Normalization

Anomalies

Boyce-Codd Normal Form

3rd Normal Form

1

Design Goals

- Goal of relational schema design is to avoid:
  - redundancy
  - anomalies.

2

Redundancy

- The same information can be extracted in multiple ways. Consequently:
  - Deleting the information must be sure to delete *all* representations of the information.
  - Updating the information must be kept *consistent* in all of the various ways.

3

Anomalies

- *Update anomaly* : one occurrence of a fact is changed, but not all occurrences.
- *Deletion anomaly* : valid fact is lost when a tuple is deleted.

4

Example of Bad Design

Drinkers(name, addr, beersLiked, manf, favBeer)

name	addr	beersLiked	manf	favBeer
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	???	WickedAle	Pete's	???
Spock	Enterprise	Bud	???	Bud

Data is **redundant**, because each of the ???'s can be figured out by using given FD's:

name -> addr favBeer
beersLiked -> manf

This Bad Design Also Exhibits Anomalies

name	addr	beersLiked	manf	favBeer
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	Voyager	WickedAle	Pete's	WickedAle
Spock	Enterprise	Bud	A.B.	Bud

- **Update** anomaly: if Janeway is transferred to *Intrepid*, e*ach* of her tuples need to be changed?
- **Deletion** anomaly: If nobody likes Bud, we lose track of the fact that Anheuser-Busch manufactures Bud.

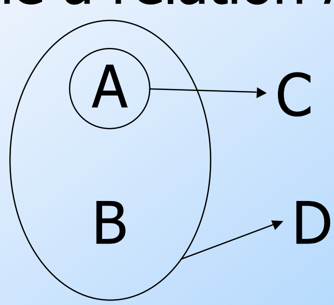
Boyce-Codd Normal Form

- We say a relation *R* is in *BCNF* if whenever *X* ->*A* is a *nontrivial* FD that holds in *R*, *X* is a *superkey*.
  - Remember: *nontrivial* means *A* is not a member of set *X*.
  - Remember, a *superkey* is any superset of a key (not necessarily a proper superset).

7

Explaining how bad things happen with non-BCNF (1)

Assume a relation ABCD



C doesn't depend on AB: update anomaly.
A is not a superkey, but A –>C is non-trivial.

8

Explaining how bad things happen with non-BCNF (2)

Assume a relation ABC

$A \to B \to C$

B is not a superkey, but B –>C is non-trivial.
A tuple ABC is required to represent just BC information.

9

Example

- Drinkers(name, addr, beersLiked, manf, favBeer)
- FD's: name->addr favBeer, beersLiked->manf
- Only key is {name, beersLiked}.
- In each FD, the left side is *not* a superkey.
- Any one of these FD's shows *Drinkers* is **not** in BCNF

10

Another Example

- Beers(name, manf, manfAddr)
- FD's: name->manf, manf->manfAddr
- Only key is {name}.
- name->manf does not violate BCNF, but manf->manfAddr does.

11

Decomposition into BCNF

- Given: relation *R* with FD's *F*.
- Look among the given FD's for a BCNF **violation** *X* ->*B*.
  - If any FD following from *F* violates BCNF, then there will surely be an FD in *F* itself that violates BCNF.
- Compute X^+.
  - Not all attributes, or else X is a superkey.

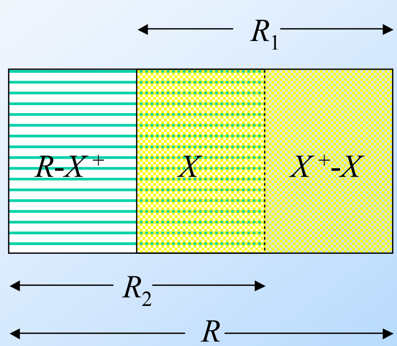
12

Decompose R Using X -> B

- Replace R by relations with schemas:
  1. $R_1 = X^+$.
  2. $R_2 = (R - X^+) \cup X$.
- Project given FD's F onto **two new relations**:
  1. Compute the closure of F = all nontrivial FD's that follow from F.
  2. Use only those FD's whose attributes are all in R_1 or all in R_2.

13

Decomposition Picture



14

Example

- Drinkers(name, addr, beersLiked, manf, favBeer)
- *F* = name->addr, name -> favBeer, beersLiked->manf
- Pick BCNF violation name->addr.
- Close the left side: $\{name\}^+$ = {name, addr, favBeer}.
- Decomposed relations:
  1. Drinkers1(name, addr, favBeer)
  2. Drinkers2(name, beersLiked, manf)

15

Example, Continued

- We are not done; we need to check Drinkers1 and Drinkers2 for BCNF.
- Projecting FD's is complex in general, easy here.
- For Drinkers1(name, addr, favBeer), relevant FD's are name->addr and name->favBeer.
  - Thus, *name* is the only key and Drinkers1 is in BCNF.

16

Example, Continued

- For Drinkers2(name, beersLiked, manf), the only FD is beersLiked->manf, and the only key is {name, beersLiked}.
  - Violation of BCNF.
- beersLiked^+ = {beersLiked, manf}, so we decompose *Drinkers2* into:
  1. Drinkers3(beersLiked, manf)
  2. Drinkers4(name, beersLiked)

17

Example, Concluded

- The resulting decomposition of *Drinkers* :
  1. Drinkers1(name, addr, favBeer)
  2. Drinkers3(beersLiked, manf)
  3. Drinkers4(name, beersLiked)
- Notice: *Drinkers1* tells us about drinkers, *Drinkers3* tells us about beers, and *Drinkers4* tells us the relationship between drinkers and the beers they like.

18

Third Normal Form - Motivation

- There is one structure of FD's that causes trouble when we decompose.
- *AB* -> *C* and *C* -> *B*.
  - Example: *A* = street address, *B* = city, *C* = zip code.
- There are two keys, {*A*,*B* } and {*A*,*C* }.
- *C* -> *B* is a BCNF violation, so we must decompose into *AC*, *BC*.

19

"Unenforceable" FD's

- The problem is that if we use *AC* and *BC* as our database schema, we **cannot enforce** the FD *AB* -> *C* by checking FD's in these decomposed relations.

- Example with *A* = street, *B* = city, and *C* = zip on the next slide.

20

An Unenforceable FD

street	zip
545 Tech Sq.	02138
545 Tech Sq.	02139

city	zip
Cambridge	02138
Cambridge	02139

Join tuples with equal zip codes.

street	city	zip
545 Tech Sq.	Cambridge	02138
545 Tech Sq.	Cambridge	02139

Although no FD's were violated in the decomposed relations, FD street city -> zip is violated by the database as a whole.

21

3NF Lets Us Avoid This Problem

- 3rd Normal Form (3NF) modifies the BCNF condition so we do not have to decompose in this problem situation.
- An attribute is *prime* if it is a member of *some* key.
- $X \to A$ violates 3NF if and only if X is not a superkey, and also A is not prime.

22

Example

- In our problem situation with FD's *AB* -> *C* and *C* -> *B*, we have keys *AB* and *AC*.
- Thus *A*, *B*, and *C* are each prime.
- Although *C* -> *B* violates BCNF, it does not violate 3NF.

23

What 3NF and BCNF Give You

- There are two important properties of a decomposition:
  1. ***Recovery*** : it should be possible to project the original relations onto the decomposed schema, and then reconstruct the original.
  2. ***Dependency preservation***: it should be possible to check in the projected relations whether all the given FD's are satisfied.

24

3NF and BCNF, Continued

- We can get (1) with a BCNF decompsition.
  - Explanation needs to wait for relational algebra.
- We can get both (1) and (2) with a 3NF decomposition.
- But we can't always get (1) and (2) with a BCNF decomposition.
  - street-city-zip is an example.

25