Higher Forms of Normalization

Chapter 13.1-13.3
(skim)

"Lossless" Joins

- The main idea: if you decompose a relation schema, then join the parts of an instance via a natural join, you might get more rows than you started with, i.e., spurious tuples
  - This is bad!
  - Called a "lossy join".
- Goal: decompositions which produce only "lossless" joins
  - "non-additive" join is more descriptive

Preserving FDs

- What if, when a relation is decomposed, the X of an X→Y ends up only in one of the new relations and the Y ends up only in another?
- Such a decomposition is not dependency-preserving.
- Goal: Always have FD-preserving decompositions

Fact of life...

Finding a decomposition which is both lossless and dependency-preserving is not always possible.

Multivalued Dependencies (MVDs)

- X→→Y means that given X, there is a unique set of possible Y values (which do not depend on other attributes of the relation)
- Classic example: PARENTNAME→→CHILDNAME
- An FD is also a MVD
- MVD problems arise if there are two independent 1:N relationships in a relation.

Fourth Normal Form

- A relation R is in 4NF if for every nontrivial X→→Y, X is a superkey of R.
- Decomposition into 4NF: If there is a non-trivial X→→Y, form one relation with only X and Y, and another with R-Y.
- This will be lossless, but not necessarily FD-preserving.
  - Achieving 4NF is a trade-off

Fifth Normal Form

- Sometimes a relation cannot be losslessly decomposed into two relations, but can be into three or more.
- 5NF captures the idea that a relation scheme must have some particular lossless decomposition ("join dependency").
- Finding actual 5NF cases is difficult.

Normalization Summary

- 1NF: usually part of the woodwork
  - even so, know how to decompose
- 2NF: usually skipped
  - but lots of defs. that make great exam Q's!
- 3NF: a biggie
  - Always aim for this
- BCNF and 4NF: tradeoffs start here
  - in re: d-preserving and losslessness
- 5NF: You can say you've heard of it...

Caveat

- Normalization is not the be-all and end-all of DB design
- Example: suppose attributes A and B are always used together, but normalization theory says they should be in different tables.
  - Normalization might produce unacceptable performance loss (extra disk reads)