

## **Functional Dependency**

Murat Kantarcioglu

## **UTD** Functional Dependencies

- Let *R* be a relation schema  $\alpha \subseteq R$  and  $\beta \subseteq R$
- The functional dependency

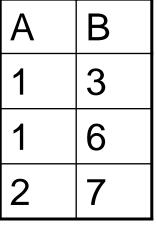
 $\alpha \rightarrow \beta$ holds on *R* if and only if for any legal relations *r*(R), whenever any two tuples  $t_1$  and  $t_2$  of *r* agree on the attributes  $\alpha$ , they also agree on the attributes  $\beta$ . That is,

$$t_1[\alpha] = t_2[\alpha] \implies t_1[\beta] = t_2[\beta]$$





• Example: Consider r(A,B) with the following instance of r.



• On this instance,  $A \rightarrow B$  does **NOT** hold, but  $B \rightarrow A$  does hold.





Α	В	С	D
a1	b1	c1	d1
a1	b1	c1	d2
a1	b2	c2	d1
a2	b1	с3	d1

- Does AB  $\rightarrow$  C hold?
- Does ABC  $\rightarrow$  D hold ?
- Does BC  $\rightarrow$  D hold?





SSN	LastName	FirstName	City
111111111	Smith	John	Richardson
222222222	Li	Peng	Richardson
333333333	Kant	John	Plano
44444444	Smith	Mark	Plano

- Does {ssn}  $\rightarrow$  {LastName} hold?
- Does {ssn} → {LastName,FirstName} hold ?
- Does {LastName, FirstName}→ {City} hold?
- Does  $\{City\} \rightarrow \{FirstName\} \text{ hold} \}$

## **UTD** Procedure for Computing F<sup>+</sup>

- $F^+ = F$
- repeat

for each functional dependency f in F<sup>+</sup>
 apply reflexivity and augmentation rules on f
 add the resulting functional dependencies to F<sup>+</sup>

- for each pair of functional dependencies  $f_1$  and  $f_2$  in  $F^+$ if  $f_1$  and  $f_2$  can be combined using transitivity then add the resulting functional dependency to  $F^+$
- until F<sup>+</sup> does not change any further





- R = (A, B, C, G, H, I) $F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$
- some members of F+
  - $-A \rightarrow H$ 
    - by transitivity from  $A \rightarrow B$  and  $B \rightarrow H$
  - $-AG \rightarrow I$ 
    - by augmenting  $A \rightarrow C$  with G, to get  $AG \rightarrow CG$ and then transitivity with  $CG \rightarrow I$
  - $-CG \rightarrow HI$ 
    - by augmenting CG → I to infer CG → CGI, and augmenting of CG → H to infer CGI → HI, and then transitivity



- Given a set of attributes α, define the *closure* of α under *F* (denoted by α<sup>+</sup>) as the set of attributes that are functionally determined by α under *F*
- Algorithm to compute α<sup>+</sup>, the closure of α under F result := α;
   while (changes to result) do for each β → γ in F do begin
   if β ⊆ result then result := result ∪ γ
  end



## Example of Attribute Set Closure

- R = (A, B, C, G, H, I)
- $F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$
- (*AG*)+

1. result = AG 2. result = ABCG  $(A \rightarrow C \text{ and } A \rightarrow B)$ 3. result = ABCGH  $(CG \rightarrow H \text{ and } CG \subseteq AGBC)$ 4. result = ABCGHI  $(CG \rightarrow I \text{ and } CG \subseteq AGBCH)$ 

• Is AG a candidate key?

1. Is AG a super key?

1. Does  $AG \rightarrow R? == Is (AG)^+ \supseteq R$ 

2. Is any subset of AG a superkey?

1. Does  $A \rightarrow R? ==$  Is  $(A)^+ \supseteq R$ 

2. Does  $G \rightarrow R$ ? == Is (G)<sup>+</sup>  $\supseteq$  R