CS 235: Introduction to Databases

Svetlozar Nestorov

Lecture Notes #6

Normalization

- Improve the schema by decomposing relations and removing anomalies.
- Boyce-Codd Normal Form (BCNF): all FD's follow from the fact *key* $\rightarrow$ *everything*.
- Formally, R is in BCNF if every nontrivial FD for R, say $X \rightarrow A$, has X a superkey.
  - "Nontrivial" = right-side attribute not in left side.

BCNF properties

1. Guarantees no redundancy due to FD's.
2. Guarantees no *update anomalies* = one occurrence of a fact is updated, not all.
3. Guarantees no *deletion anomalies* = valid fact is lost when tuple is deleted.

Example

Beers(name, manf, manfAddr).

- FD's:
  - *name* $\rightarrow$ *manf,*
  - *manf* $\rightarrow$ *manfAddr.*
- Only key is *name.*
- *manf* $\rightarrow$ *manfAddr* violates BCNF with a left side unrelated to any key.

Decomposition into BCNF

- Setting: relation R, given FD's F. Suppose relation R has BCNF violation $X \rightarrow B$.
- We need only look among FD's of F for a BCNF violation.
- If there are no violations, then the relation is in BCNF.
- Don't we have to consider *implied* FD's?
- No, because…

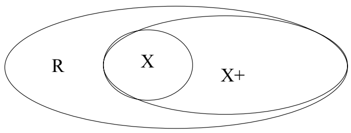
Proof

- Let $Y \rightarrow A$ is a BCNF violation and follows from F
- Then the computation of Y^+ used at least one FD $X \rightarrow B$ from F.
- X must be a subset of Y.
- Thus, if Y is not a superkey, X cannot be a superkey either, and $X \rightarrow B$ is also a BCNF violation.

Decomposition Algorithm (1/2)

For every violation $X \rightarrow B$ among given FD's:

1. Compute X^+.
   - Cannot be all attributes – why?
2. Decompose R into X^+ and $(R–X^+) \cup X$.



Decomposition Algorithm (2/2)

3. Find the FD's for the decomposed relations.
   – Project the FD's from F = calculate all consequents of F that involve only attributes from X^+ or only from $(R–X^+) \cup X$.

Example (1/3)

R = Drinkers(name, addr, beersLiked, manf, favoriteBeer)

F D's:

- *name → addr*
- *name → favoriteBeer*
- *beersLiked → manf*

Pick BCNF violation *name → addr*.

- Close the left side: *name* $^+$ = *name addr favoriteBeer*.
- Decomposed relations:
  Drinkers1(name, addr, favoriteBeer)
  Drinkers2(name, beersLiked, manf)
- Projected FD's (skipping a lot of work):
  – For *Drinkers1: name → addr* and *name → favoriteBeer*.
  – For *Drinkers2: beersLiked → manf*.

Example (2/3)

- BCNF violations?
  – For *Drinkers1, name* is key and all left sides of FD's are superkeys.
  – For *Drinkers2, {name, beersLiked}* is the key, and *beersLiked → manf* violates BCNF.

Example (3/3)

- Decompose *Drinkers2*
- Close *beersLiked* $^+$ = *beersLiked, manf*.
- Decompose:
  Drinkers3(beersLiked, manf)
  Drinkers4(name, beersLiked)
- Resulting relations are all in BCNF:
  Drinkers1(name, addr, favoriteBeer)
  Drinkers3(beersLiked, manf)
  Drinkers4(name, beersLiked)

Third Normal Form (3NF)

- Sometimes we have a dilemma:
  – If you decompose, you can't check the FD's in the decomposed relations.
  – If you don't decompose, you violate BCNF.
- Abstractly: $AB \rightarrow C$ and $C \rightarrow B$.
- In book: *title city → theatre* and *theatre → city.*
- Another example: *street city → zip, zip → city.*
- Keys: *AB* and *AC*, but $C \rightarrow B$ has a left side not a superkey.
- Suggests decomposition into *BC* and *AC*.
  – But you can't check the FD $AB \rightarrow C$ in these relations.

Example

- What can go wrong if we decompose:

A = street,
B = city,
C = zip.

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

city	zip
Cambridge	02138
Cambridge	02139

Join:

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

"Elegant" Workaround

- Define the problem away.
- A relation *R* is in 3NF iff for every nontrivial FD $X \rightarrow A$, either:
  1. *X* is a superkey, or
  2. *A* is *prime* = member of at least one key.
- Thus, the canonical problem goes away: you don't have to decompose because all attributes are prime.

Decomposition Properties

1. We should be able to recover from the decomposed relations the data of the original.
   – Recovery involves projection and join (next time).
2. We should be able to check that the FD's for the original relation are satisfied by checking the projections of those FD's in the decomposed relations.

3NF vs. BCNF

- Without proof, we assert that it is always possible to decompose into BCNF and satisfy (1).
- Also without proof, we can decompose into 3NF and satisfy both (1) and (2).
- But it is not possible to decompose into BCNF and get both (1) and (2).
  – Street-city-zip is an example of this point.

Multivalued Dependencies

- The *multivalued dependency* $X \rightarrow\rightarrow Y$ holds in a relation *R* if whenever we have two tuples of *R* that agree in all the attributes of *X*, then we can swap their *Y* components and get two new tuples that are also in *R*.

Example

- *Drinkers(name, addr, phones, beersLiked)*
- MVD *name* $\rightarrow\rightarrow$ *phones*.
- If *Drinkers* has the two tuples:

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

it must also have the same tuples with *phones* components swapped:

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

MVD Rules

- Every FD is an MVD: if $X \rightarrow Y$, then swapping *Y*'s between tuples that agree on *X* doesn't create new tuples.
- Example, in *Drinkers: name* $\rightarrow\rightarrow$ *addr*.
- *Complementation*: if $X \rightarrow\rightarrow Y$, then $X \rightarrow\rightarrow Z$, where *Z* is all attributes not in *X* or *Y*.
- Example: since *name* $\rightarrow\rightarrow$ *phones* holds in *Drinkers*, so does *name* $\rightarrow\rightarrow$ *addr beersLiked*.

Splitting Doesn't Hold

- Sometimes you need to have several attributes on the right of an MVD.
- For example: *Drinkers(name, areaCode, phones, beersLiked, beerManf)*

name	areaCode	phones	beersLiked	beerManf
Leo	773	555-1111	Bud	A.B.
Leo	773	555-1111	Honkers	G.I.
Leo	800	555-9999	Bud	A.B.
Leo	800	555-9999	Honkers	G.I.

- *name* $\rightarrow\rightarrow$ *areaCode phones* holds, but neither *name* $\rightarrow\rightarrow$ *areaCode* nor *name* $\rightarrow\rightarrow$ *phones* do.

Fourth Normal Form (4NF)

- Eliminate redundancy due to multiplicative effect of MVD's.
- Roughly: treat MVD's as FD's for decomposition, but not for finding keys.
- Formally: *R* is in Fourth Normal Form if whenever MVD $X \rightarrow\rightarrow Y$ is *nontrivial* (*Y* is not a subset of *X*, and $X \cup Y$ is not all attributes), then *X* is a superkey.
  - Remember, $X \rightarrow Y$ implies $X \rightarrow\rightarrow Y$, so 4NF is more stringent than BCNF.
- Decompose *R*, using 4NF violation $X \rightarrow\rightarrow Y$, into *XY* and $X \cup (R—Y)$.