

Relational Database Design Theory Part II

CPS 116
Introduction to Database Systems

Announcements (October 12)

- ❖ Midterm graded; sample solution available
- ❖ Homework #2 graded; sample solution already distributed
- ❖ Please verify all your grades on Blackboard
- ❖ Project milestone #1 due today

Review

- ❖ Functional dependencies
 - $X \rightarrow Y$: If two rows agree on X , they must agree on Y
 - ☞ A generalization of the key concept
- ❖ Non-key functional dependencies: a source of redundancy
 - Non-trivial $X \rightarrow Y$ where X is not a superkey
 - ☞ Called a BCNF violation
- ❖ BCNF decomposition: a method for removing redundancies
 - Given $R(X, Y, Z)$ and a BCNF violation $X \rightarrow Y$, decompose R into $R_1(X, Y)$ and $R_2(X, Z)$
 - ☞ A lossless join decomposition
- ❖ Schema in BCNF has no redundancy due to FD's

Next

4

- ❖ 3NF (BCNF is too much)
- ❖ Multivalued dependencies: another source of redundancy
- ❖ 4NF (BCNF is not enough)

Motivation for 3NF

5

- ❖ *Address* (*street_address*, *city*, *state*, *zip*)
 - $street_address, city, state \rightarrow zip$
 - $zip \rightarrow city, state$
- ❖ Keys
- ❖ BCNF?

To decompose or not to decompose

6

- Address₁* (*zip*, *city*, *state*)
- Address₂* (*street_address*, *zip*)
- ❖ FD's in *Address₁*
- ❖ FD's in *Address₂*
- ❖ Hey, where is $street_address, city, state \rightarrow zip$?
 - Cannot check without joining *Address₁* and *Address₂* back together
- ❖ Problem: Some lossless join decomposition is not dependency-preserving
- ❖ Dilemma: Should we get rid of redundancy at the expense of making constraints harder to enforce?

3NF

7

- ❖ R is in Third Normal Form (3NF) if for every non-trivial FD $X \rightarrow A$ (where A is single attribute), either
 - X is a superkey of R , or
 - A is a member of at least one key of R
- ☞ Intuitively, BCNF decomposition on $X \rightarrow A$ would “break” the key containing A
- ❖ So *Address* is already in 3NF
- ❖ Tradeoff:
 - Can enforce all original FD's on individual decomposed relations
 - Might have some redundancy due to FD's

BCNF = no redundancy?

8

- ❖ *Student* (SID , CID , $club$)
 - Suppose your classes have nothing to do with the clubs you join
 - FD's?
 - BCNF?
 - Redundancies?

SID	CID	$club$
142	CPS116	ballet
142	CPS116	sumo
142	CPS114	ballet
142	CPS114	sumo
123	CPS116	chess
123	CPS116	golf
...

Multivalued dependencies

9

- ❖ A multivalued dependency (MVD) has the form $X \twoheadrightarrow Y$, where X and Y are sets of attributes in a relation R
- ❖ $X \twoheadrightarrow Y$ means that whenever two rows in R agree on all the attributes of X , then we can swap their Y components and get two new rows that are also in R

X	Y	Z
a	$b1$	$c1$
a	$b2$	$c2$
a	$b1$	$c2$
a	$b2$	$c1$
...

} Must be in R too

MVD examples

10

Student (SID, CID, club)

Complete MVD + FD rules

11

- ❖ FD reflexivity, augmentation, and transitivity
- ❖ MVD complementation:
If $X \twoheadrightarrow Y$, then $X \twoheadrightarrow \text{attrs}(R) - X - Y$
- ❖ MVD augmentation:
If $X \twoheadrightarrow Y$ and $V \subseteq W$, then $XW \twoheadrightarrow YV$
- ❖ MVD transitivity:
If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then $X \twoheadrightarrow Z - Y$
- ❖ Replication (FD is MVD):
If $X \rightarrow Y$, then $X \twoheadrightarrow Y$ Try proving things using these!
- ❖ Coalescence:
If $X \twoheadrightarrow Y$ and $Z \subseteq Y$ and there is some W disjoint from Y such that $W \rightarrow Z$, then $X \rightarrow Z$

An elegant solution: chase

12

- ❖ Given a set of FD's and MVD's \mathcal{D} , does another dependency d (FD or MVD) follow from \mathcal{D} ?
- ❖ Procedure
 - Start with the hypothesis of d , and treat them as "seed" tuples in a relation
 - Apply the given dependencies in \mathcal{D} repeatedly
 - If we apply an FD, we infer equality of two symbols
 - If we apply an MVD, we infer more tuples
 - If we infer the conclusion of d , we have a proof
 - Otherwise, if nothing more can be inferred, we have a counterexample

Proof by chase

13

❖ In $R(A, B, C, D)$, does $A \twoheadrightarrow B$ and $B \twoheadrightarrow C$ imply that $A \twoheadrightarrow C$?

	Have				Need			
	A	B	C	D	A	B	C	D
	a	b1	c1	d1	a	b1	c2	d1
	a	b2	c2	d2	a	b2	c1	d2
$A \twoheadrightarrow B$	a	b2	c1	d1				
	a	b1	c2	d2				
$B \twoheadrightarrow C$	a	b2	c1	d2				
	a	b2	c2	d1				

Another proof by chase

14

❖ In $R(A, B, C, D)$, does $A \rightarrow B$ and $B \rightarrow C$ imply that $A \rightarrow C$?

	Have				Need
	A	B	C	D	$c1 = c2$
	a	b1	c1	d1	
	a	b2	c2	d2	
$A \rightarrow B$		$b1 = b2$			
$B \rightarrow C$			$c1 = c2$		

In general, both new tuples and new equalities may be generated

Counterexample by chase

15

❖ In $R(A, B, C, D)$, does $A \twoheadrightarrow BC$ and $CD \rightarrow B$ imply that $A \rightarrow B$?

	Have				Need
	A	B	C	D	$b1 = b2$
	a	b1	c1	d1	
	a	b2	c2	d2	
$A \twoheadrightarrow BC$	a	b2	c2	d1	
	a	b1	c1	d2	

Counterexample!

4NF

16

- ❖ A relation R is in Fourth Normal Form (4NF) if
 - For every non-trivial MVD $X \twoheadrightarrow Y$ in R , X is a superkey
 - That is, all FD's and MVD's follow from "key \rightarrow other attributes" (i.e., no MVD's and no FD's besides key functional dependencies)
- ❖ 4NF is stronger than BCNF
 - Because every FD is also a MVD

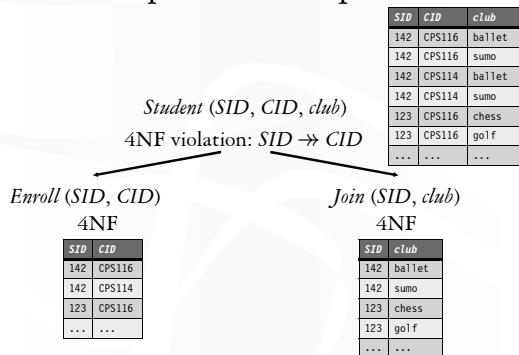
4NF decomposition algorithm

17

- ❖ Find a 4NF violation
 - A non-trivial MVD $X \twoheadrightarrow Y$ in R where X is not a superkey
- ❖ Decompose R into R_1 and R_2 , where
 - R_1 has attributes $X \cup Y$
 - R_2 has attributes $X \cup Z$ (Z contains attributes not in X or Y)
- ❖ Repeat until all relations are in 4NF
- ❖ Almost identical to BCNF decomposition algorithm
- ❖ Any decomposition on a 4NF violation is lossless

4NF decomposition example

18



3NF, BCNF, 4NF, and beyond

19

	3NF	BCNF	4NF
Lose FD's?			
Redundancy due to FD's			
Redundancy due to MVD's			

❖ Of historical interests

- 1NF: All column values must be atomic
- 2NF: Slightly more relaxed than 3NF

Summary

20

- ❖ Philosophy behind BCNF, 4NF:
Data should depend on the key, the whole key, and nothing but the key!
- ❖ Philosophy behind 3NF:
... But not at the expense of more expensive constraint enforcement!
