Relational Database Design Theory

Introduction to Databases

CompSci 316 Spring 2017



Announcements (Wed. Feb. 1)

• Homework #1 due Monday 02/06 (11:59 pm)

Review: Motivation

uid	uname	gid
142	Bart	dps
123	Milhouse	gov
857	Lisa	abc
857	Lisa	gov
456	Ralph	abc
456	Ralph	gov

- redundancy is bad
 - user name is recorded multiple times
- Leads to update, insertion, deletion anomalies
- Have a systematic approach to detecting and removing redundancy in designs
- Dependencies, decompositions, and normal forms

Review: Functional dependencies

- A functional dependency (FD) $X \rightarrow Y$
 - X and Y are sets of attributes in a relation R
- whenever two tuples in *R* agree on <u>all</u> the attributes in *X*, they must also agree on all attributes in *Y*

X	Y	Z
а	b	С
а	b	<i>c</i> 1
a1	b	с1

 $X \to Y$

 $XY \to Z$

NOTE: You can only say which FDs <u>do not hold</u> in an instance Cannot say which ones hold FDs are given by schema : must be true for all instances (like keys)

Review: Attribute closure

- Given
 - *R*
 - a set of FD's \mathcal{F} that hold in R, and
 - a set of attributes Z in R
- The closure of Z (denoted Z⁺) with respect to F is the set of all attributes {A₁, A₂, ... } functionally determined by Z
 - that is, $Z \rightarrow A_1 A_2 \dots$

uid \rightarrow uname, twitterid twitterid \rightarrow uid uid, gid \rightarrow fromDate

- {gid, twitterid}⁺ = ?
- twitterid \rightarrow uid ------ Closure grows to { gid, twitterid, uid }
- uid \rightarrow uname, twitterid ------ Closure grows to { gid, twitterid, uid, uname }
- uid, gid → fromDate ------ Closure is now all attributes in UserJoinsGroup

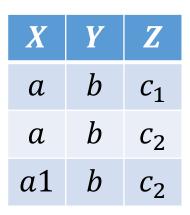
Review: Superkeys and Keys

Given a relation R and set of FD's \mathcal{F}

- Compute K^+ with respect to \mathcal{F}
- If *K*⁺ contains all the attributes of *R*, *K* is a super key
- If K is also minimal (no proper subset is a superkey),
 K is a key

Review: Motivation of BCNF decomposition

• Non-key FDs cause redundancy



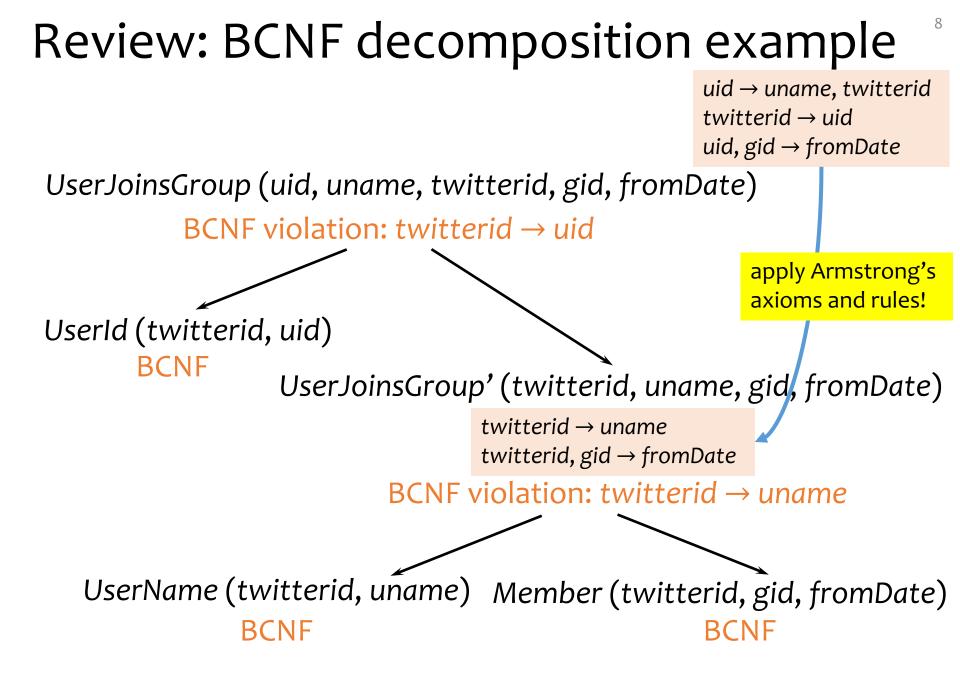
Here $X \to Y$

Detect such FDs where X is not a superkey, and decompose into two relations

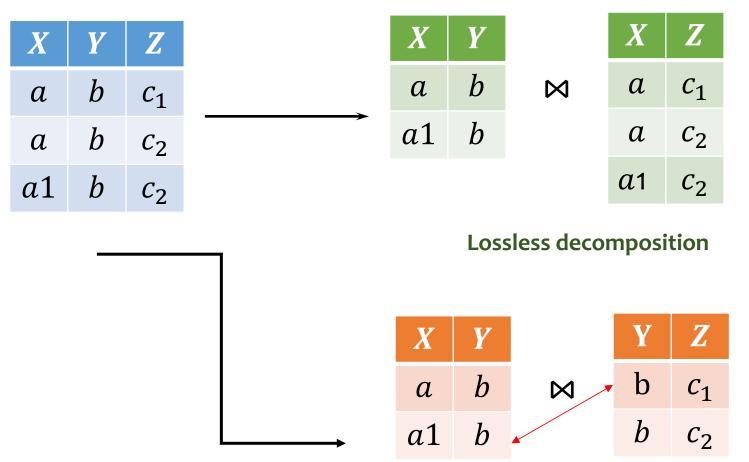
- 1. One relation gets X, Y
- 2. The other one gets X, Z

(X is a superkey there! this makes it lossless) (in general Z = everything else)

> Note: you need to consider all FDs that can be inferred! not only the ones that are given



Lossy and Lossless Decomposition



Check yourself! if in one of the two new relations, the common join attributes is a superkey, then lossless

Lossy decomposition

Review: Multi-valued Dependency motivation

- User (uid, gid, place)
- No FD like uid \rightarrow gid or uid \rightarrow place
- Still redundancy

uid	gid	place
142	dps	Springfield
142	dps	Australia
456	abc	Springfield
456	abc	Morocco
456	gov	Springfield
456	gov	Morocco

Given a user, gid and place are independent
 e.g. given uid = 456, all combinations exist for
 (abc, gov) x (Springfield, Morocco)

Multivalued dependencies

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

X	Y	Z
а	b_1	<i>C</i> ₁
а	<i>b</i> ₂	<i>C</i> ₂
а	<i>b</i> ₂	<i>C</i> ₁
а	<i>b</i> ₁	<i>c</i> ₂
•••	•••	•••

check yourself!

Complete MVD + FD rules

- FD reflexivity, augmentation, and transitivity
- MVD complementation: If $X \rightarrow Y$, then $X \rightarrow attrs(R) - X - Y$
- MVD augmentation: If $X \rightarrow Y$ and $V \subseteq W$, then $XW \rightarrow YV$
- MVD transitivity: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z - Y$
- Replication (FD is MVD): If $X \rightarrow Y$, then $X \rightarrow Y$
- Coalescence: If $X \rightarrow Y$ 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

- 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 premise 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

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

Have:	A	B	<i>C</i>	D
	а	b_1	C_1	d_1
		b_2		
$A \twoheadrightarrow B$		<i>b</i> ₂		
A "D	а	b_1	<i>C</i> ₂	d_2
$B \twoheadrightarrow C$	а	b_2	<i>c</i> ₁	d_2
D // C	а	b_2	<i>c</i> ₂	d_1
$B \twoheadrightarrow C$	а		<i>C</i> ₂	
<i>D "</i> C	а	b_1	C_1	d_2

Need:	A	B	С	D	
	а	b_1	<i>C</i> ₂	d_1	ef.
	а	b_2	<i>c</i> ₁	d_2	al l

Another proof by chase

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

 $a \quad b_1 \quad c_1 \quad d_1$

 $a b_2 c_2 d_2$

 $A \rightarrow B$ $b_1 = b_2$ $B \rightarrow C$ $c_1 = c_2$

Have:

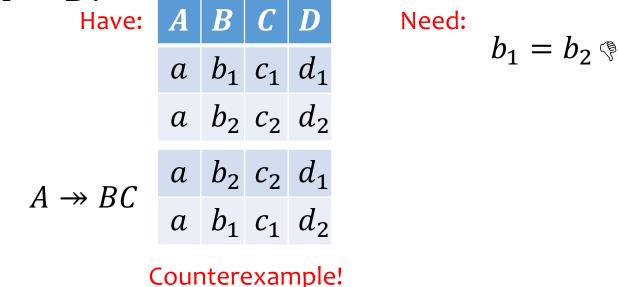
In general, with both MVD's and FD's, chase can generate both new tuples and new equalities

Need:

 $c_1 = c_2$ »

Counterexample by chase

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



Note: the FD must hold on all instances, so showing one instance as a counterexample suffices!

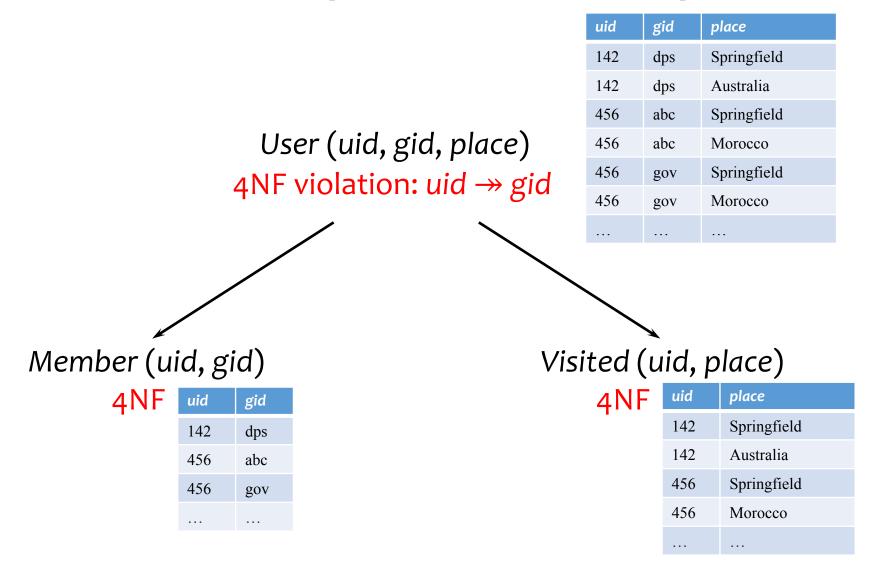
4NF

- A relation *R* is in Fourth Normal Form (4NF) if
 - For every non-trivial MVD $X \rightarrow Y$ in R, X is a superkey
 - That is, all FD's and MVD's follow from "key → 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
 - why? because trivially if two tuples have same X value, they also have the same Y value, no question in swapping the Y values!

4NF decomposition algorithm

- Find a 4NF violation
 - A non-trivial MVD $X \rightarrow Y$ 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 $X \cup Z$ (where Z contains R 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



Summary

- Philosophy behind BCNF, 4NF: Data should depend on the key, the whole key, and nothing but the key!
 - You could have multiple keys though



- Other normal forms
 - 3NF: More relaxed than BCNF; will not remove redundancy if doing so makes FDs harder to enforce
 - 2NF: Slightly more relaxed than 3NF
 - 1NF: All column values must be atomic

Next: Project Mixer!