>>

Inference on FDs

- Closures
- Determining Keys
- Minimal Covers

∧ >>

❖ Closures

Given a set F of *fd*s, how many new *fd*s can we derive?

For a finite set of attributes, there must be a finite set of derivable *fd*s.

The largest collection of dependencies that can be derived from F is called the closure of F and is denoted F^+.

Closures allow us to answer two interesting questions:

- is a particular dependency $X \rightarrow Y$ derivable from F?
- are two sets of dependencies F and G equivalent?

❖ Closures (cont)

For the question "is $X \rightarrow Y$ derivable from F?" ...

- compute the closure F^+; check whether $X \rightarrow Y \in F^+$

For the question "are F and G equivalent?" ...

- compute closures F^+ and G^+; check whether they're equal

Unfortunately, closures can be very large, e.g.

$$
\begin{aligned}
&R = ABC, \quad F = \{ AB \rightarrow C,\ C \rightarrow B \} \\
&F^+ = \{ A \rightarrow A,\ AB \rightarrow A,\ AC \rightarrow A,\ AB \rightarrow B,\ BC \rightarrow B,\ ABC \rightarrow B, \\
&\qquad C \rightarrow C,\ AC \rightarrow C,\ BC \rightarrow C,\ ABC \rightarrow C,\ AB \rightarrow AB,\ \ldots\ldots, \\
&\qquad AB \rightarrow ABC,\ AB \rightarrow ABC,\ C \rightarrow B,\ C \rightarrow BC,\ AC \rightarrow B,\ AC \rightarrow AB \}
\end{aligned}
$$

<< ∧ >>

❖ Closures (cont)

Algorithms based on F^+ rapidly become infeasible.

To solve this problem ...

- use closures based on sets of attributes rather than sets of *fd*s.

Given a set X of attributes and a set F of *fd*s, the closure of X (denoted X^+) is

- the largest set of attributes that can be derived from X using F

Determining X+ from {X→Y, Y→Z} ... X → XY → XYZ = X+

For computation, $|X^+|$ is bounded by the number of attributes.

<< ^ >>

❖ Closures (cont)

Algorithm for computing attribute closure:

```
Input: F (set of FDs), X (starting attributes)
Output: X+ (attribute closure)

Closure = X
while (not done) {
   OldClosure = Closure
   for each A → B such that A ⊂ Closure
      add B to Closure
   if (Closure == OldClosure) done = true
}
```

❖ Closures (cont)

For the question "is $X \rightarrow Y$ derivable from F?" ...

- compute the closure X^+, check whether $Y \subset X^+$

For the question "are F and G equivalent?" ...

- for each dependency in G, check whether derivable from F
- for each dependency in F, check whether derivable from G
- if true for all, then $F \Rightarrow G$ and $G \Rightarrow F$ which implies $F^+ = G^+$

For the question "what are the keys of R implied by F?" ...

- find subsets $K \subset R$ such that $K^+ = R$

<< ^ >>

❖ Determining Keys

Example: determine primary keys for each of the following:

1. $FD = \{ A \rightarrow B, C \rightarrow D, E \rightarrow FG \}$
    - A? A+ = AB, so no ... AB? AB+ = ABCD, so no
    - ACE? ACE+ = ABCDEFG, so yes!

2. $FD = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D \}$
    - B? B+ = BCD, so no ... A? A+ = ABCD, so yes!

3. $FD = \{ A \rightarrow B, B \rightarrow C, C \rightarrow A \}$
    - A? A+ = ABC, so yes! ... B? B+ = ABC, so yes!

<< ^ >>

❖ Minimal Covers

For a given application, we can define many different sets of *fd*s with the same closure (e.g. F and G where $F^+ = G^+$)

Which one is best to "model" the application?

- any model has to be complete (i.e. capture entire semantics)
- models should be as small as possible
  (we use them to check DB validity after update; less checking is better)

If we can ...

- determine a number of candidate *fd* sets, F, G and H
- establish that $F^+ = G^+ = H^+$
- we would then choose the smallest one for our "model"

Better still, can we *derive* the smallest complete set of *fd*s?

<< ⋀ >>

❖ Minimal Covers (cont)

Minimal cover F_c for a set F of *fd*s:

- F_c is equivalent to F
- all *fd*s have the form $X \to A$ (where A is a single attribute)
- it is not possible to make F_c smaller
  - either by deleting an *fd*
  - or by deleting an attribute from an *fd*

An *fd* d is redundant if $(F\text{-}\{d\})^+ = F^+$

An attribute a is redundant if $(F\text{-}\{d\} \cup \{d'\})^+ = F^+$
(where d' is the same as d but with attribute A removed)

<< ∧ >>

❖ Minimal Covers (cont)

Algorithm for computing minimal cover:

Inputs: set F of *fd*s
Output: minimal cover F_c of F

$F_c = F$

Step 1: put $f \in F_c$ into canonical form

Step 2: eliminate redundant attributes from $f \in F_c$

Step 3: eliminate redundant *fd*s from F_c

Step 1: put *fd*s into canonical form

for each $f \in F_c$ like $X \rightarrow \{A_1,...,A_n\}$
    remove $X \rightarrow \{A_1,...,A_n\}$ from F_c
    add $X \rightarrow A_1$, ... $X \rightarrow A_n$ to F_c
end

<< ^ >>

❖ Minimal Covers (cont)

Step 2: eliminate redundant attributes

```
for each f ∈ Fc like X → A
    for each b in X
        f' = (X-{b}) → A;   G = Fc - {f} ∪ {f'}
        if (G+ == Fc+) Fc = G
    end
end
```

Step 3: eliminate redundant functional dependencies

```
for each f ∈ Fc
    G = Fc - {f}
    if (G+ == Fc+) Fc = G
end
```

<< ^

❖ Minimal Covers (cont)

Example: compute minimal cover

E.g. $R = ABC$, $F = \{ A \to BC,\ \ B \to C,\ \ A \to B,\ \ AB \to C \}$

Working ...

- canonical *fd*s: $A \to B,\ \ A \to C,\ \ B \to C,\ \ AB \to C$
- redundant attrs: $A \to B,\ \ A \to C,\ \ B \to C,\ \ AB \to C$
- redundant *fd*s: $A \to B,\ \ A \to C,\ \ B \to C$

This gives the minimal cover $F_C = \{ A \to B,\ \ B \to C \}$.

Produced: 25 Mar 2021