Relational Design Theory: Anomalies

Problems arising from schemas with bad combinations of attributes:

- redundancy
- modification anomalies
- insert anomalies
- delete anomalies

Example: Loans = {<u>copy#, ISBN</u>, borrowerID, borrowerMajor, dept}

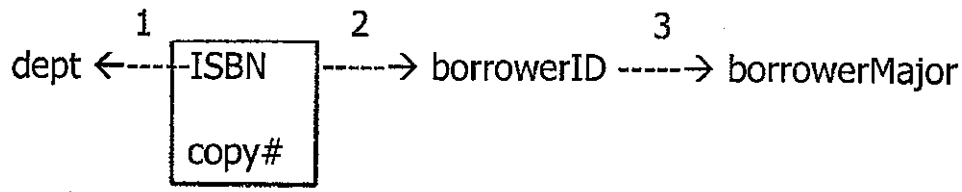
This table represents students who've presently borrowed particular copies of particular books from the university library. The "dept" attribute indicates the academic department responsible for that book.

All four anomalies arise from the functional dependency **ISBN -> dept**

- The instance that **0122-34-422** is the responsibility of **CSC** re-appears for each copy of the book. This is *redundancy.*
- If we wish to change the department for that book from **CSC** to **MAT**, that single update would need to be made in each row that mentions that ISBN. This is a *modification anomaly.*
- If we wished to add a new instance, that **2344-44-518** will be handled by **BIO**, we would be prevented from doing so unless we get a copy of the book (digression: why?). This is an *insert anomaly.*
- If we have only one copy of the book with ISBN **2231-15-2233**, and it's discarded, then we lose the fact that it's handled by **CHM**. This is a *delete anomaly.*

Note that a similar set of scenarios also arises from the FD **borrowerID -> borrowerMajor**.

Sources of Anomalies



1, 2, and 3 are **functional dependencies** (FD's).

Schemes are classified in **normal forms** (NF's); the higher, the better.

FD #1 is a **partial dependency**; as such, it prevents Loans from being in **2NF** or anything higher.

FD #3 is a **transitive dependency**; it therefore keeps Loans out of **3NF** or anything higher.

Families of NF's:

- 1NF
- 2NF, 3NF, BCNF
- 4NF
- 5NF

Two main theorems, informally stated:

1. You can decompose any schema in a good way into smaller 3NF ones.
2. You can decompose any schema in a pretty good way into smaller BCNF ones.

Sometimes we need to weigh the tradeoff between these two strategies.

Relational Design Theory – Some Formal Definitions

Assumptions: R is a relational schema, $\mathcal{F}$ is a set of FD's on R, X and K are subsets of R, A is an element of R, and when we assert that an FD holds, what we mean is that it is in $\mathcal{F}^+$.

X is a **determinant** if there is some A such that X -> A is nontrivial.

A is **prime** if it is a member of some key.

A is **partially dependent** on a key K if there is a nontrivial X -> A, where $X \subseteq K$ but X != K.

A is **transitively dependent** on a key K if there is a set X, such that K -> X and X -> A are nontrivial, A is not in K or X, and X does not determine K.

R is in **Second Normal Form** (2NF) if no nonprime attribute is partially dependent on any key.

R is in **Third Normal Form** (3NF) (DEFINITION 1) if it is in 2NF, and no nonprime attribute is transitively dependent on any key.

R is in **Third Normal Form** (3NF) (DEFINITION 2) if every determinant of every nonprime attribute is a superkey.

R is in **BCNF** if every determinant is a superkey.

Decompositions: a Little Practice

Classify the decomposition:

R = {A, B, C, D}
Decide whether each is FDP; is LJD.

- $\mathcal{F}$ = {A->D; B->C; C->AD}, $\mathcal{D}$ = {BC, ABD}
- $\mathcal{F}$ = {A->D; AB->C; C->AD}, $\mathcal{D}$ = {ABC, AD}

Practice with the Relational Synthesis Algorithms (Algorithms 11.2 and 11.4):

R = CSJDPQV
$\mathcal{F}$ = {C->CSJDPQV; JP->C; SD->P; J->S}