CS422 Principles of Database Systems
Functional Dependencies

Chengyu Sun
California State University, Los Angeles

Functional Dependency

- A functional dependency on relation R is the assertion that when two tuples agree on attributes $\{A_1,\ldots,A_n\}$, they must also agree on attribute B.
- $\{A_1,\ldots,A_n\} \rightarrow B$, or $\{A_1,\ldots,A_n\}$ functionally determine B

Example

- Drinkers(name, addr, beersLiked, manf, favBeer).
- Reasonable FD's to assert:
  1. name -> addr
  2. name -> favBeer
  3. beersLiked -> manf

Example Data

name	addr	beersLiked	manf	favBeer
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	Voyager	WickedAle	Pete's	WickedAle
Spock	Enterprise	Bud	A.B.	Bud

Because name -> addr

Because name -> favBeer

Because beersLiked -> manf

FD's With Multiple Attributes

- No need for FD's with > 1 attribute on right.
  - But sometimes convenient to combine FD's as a shorthand.
  - Example: name -> addr and name -> favBeer become name -> addr favBeer
- > 1 attribute on left may be essential.
  - Example: bar beer -> price

Keys of Relations

- $\{A_1,\ldots,A_n\}$ is a key of R if
  - $\{A_1,\ldots,A_n\}$ functionally determines all attributes of R
  - No proper subset of $\{A_1,\ldots,A_n\}$ functionally determines all attributes of R
- Superkey
- Note:
  - A relation may have multiple keys
  - A key many have multiple attributes
  - Minimal != Minimum
  - ER keys != relational keys

Example

- Consider relation Drinkers(name, addr, beersLiked, manf, favBeer).
- {name, beersLiked} is a superkey because together these attributes determine all the other attributes.
  - name -> addr favBeer
  - beersLiked -> manf

Example, Cont.

- {name, beersLiked} is a **key** because neither {name} nor {beersLiked} is a superkey.
  - name doesn't -> manf; beersLiked doesn't -> addr.
- In this example, there are no other keys, but lots of superkeys.
  - Any superset of {name, beersLiked}.

Discovering FDs and Keys

- Obvious ones
  - SSN, VIN, StudentID …
- Less obvious ones
  - (hour,room) →(class)
  - (playerID,year)→(teamID)
- Keys from ER
  - Entity Set
  - Binary relationships
  - Multi-way relationships

Trivial Functional Dependency

FD: $\{A_1,\ldots,A_n\} \rightarrow \{B_1,\ldots,B_m\}$

- FD is *trivial* if all B's are in **A**
- FD is *nontrivial* if at least one B is not in **A**
- FD is *completely nontrivial* if no B is in **A**

Armstrong's Axioms

- Reflexivity
  - If $\{B_1,\ldots,B_m\} \subseteq \{A_1,\ldots,A_n\}$, then $\mathbf{A}\rightarrow\mathbf{B}$
- Transitivity
  - If $\{A_1,\ldots,A_n\}\rightarrow\{B_1,\ldots,B_m\}$, and $\{B_1,\ldots,B_m\}\rightarrow\{C_1,\ldots,C_k\}$, then $\mathbf{A}\rightarrow\mathbf{C}$
- Augmentation
  - If $\{A_1,\ldots,A_n\}\rightarrow\{B_1,\ldots,B_m\}$, then $\{A_1,\ldots,A_n,C_1,\ldots,C_k\}\rightarrow\{B_1,\ldots,B_m,C_1,\ldots,C_k\}$

Closure of Attributes

- Given
  - a set of attributes **A**
  - a set of functional dependencies **S**
- Closure of **A** under **S**, $\mathbf{A}^+$, is the set of all possible attributes that are functionally determined by **A** based on the functional dependencies inferable from **S**

Simple Closure Example

- R: {A,B,C}
  - S: {A→B, B→C}
- $\{A\}^+$??
- $\{B\}^+$??
- $\{C\}^+$??

Computing $\mathbf{A}^+$

- Initialize $\mathbf{A}^+ = \mathbf{A}$
- Search in **S** for **B**→C where
  - $\mathbf{B} \subseteq \mathbf{A}^+$
  - $C \notin \mathbf{A}^+$
- Add C to $\mathbf{A}^+$
- Repeat until nothing can be added to $\mathbf{A}^+$

Computing $\mathbf{A}^+$ Example

- R: (A,B,C,D,E,F)
- S: (AB→C, BC→AD, D→E, CF →B)

- $\{A,B\}^+$??
- Is {A,B} a key ??

Correctness of the Closure Algorithm

- If $B \in \mathbf{A}^+$, then **A**→B
  - Proof by induction
- If **A**→B, then $B \in \mathbf{A}^+$
  - Proof by contradiction – if such B exists, that's because **A**→B cannot be inferred from **S**
    - If **A**→B is inferable from **S**, then all relations that satisfy **S** also satisfy **A**→B
    - A counter-example can be constructed where it satisfies **S** but not **A**→B

Projection

- We often want to break one relation into two or more relations
  - E.g. breaks (A,B,C,D) into (A,B,C) and (C,D)
- The resulting relations can be considered as *projections* of the original relation
- Given a set of FDs for the original relation, what can we say about the FDs of the projected relations?

Compute Functional Dependencies After Projection

- Let the new relation be R', compute the closures of all subset's of R's attributes, and exclude the FD's that involves the attributes that are projected out.
  - No need to compute the closures of the empty set and the full set
  - If $\mathbf{A}^+$ is already the set of all attributes, no need to compute the closures of A's superset

Example

- R:(A,B,C,D), S:(A→B,B→C,C→D)
- R':(A,C,D)

- A→C,A→D,C→D: *basis*
- A→C,C→D: *minimal basis*