Functional Dependencies, Normalization

Rose-Hulman Institute of Technology
Curt Clifton

Or…

Fixing Broken Database Designs

This material will almost certainly appear on Exam II next week.

Outline

- Functional Dependencies
- Keys Revisited
- Redundancy and Anomalies
- Normalization

Functional Dependencies (FD)

- Let X be a set of attributes of a relation R
- Let A be a single attribute of R
- $X \rightarrow A$ holds for R if:
  - whenever two tuples of R agree on all the attributes of X,
  - then they must also agree on the attribute A.
- We say X "uniquely determines" A in R

Example

- Customer(Name, Addr, SodaLiked, Manf, FavSoda), with name identifying a unique person
- Lots of redundancy here…

Name	**Addr**	**SodaLiked**	**Manf**	**FavSoda**
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does Name → Addr?

Name	**Addr**	**SodaLiked**	**Manf**	**FavSoda**
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does Name → Addr?
- Yes, since we assumed unique names

Name	Addr	SodaLiked	Manf	FavSoda
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does Name → FavSoda?

Name	Addr	SodaLiked	Manf	FavSoda
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does Name → FavSoda?
- Yes, we want just one favorite per person

Name	Addr	SodaLiked	Manf	FavSoda
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does SodaLiked → Manf?

Name	**Addr**	**SodaLiked**	**Manf**	**FavSoda**
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does SodaLiked → Manf?
- Yes, since each soda has just one manf.

Name	**Addr**	**SodaLiked**	**Manf**	**FavSoda**
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does FavSoda → Name?

Name	**Addr**	**SodaLiked**	**Manf**	**FavSoda**
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from Data

- Does FavSoda → Name?
- No, two people might have the same favorite

Name	Addr	SodaLiked	Manf	FavSoda
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

FDs from ER Diagrams

- From entity sets
  - (Key of entity set) → other attributes of entity set
- From many-one relationship
  - (Key of "many" set) → attributes of "one" set

Drawing FDs

- Use arrows to indicate FDs on schemas:

Notation Shorthand

- Technically FDs go from sets to single attributes
  - { Name } → Addr
  - { Name } → FavSoda
- Often just combine to write:
  - Name → Addr, FavSoda
- Usually omit set braces on left side also:
  - Restaurant, Soda → Price

Keys Revisited

- Let K be a set of attributes of a relation R
- K is a *super key* for R if:
  - For all attributes A in R, $K \rightarrow A$
- K is a *key* for R if:
  - No proper subset of K is a super key for R
- An attribute B is a prime attribute of R if:
  - B is an element of some key of R

Example

- What is the key here?
- What are the prime attributes?

Customer(Name, Addr, SodaLiked, Manf, FavSoda)

Two Ways to Find Keys

- Guess a superkey K:
  - Show that $K \rightarrow A$ for all attributes A
  - Show that no subset of K is a superkey
- Find all functional dependencies
  - Check all possible keys

Why Talk About FDs?

- Let us formally identify redundancy
- Tell us how to fix it!

Redundancy Leads to Anomalies

- *Update anomaly*: one occurrence of a fact is changed, but not all occurrences
- *Deletion anomaly*: valid fact is lost when a tuple is deleted

Example

Name	Addr	SodaLiked	Manf	FavSoda
Janeway	Voyager	Pepsi	PepsiCo	Coke
Janeway	Voyager	Sprite	CocaCola	Coke
Spock	Enterprise	Pepsi	PepsiCo	Coke

Redundant with first row since Name → Addr, FavSoda

Redundant with first row since SodaLiked → Manf

Normalization

- Using functional dependencies to eliminate redundancy
- An extremely powerful technique

Third Normal Form

- A relation R is in *Third Normal Form* (3NF) if whenever $X \rightarrow A$ is a nontrivial functional dependency that holds in R, then either:
  - X is a superkey for R, or
  - A is a prime attribute of R

Normalization Algorithm

- To normalize a relation R:
  - Find the functional dependencies for R
  - Check that whether each FD satisfies 3NF
    - If so, we're done and R is normalized
  - Otherwise let $X \rightarrow A$ be an FD that violates 3NF
    - Find the closure of X in R, denoted X^+
    - Split R into new relations (R - X^+ + X) and X^+
  - Repeat algorithm for each new relation

Example: Grades Relation

- Grade(Term, Yr, C#, Sec#, IName,
  SName, SAddr, S#, SSSN, Gr)

Step 1: Find the FDs

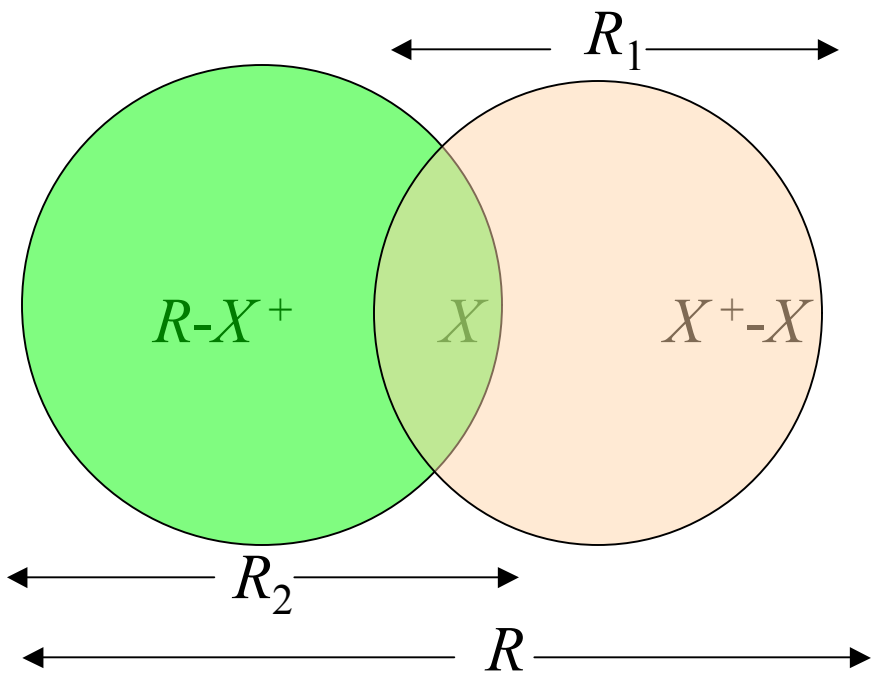
Step 2: Check for 3NF Violations

- A relation R is in *Third Normal Form* (3NF) if whenever $X \rightarrow A$ is a nontrivial functional dependency that holds in R, then either:
  - X is a superkey for R, or
  - A is a prime attribute of R

Step 3: Pick a Violating FD, Find Closure

- For $X \rightarrow A$ the *closure* of X, denoted X^+, is:
  - The set of all attributes that can be reached from any subset of X by following any FDs
- Or, just follow the arrows

Step 4: Split R into Two Relations



Repeat for the New Relations

- Find FDs
- Check for 3NF violations
- …