one many-many relationship.

name	addr	phones	beersLiked
sue	a	p1	b1
sue	a	p2	b2
sue	a	p2	b1
sue	a	p1	b2

Then these tuples must also be in the relation.

- 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.

# Definition of MVD

- A *multivalued dependency* (MVD) on  $R$ ,  $X \twoheadrightarrow Y$ , 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  $R-X-Y$ .

## Example: MVD

Drinkers(name, addr, phones, beersLiked)

- A drinker many have several phones, like several beers.
  - Name $\rightarrow\rightarrow$ phones and name  $\rightarrow\rightarrow$ beersLiked.
  - name $\rightarrow$ addr is the only FD
- Each of a drinker's *phones* appears with each of the *beers* they like in all combinations.
- This repetition is *unlike* FD redundancy.

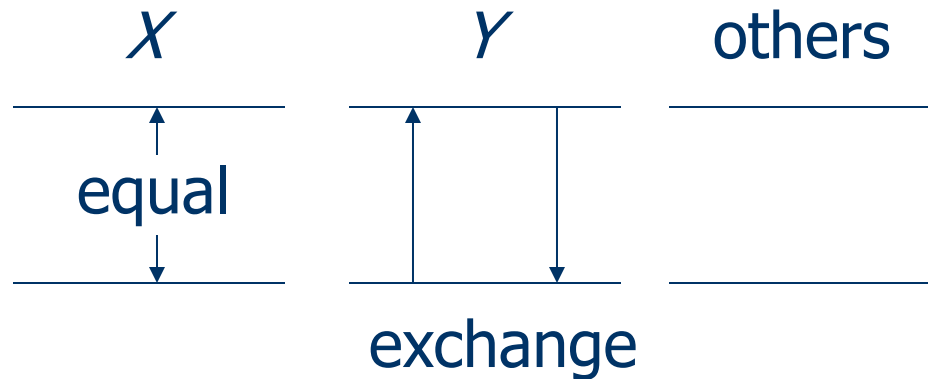
# Tuples Implied by $\text{name} \twoheadrightarrow \text{phones}$

If we have tuples:

name	addr	phones	beersLiked
sue	a	p1	b1
sue	a	p2	b2
sue	a	p2	b1
sue	a	p1	b2

Then these tuples must also be in the relation.

# Picture of MVD $X \twoheadrightarrow Y$



# MVD Rules

- Every FD is an MVD (*promotion*).
  - If  $X \twoheadrightarrow Y$ , 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  $X \twoheadrightarrow\!\!\rightarrow Y$ .
- *Complementation* : If  $X \twoheadrightarrow\!\!\rightarrow Y$ , and  $Z$  is all the other attributes, then  $X \twoheadrightarrow\!\!\rightarrow Z$ .



## MVD Rules (cont.)

- INTERSECTION RULE

$X \twoheadrightarrow Y, X \twoheadrightarrow Z$

THEN  $X \twoheadrightarrow Y \text{ INTERSECTION } Z$

- TRANSITIVE RULE

$X \twoheadrightarrow Y, Y \twoheadrightarrow Z$  THEN  $X \twoheadrightarrow Z$

# 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

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 Continued

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

name  $\twoheadrightarrow$  areaCode phone

name  $\twoheadrightarrow$  beersLiked manf

# Example Data

Here is possible data satisfying these MVD's:

name	areaCode	phone	beersLiked	manf
Sue	650	555-1111	Bud	A.B.
Sue	650	555-1111	WickedAle	Pete's
Sue	415	555-9999	Bud	A.B.
Sue	415	555-9999	WickedAle	Pete's



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

- The redundancy that comes from MVD's is not removable by putting the database schema in BCNF.

- Drinkers(name, phones, beersLiked)

*MVD's:  $Name \twoheadrightarrow \twoheadrightarrow phones, name \twoheadrightarrow \twoheadrightarrow beersLiked$*

*FD's: no:*

*Key: all attributes*

- Drinkers is in BCNF.

# 4NF Definition

- A relation  $R$  is in **4NF** if: whenever  $X \twoheadrightarrow Y$  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,  $X \twoheadrightarrow Y$ .
- 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

**Input:** relation  $R$  + FDs for  $R$  + MVDs for  $R$

**Output:** decomposition of  $R$  into 4NF relations

with “lossless join”

1. **Compute keys for  $R$**
2. Repeat until all relations are in 4NF:

**Pick any  $R'$  with nontrivial  $A \twoheadrightarrow B$  that violates 4NF**

Decompose  $R'$  into  $R_1(A, B)$  and  $R_2(A, \text{rest})$

Compute FDs and MVDs for  $R_1$  and  $R_2$

Computer keys for  $R_1$  and  $R_2$

## Example: 4NF Decomposition

Drinkers(name, addr, phones, beersLiked)

FD:            name  $\rightarrow$  addr

MVD's:        name  $\twoheadrightarrow$  phones

                name  $\twoheadrightarrow$  beersLiked

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

## Example Continued

- Decompose using  $\text{name} \rightarrow \text{addr}$ :
  1. Drinkers1(name, addr)
    - ◆ In 4NF; only dependency is  $\text{name} \rightarrow \text{addr}$ .
  2. Drinkers2(name, phones, beersLiked)
    - ◆ Not in 4NF. MVD's  $\text{name} \twoheadrightarrow \text{phones}$  and  $\text{name} \twoheadrightarrow \text{beersLiked}$  apply. No FD's, so all three attributes form the key.

# Example: Decompose Drinkers2

- Either MVD  $\text{name} \twoheadrightarrow \text{phones}$  or  $\text{name} \twoheadrightarrow \text{beersLiked}$  tells us to decompose to:
  - Drinkers3(name, phones)
  - Drinkers4(name, beersLiked)
  - Drinkers3 is in 4NF
  - Drinker4 is in 4NF

# Reasoning About MVD's + FD's

- **Problem:** given a set of MVD's and/or FD's that hold for a relation  $R$ , does a certain FD or MVD also hold in  $R$  ?  
e.g: if  $A \twoheadrightarrow BC$  and  $D \rightarrow C$ , does  $A \rightarrow C$  hold?
- **Solution:** Use a **tableau** to explore all inferences from the given set, to see if you can prove the target dependency.

# Main Points of Chasing a Tableau With MVD's and FD's

- To test  $A \rightarrow B$ ? Assume  $(a, b_1)$   $(a, b_2)$  to prove  $b_1 = b_2$
- To apply a FD, equate symbols, as before.
- To test  $A \rightarrow \rightarrow B$ ? Assume  $(a, b_1, \text{rest1})$   $(a, b_2, \text{rest2})$  to prove  $(a, b_2, \text{rest1})$  in R.
- To apply an MVD, generate one or both of the tuples we know must also be in the relation represented by the tableau.

**Example:**

if  $A \rightarrow BC$  and  $D \rightarrow C$ , then  $A \rightarrow C$ .

Goal: prove that  $c_1 = c_2$ .

$A$	$B$	$C$	$D$
$a$	$b_1$	<del><math>c_1</math></del> $c_2$	$d_1$
$a$	$b_2$	$c_2$	$d_2$
$a$	$b_2$	$c_2$	$d_1$

Use  $A \rightarrow BC$  (first row's  $D$  with second row's  $BC$ ).

Use  $D \rightarrow C$  (first and third row agree on  $D$ , therefore agree on  $C$ ).

## Example: Transitive Law for MVD's

- If  $A \twoheadrightarrow B$  and  $B \twoheadrightarrow C$ , then  $A \twoheadrightarrow C$ .
  - Obvious from the complementation rule if the Schema is  $ABC$ .
  - But it holds no matter what the schema; we'll assume  $ABCD$ .



# The Tableau for $A \rightarrow \rightarrow C$

Goal: derive tuple  $(a, b_1, c_2, d_1)$ .

$A$	$B$	$C$	$D$
$a$	$b_1$	$c_1$	$d_1$
$a$	$b_2$	$c_2$	$d_2$
$a$	$b_1$	$c_2$	$d_2$
$a$	$b_1$	$c_2$	

Use  $A \rightarrow \rightarrow B$  to swap  $B$  from the first row into the second.

Use  $B \rightarrow \rightarrow C$  to swap  $C$  from the third row into the first.

# Rules for Inferring MVD's + FD's

- Start with a tableau of **two rows**.
  - These rows agree on the attributes of **the left side** of the dependency to be inferred.
  - they **disagree** on all other attributes.
  - Use unsubscripted variables where they agree, subscripts where they disagree.

## Inference: Applying a FD

- Apply a FD  $X \rightarrow Y$  by finding rows that agree on all attributes of  $X$ . Force the rows to agree on all attributes of  $Y$ .
  - Replace one variable by the other.
  - If the replaced variable is part of the **goal tuple**, replace it there too.

## Inference: Applying a MVD

- Apply a MVD  $X \twoheadrightarrow Y$  by finding two rows that agree in  $X$ .
  - Add to the tableau one or both rows that are formed by swapping the  $Y$ -components of these two rows.

## Inference: Goals

- To test whether  $U \rightarrow V$  holds, we succeed by inferring that the two variables in each column of  $V$  are actually the same.
- If we are testing  $U \rightarrow \neg V$ , we succeed if we infer in the tableau a row that is the original two rows with the components of  $V$  swapped.

## Inference: Endgame

- Apply all the given FD's and MVD's until we cannot change the tableau.
- If we meet the goal, then the dependency is inferred.
- If not, then the final tableau is a counterexample relation.
  - Satisfies all given dependencies.
  - Original two rows violate target dependency.

# Relationships Among Normal Forms

Property	3NF	BCNF	4NF
Eliminates			
Redundancy due to FDs	Most	Yes	Yes
Eliminates redundancy			
Due to MVDs	No	No	Yes
Preserves FDs	Yes	maybe	maybe
Preserves MVD	maybe	maybe	maybe
Equal to original relation	Yes	Yes	Yes

# Relational Database Design

- Normal forms – “good” relation
- Design by decomposition, usually intuitive and works well
- Some Shortcomings:
  - a) Over-decomposition
  - b) Query workload
  - c) Dependency enforcement



# Summary of the chapter

- Functional dependencies
- Keys of a relation
- Minimal basis for a set of FD's
- BCNF and 3NF
- BCNF decomposition and 3NF synthesis (with lossless-join and dependency-preserving)
- Multivalued dependencies
- 4NF
- Reasoning about MVD and FD

# Homework

- Exercise 3.2.1
- Exercise 3.5.2
- Exercise 3.6.3 a), c)

ftp://public.sjtu.edu.cn to

public-files/upload/**chapter3**

User name: **fli** Password: **public**

Name of your homework is your **studentID**