CS422 Principles of Database Systems
Functional Dependency

Chengyu Sun
California State University, Los Angeles

Adapted from Jeffrey Ullman's lecture notes at http://www-db.stanford.edu/~ullman/dscb.html

Functional Dependency (FD)

- 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

FD Example

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

Drinkers

- Functional dependencies??

FD with Multiple Attributes

$$\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow B_1$$
$$\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow B_2$$
$$\ldots$$
$$\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow B_m$$

$$\Updownarrow$$

$$\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow \{B_1, B_2, B_3, \ldots, B_m\}$$

$$\Updownarrow$$

A → B

Trivial Functional Dependency

FD: $\{A_1, A_2, A_3, \ldots, A_n\} \rightarrow \{B_1, B_2, B_3, \ldots, B_m\}$

- 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**

Key

- **A** is a key of relation R if
  - **A** functionally determines all attributes of R
  - No proper subset of **A** functionally determines all attributes of R

Key Example

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

Drinkers

- Key??

A Few Things about Keys

- A relation may have multiple keys
- A key may consist of multiple attributes
- Superset of a key is called a super key
- A key has to be *minimal*, but not necessarily *minimum*
- The definition doesn't say anything about *uniqueness*

Discovering Keys

- Obvious ones
  - SSN, VIN, CIN …
- Less obvious ones
  - {hour, room} → class
  - {playerID, year} → team
- Keys from ER
- *Infer from functional dependencies*

From FD to Keys

R(A, B, C, D, E)
FD: A → B, B → C

??

{??} → {A, B, C, D, E}

Armstrong's Axioms

Reflexivity
If **B** ⊆ **A**, then **A** → **B**

Transitivity
If **A** → **B** and **B** → **C**, then **A** → **C**

Augmentation
If **A** → **B**, then **AC** → **BC** for any **C**

Why Transitivity Works

Let (a_1,b_1,c_1) and (a_2,b_2,c_2) be two tuples in R.

If $a_1 = a_2$,

??

So $c_1 = c_2$, and by definition of FD, A → C

Two More FD Rules

Union
If **A** → **B** and **A** → **C**, then **A** → **BC**

Decomposition
If **A** → **BC**, then **A** → **B** and **A** → **C**

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 ??

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

Compute Functional Dependencies for Projections

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

- *Closure of Functional Dependencies*
- A→C, A→D, C→D: *basis*
- A→C, C→D: *minimal basis*