Lecture 11:
Design Theory II

Today's Lecture

1. Closures, superkeys & keys
   - ACTIVITY: The key or a key?
2. Boyce-Codd Normal Form
   - ACTIVITY
3. Decompositions & 3NF
   - ACTIVITY
4. MVDs
   - ACTIVITY

Closures (بستار)

Closure Algorithm

Start with X = $\{A_1, \ldots, A_n\}$, FDs F.
Repeat until X doesn't change; **do**:
if $\{B_1, \ldots, B_n\} \rightarrow C$ is in F **and** $\{B_1, \ldots, B_n\} \subseteq X$:
then add C to X.
Return X as X^+

```
{name, category}+ =
{name, category}
```

F =

```
{name} → {color}

{category} → {dept}

{color, category} →
{price}
```

Closure Algorithm

Start with X = $\{A_1, \ldots, A_n\}$, FDs F.
Repeat until X doesn't change; **do**:
if $\{B_1, \ldots, B_n\} \rightarrow C$ is in F **and** $\{B_1, \ldots, B_n\} \subseteq X$:
then add C to X.
Return X as X^+

`{name, category}+ =`
`{name, category}`

`{name, category}+ =`
`{name, category, color}`

F =

```
{name} → {color}

{category} → {dept}

{color, category} → {price}
```

Closure Algorithm

Start with X = $\{A_1, \ldots, A_n\}$, FDs F.
Repeat until X doesn't change; **do**:
  if $\{B_1, \ldots, B_n\} \rightarrow C$ is in F **and** $\{B_1, \ldots, B_n\} \subseteq X$:
    then add C to X.
Return X as X^+

F =

```
{name} → {color}
{category} → {dept}
{color, category} → {price}
```

```
{name, category}+ =
{name, category}
```

```
{name, category}+ =
{name, category, color}
```

```
{name, category}+ =
{name, category, color, dept}
```

Closure Algorithm

Start with X = $\{A_1, \ldots, A_n\}$, FDs F.
Repeat until X doesn't change; **do**:
if $\{B_1, \ldots, B_n\} \rightarrow C$ is in F **and** $\{B_1, \ldots, B_n\} \subseteq X$:
then add C to X.
Return X as X^+

F =

```
{name} → {color}

{category} → {dept}

{color, category} → {price}
```

```
{name, category}+ =
{name, category}
```

```
{name, category}+ =
{name, category, color}
```

```
{name, category}+ =
{name, category, color, dept}
```

```
{name, category}+ =
{name, category, color, dept, price}
```

Example

`R(A,B,C,D,E,F)`

```
{A,B} → {C}
{A,D} → {E}
{B} → {D}
{A,F} → {B}
```

Compute $\{A,B\}^+ = \{A, B, \qquad\qquad \}$

Compute $\{A, F\}^+ = \{A, F, \qquad\qquad \}$

Example

R(A,B,C,D,E,F)

$\{A,B\} \rightarrow \{C\}$
$\{A,D\} \rightarrow \{E\}$
$\{B\} \rightarrow \{D\}$
$\{A,F\} \rightarrow \{B\}$

Compute $\{A,B\}^+ = \{A, B, C, D \qquad\qquad \}$

Compute $\{A, F\}^+ = \{A, F, B \qquad\qquad \}$

Example

```
R(A,B,C,D,E,F)
```

$\{A,B\} \rightarrow \{C\}$
$\{A,D\} \rightarrow \{E\}$
$\{B\} \rightarrow \{D\}$
$\{A,F\} \rightarrow \{B\}$

Compute $\{A,B\}^+ = \{A, B, C, D, E\}$

Compute $\{A, F\}^+ = \{A, B, C, D, E, F\}$

1. Closures, Superkeys & Keys

What you will learn about in this section

1. Closures Pt. II
2. Superkeys & Keys
3. ACTIVITY: The key or a key?

Why Do We Need the Closure?

- With closure we can find all FD's easily

- To check if $X \rightarrow A$

  1. Compute X^+

  2. Check if $A \in X^+$

Note here that **X** is a *set* of attributes, but **A** is a *single* attribute. Why does considering FDs of this form suffice?

Recall the <u>Split/combine</u> rule:
$X \rightarrow A_1, \ldots, X \rightarrow A_n$
implies
$X \rightarrow \{A_1, \ldots, A_n\}$

Using Closure to Infer ALL FDs

Example:
Given F =

```
{A,B} → C
{A,D} → B
{B}   → D
```

Step 1: Compute X^+, for every set of attributes X:

```
{A}+ = {A}
{B}+ = {B,D}
{C}+ = {C}
{D}+ = {D}
{A,B}+ = {A,B,C,D}
{A,C}+ = {A,C}
{A,D}+ = {A,B,C,D}
{A,B,C}+ = {A,B,D}+ = {A,C,D}+ = {A,B,C,D}
{B,C,D}+ = {B,C,D}
{A,B,C,D}+ = {A,B,C,D}
```

No need to compute all of these- why?

Using Closure to Infer ALL FDs

<u>Example:</u>
Given F =

```
{A,B} → C
{A,D} → B
{B}   → D
```

Step 1: Compute X^+, for every set of attributes X:

```
{A}+ = {A}, {B}+ = {B,D}, {C}+ = {C}, {D}+ =
{D}, {A,B}+ = {A,B,C,D}, {A,C}+ = {A,C},
{A,D}+ = {A,B,C,D}, {A,B,C}+ = {A,B,D}+ =
{A,C,D}+ = {A,B,C,D}, {B,C,D}+ = {B,C,D},
{A,B,C,D}+ = {A,B,C,D}
```

Step 2: Enumerate all FDs X → Y, s.t. $Y \subseteq X^+$ and $X \cap Y = \emptyset$:

```
{A,B} → {C,D}, {A,D} → {B,C},
{A,B,C} → {D}, {A,B,D} → {C},
{A,C,D} → {B}
```

Using Closure to Infer ALL FDs

Example:

Given F =

```
{A,B} → C
{A,D} → B
{B}   → D
```

Step 1: Compute X^+, for every set of attributes X:

```
{A}+ = {A}, {B}+ = {B,D}, {C}+ = {C}, {D}+ =
{D}, {A,B}+ = {A,B,C,D}, {A,C}+ = {A,C},
{A,D}+ = {A,B,C,D}, {A,B,C}+ = {A,B,D}+ =
{A,C,D}+ = {A,B,C,D}, {B,C,D}+ = {B,C,D},
{A,B,C,D}+ = {A,B,C,D}
```

Step 2: Enumerate all FDs X → Y, s.t. $Y \subseteq X^+$ and $X \cap Y = \emptyset$:

```
{A,B} → {C,D}, {A,D} → {B,C},
{A,B,C} → {D}, {A,B,D} → {C},
{A,C,D} → {B}
```

"Y is in the closure of X"

Using Closure to Infer ALL FDs

Example:

Given F =

```
{A,B} → C
{A,D} → B
{B}   → D
```

Step 1: Compute X^+, for every set of attributes X:

```
{A}+ = {A}, {B}+ = {B,D}, {C}+ = {C}, {D}+ =
{D}, {A,B}+ = {A,B,C,D}, {A,C}+ = {A,C},
{A,D}+ = {A,B,C,D}, {A,B,C}+ = {A,B,D}+ =
{A,C,D}+ = {A,B,C,D}, {B,C,D}+ = {B,C,D},
{A,B,C,D}+ = {A,B,C,D}
```

Step 2: Enumerate all FDs X → Y, s.t. $Y \subseteq X^+$ and $X \cap Y = \emptyset$:

```
{A,B} → {C,D}, {A,D} → {B,C},
{A,B,C} → {D}, {A,B,D} → {C},
{A,C,D} → {B}
```

The FD X → Y is non-trivial

Superkeys and Keys

Keys and Superkeys

A <u>superkey</u> is a set of attributes $A_1, \ldots, A_n$ s.t.
for *any other* attribute **B** in R,
we have $\{A_1, \ldots, A_n\} \rightarrow B$

I.e. all attributes are *functionally determined* by a superkey

A <u>key</u> is a *minimal* superkey

This means that no subset of a key is also a superkey (i.e., dropping any attribute from the key makes it no longer a superkey)

Finding Keys and Superkeys

- For each set of attributes X

1. Compute X^+
2. If X^+ = set of all attributes then X is a **superkey**
3. If X is minimal, then it is a **key**

Example of Finding Keys

```
Product(name, price, category, color)
```

```
{name, category} → price
{category} → color
```

What is a key?

Example of Keys

`Product(name, price, category, color)`

```
{name, category} → price
{category} → color
```

`{name, category}`$^+$ = `{name, price, category, color}`

= the set of all attributes

⇒ this is a **superkey**

⇒ this is a **key**, since neither `name` nor `category` alone is a superkey

Activity-11-1.ipynb