ICS 321 Data Storage & Retrieval

Functional Dependencies

Prof. Lipyeow Lim
Information & Computer Science Department
University of Hawaii at Manoa

Example: Movies1

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

- What are the keys for this relation ?
- What if you ignore the column starName ?
- Can starName be a key ?

Functional Dependency

- A <u>functional dependency</u> X -> Y holds over relation R if, for every allowable instance *r* of R:
  - for all tuples *t1,t2* in *r,*

$$\pi_X(t1) = \pi_X(t2) \text{ implies } \pi_Y(t1) = \pi_Y(t2)$$

  - i.e., given two tuples in *r*, if the X values agree, then the Y values must also agree. (X and Y are *sets* of attributes.)
- An FD is a statement about *all* allowable instances.
  - Must be identified based on semantics of application.
  - Given some allowable instance *r1* of R, we can check if it violates some FD *f*, but we cannot tell if *f* holds over R!
- K is a candidate key for R means that K -> R
  - However, K -> R does not require K to be *minimal*!

Keys & Superkeys

- A set of one or more attributes $\{A_1, A_2, \dots A_n\}$ is a **key** for a relation R if :
  - 1 . Those attributes *functionally determine* all other attributes of the relation .
  - 2 . No *proper subset* of $\{A_1, A_2, \dots A_n\}$ functionally determines all other attributes of R
    - a key must be minimal .
- When a key consists of a single attribute A , we often say that A (rather than {A}) is a key.
- **Superkey** : a set of attributes that contain a key.

FD Example: Movies1

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

- What are the FDs for this relation ?
- What are the keys for this relation ?
- Can starName be a key ?

Reasoning about FDs

- Given some FDs, we can usually infer additional FDs:
  - *ssn -> deptID, deptID -> building* implies *ssn -> building*
- T implies S, or S follows from T
  - Every relation instance that satisfies all the FDs in T also satisfies all the FDs in S
- S is equivalent to T
  - The set of relation instances satisfying S is exactly the same as the set satisfying T
  - Alternatively, S implies T AND T implies S

Armstrong's Axioms

Let X, Y, Z are sets of attributes:

- *Reflexivity*
  - If X is a subset of Y, then Y -> X
- *Augmentation*
  - If X -> Y, then XZ -> YZ for any Z
- *Transitivity*
  - If X -> Y and Y -> Z, then X -> Z

These are *sound* and *complete* inference rules for FDs!

Example: Armstrong's Axioms

Hourly_Emps

SSN	Name	Lot	Rating	Hourly_Wages	Hours_worked
123-22-2366	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

- *Reflexivity*: If X is a subset of Y, then Y -> X
  - SNLR is a subset of SNLRWH, SNLRWH -> SNLR
- *Augmentation*: If X -> Y, then XZ -> YZ for any Z
  - S -> N, then SLR -> NLR
- *Transitivity*: If X -> Y and Y -> Z, then X -> Z
  - S -> R, R -> W, then S -> W

Two More Rules

Firstname	Lastname	DOB	Address	Telephone
John	Smith	Sep 9 1979	Honolulu,HI	808-343-0809

- **Union / Combining**
  - If X → Y and X → Z, then X → YZ
  - Eg. FLD → A and FLD → T, then FLD → AT
- **Decomposition / Splitting**
  - If X → YZ, then X → Y and X → Z
  - Eg. FLD → AT , then FLD → A and FLD → T
- **Trivial FDs**
  - Right side is a subset of Left side
  - Eg. F → F, FLD → FD
- Does "XY → Z imply X →Z and Y →Z" ?

Closure

- **Implication**: An FD *f* is *implied by* a set of FDs *F* if *f* holds whenever all FDs in *F* hold.
  - f=A →C is implied by F={ A→B, B →C} (using Armstrong's transitivity)
- **Closure F⁺** : the set of all FDs implied by F
  - Algorithm:
    - start with $F^+ = F$
    - keep adding new implied FDs to F^+ by applying the 5 rules (Armstrong's Axioms + union + decomposition)
    - Stop when F^+ does not change anymore.

Example: Closure

Firstname	Lastname	DOB	Street	CityState	Zipcode	Telephone
John	Smith	Sep 9 1979	1680 East West Rd.	Honolulu,HI	96822	808-343-0809

- Given FLD is the primary key and C → Z
- Find the closure:
  - Start with { FLD → FLDSCZT, C→Z }
  - Applying reflexivity, { FLD → F, FLD →L, FLD → D, FLD → FL, FLD → LD, FLD →DF, FLDSCZT → FLD, …}
  - Applying augmentation, { FLDS → FS, FLDS → LS, …}
  - Applying transitivity …
  - Applying union …
  - Applying decomposition …
  - Repeat until F^+ does not change

Attribute Closure

- Computing the closure of a set of FDs can be expensive. (Size of closure is exponential in # attrs!)
- Typically, we just want to check if a given FD $X \rightarrow Y$ is in the closure of a set of FDs *F*. An efficient check:
  - Compute *attribute closure* of X (denoted X^+) wrt *F:*
    - Set of all attributes A such that $X \rightarrow A$ is in F^+
    - There is a linear time algorithm to compute this.
  - Check if Y is in X^+
- Does $F = \{A \rightarrow B,\ B \rightarrow C,\ C\ D \rightarrow E\}$ imply $A \rightarrow E$?
  - i.e, is $A \rightarrow E$ in the closure F^+ ? Equivalently, is E in A^+ ?