CIS4301 Notes: Database Design

Ryan Roden-Corrent

March 31, 2014

Note: Some class resources available here:
Database Systems Textbook
Inference Rules

1 Functional Dependencies and Normalization

1.1 Outline

- Informal Design Guidelines for Relation Schemas
- Functional Dependencies
- normal forms based on primary keys

1.2 Outline

- Levels at which we can discuss **goodness** of relational schemas
  - Logical (conceptual)
  - Implementation (physical storage)
- Approaches to DB design
  - Top down: start with large table, break down to details
  - Bottom Up: start with details, merge to create larger structure

1.3 Informal Design Guidelines

1.3.1 Measures of Quality

- make attribute semantics clear
- Reduce redundant info in tuples

- Reduce NULL values in tuples
- Disallow possibility of creating spurious tuples

1.3.2 Natural Join Example

Schema:
EMP_LOCS(Ename, Plocation)
EMP_PROJ(SSN, Pnumber, Hours, Pnme, Plocation)
Want to use natural join to find out how many hours each employee worked at a location
see this and this for the textbook problems
note that the natural join introduces duplicate information meaning is ambiguous.
update anomaly: need to update data in more than one spot. Sometimes these are unavoidable, so must be **documented well**.
Triggers are useful in these, but can get messy.

1.3.3 Redundant Information and Update Anomalies

- Types: Insertion, Deletion, Modifications
- Result of storing natural joins of base relations
- Significant effect on storage stapce

Deletion Anomaly Example:

Team Name	Player
NULL	12

Playerid	pts
12	2

Player 12 kicked off team, but still exists

Insertion Anomaly Example:

Team Name	Player
Gators	
Louisvile	
?	12

Must know team name before inserting player.

1.4 Why are NULLs bad?

- Way to group many attributes together into a "fat" relation
- Wasted storage space (reserves space in every column for that row)

Suppose an essay relation has a text field that is a char(40000). A null essay will still reserve this space.

1.4.1 Guideline 4

- Design relation schemas to be joined with equality conditions on attributes that are appropriately related
- guarantees no spurious tuples generated

1.5 Summary

- anomalies cause redundant work
- NULL wastes space

2 Functional Dependencies

denoted by $X \to Y$
for any two tuples t_1 and t_2 in r that have $t_1[X] = t_2[X]$, there must be a mapping $t_1[Y] = t_2[Y]$

	W	X	Y	Z
t_1		billy d	gators	
t_2		billy d	gators	

- formal tool for analysis of relational schemas
- detect and describe some of the above problems in precise terms
- theory of functinal dependency

2.1 normal forms based on primary keys

- normalization process
- Approaches for relational schema design
- takes a relation schema through a series of tests
  - certify whether it satisfies a certain normal form
  - proceeds in top-down fashion
- ideally only want functional dependencies on primary keys (but there is some leeway)

2.2 Normalization of Relations

- Nonadditive join Property
  - Extrememly critical
- Dependency preservation property
  - sometimes sacrificed for other factors

2.2.1 Keys and Attributes Participating in Keys

- definition of **superset** and **key**
- Candidate Key
- If more than one key in relation schema
  - one is primary
  - others are secondary

a **superkey** determines all of the attributes in a set.

2.3 Armstrong's axioms

Reflexive:
Augmentation:
Transitive: $S \to R, R \to T => S \to T$
Decomposition: $X \to YX => X \to Y, X \to Z$
Union:
PseudoTransitivity: $X \to Y, WY \to Z, WX \to Z$

2.3.1 Example

Decomposition: $pname \to pts, team$
$pname \to pts$
$pname \to team$

2.4 Closure

Full set of functional dependencies that can created given an initial set of functional dependencies, denoted **F+**.

2.5 Properties

Non-additive join shouldn't lose or gain information when performing a join

Dependency Preservation dependencies should hold through decomposition

2.6 Practical Normal Forms

- Resulting designs are high quality
- usually pay attention to normaliation up to 3NF, BCNF, of sometimes 4NF

2.6.1 First Normal Form

- only attribute values permitted are single (atomic) values

Example:

students	age	hometown
patriczy	21	{Jax, Warner Robins}

This violates 1NF as hometown is a collection, and therefore not atomic. Could be solved with hometown mapping table:

pyoung	21	jax
pyoung	21	Warner Robins

2.6.2 Second Normal Form

- based on **full functional dependency**
- nonprime attribtes are associated with only part of a primary key on which they are functionally dependent

Example:

movie name	star	film type	capacity
...	...	...	...

Functional dependencies:
$moviename \rightarrow star$
$filmtype \rightarrow Capacity$

Could split into two tables:

movie name	star
...	...

filmtype	capacity
...	...

Dependencies:
$moviename \rightarrow star$
$filmtype \rightarrow capacity$
$star \rightarrow filmtype$
Transitive rule relates filmtype to moviename.

2.6.3 Third Normal Form

- Be in first normal form
- Everything dependent on primary key
- Dependencies on primary key must not be transitive