CSC 355 Database Systems
Lecture 11

Eric J. Schwabe

School of Computing, DePaul University

Spring 2020

Today:

- Relational Database Design
  - Functional Dependencies
  - Closures and Keys
  - Boyce-Codd Normal Form (BCNF)

Relational Database Design

- Start with a set of attributes

  $R = \{A_1, A_2, \ldots, A_n\}$

  - (Can also be written as a *universal relation*

    $R(A_1, A_2, \ldots, A_n) \ldots)$

- Construct a *decomposition* of R into relations

  $D = \{R_1, R_2, \ldots, R_m\}$

  - Each R_i is a subset of R

Relational Database Design

- The decomposition D = {R_1, R_2, …, R_m} should satisfy the following conditions:
  1. The union of the R_i's is R
  2. Redundancy has been removed from the R_i's
  3. Dependencies among attributes in R are preserved
  4. The original relation R can be recovered from D
- Conditions 2.-4. have to be formalized…

Redundancy

- *Redundancy* occurs when more than one record in a table stores the same information
  - Wastes space
  - Allows *update* and *deletion* anomalies
- We eliminate redundancy by identifying (and perhaps removing) functional dependencies in R

Functional Dependencies

- A set of attributes $Y = \{Y_1, Y_2, \ldots, Y_n\}$ is *functionally dependent* on a set of attributes $X = \{X_1, X_2, \ldots, X_m\}$ if and only if every pair of tuples that have the same values for X must also have the same values for Y
  - Also "X functionally determines Y" or "X → Y"
  - X is called the *determinant*
- (Less formally, "the values of X uniquely determine the values of Y"…)

Functional Dependencies

- "Every pair of tuples that have the same values on X also have the same values on Y"
  - For X to functionally determine Y, this condition must be *satisfied by every possible relation state*
  - If *some relation state does not satisfy* the condition because two tuples have the same values on X but different values on Y, then X does not functionally determine Y

Finding Functional Dependencies

- DVD (DVDID , MovieID , Title , Genre , Length , Rating)
- GRADING (CNumber , CTitle , SID , SName , Grade)
- PERSON (First , Last , Address , City , State , Zip)
- ASSIGNMENT (EID , ELName , EFName , Project, Hours)

Closures

- For F and a set of attributes X, the set X^+ is called the *closure of X (with respect to F).*
- X^+ is the set of all attributes that can be determined from X using anything in F
  - To find X^+: Start with just X … the add any other attributes you can determine from X using F … then add any other attributes you can determine from those … and so on … until you can't add any more.
- If X^+ includes the set Y, then X→Y can be *derived* from the set F

Equivalence

- Two sets F_1 and F_2 of functional dependencies are *equivalent* if and only if both of the following are true:
  - Every functional dependency in F_1 can be derived from the set F_2
  - Every functional dependency in F_2 can be derived from the set F_1
- Use closure to test each functional dependency

Definitions of Keys

- A set of attributes X is a *superkey* of R if X determines all attributes of R (i.e., if $X^+ = R$)
- A set of attributes X is a *candidate key* of R if X is a superkey, but no proper subset Y of X is a superkey
- An attribute is *prime* if it is contained in some candidate key (and is *non-prime* otherwise)

Eliminating Redundancy

- Functional dependencies whose determinants are not superkeys (i.e, that do not include candidate keys) indicate that there is redundancy in a relation
  - If there aren't any… then we're done!
  - If there are… then we use the functional dependencies to construct a decomposition that gets rid of the redundancy

BCNF

- A relation R is in Boyce-Codd Normal Form (BCNF) if for every non-trivial functional dependency X→Y in R, X is a superkey
  - "Every determinant must contain a candidate key"
  - A relation in BCNF will not have any redundancy, since every functional dependency in the relation will have a superkey as its determinant

Removing a Functional Dependency

- Suppose R contains the functional dependency X→Y where X is not a superkey
- Replace R with two relations:
  - R – Y
    - No longer contains X→Y
  - X ∪ Y
    - Contains X→Y, but X is a superkey in this relation
  - (Be sure that Y contains the complete closure of X…)

BCNF Decomposition Algorithm

Set D = {R}

While there is some Q in D that is not in BCNF:

Choose a Q that is not in BCNF

Find an X→Y in Q that violates BCNF

Replace Q with two relations:

Q – Y and (X union Y)

(When finished, all relations in D will be in BCNF)

Next:

- More Relational Database Design
  - Boyce-Codd Normal Form (BCNF)
  - Dependency preservation
  - Lossless join