Functional Dependencies

Consider New-Accounts (Cust-name, Acct#, bal, br-name)

- each tuple specifies account details together with the name of the owner.

- it is possible to have multiple owners for an accounts

ex.

Cust-name	Acct #	bal	br-name
John	100	10k	UIC
Jeff	200	15k	LaSalle
Mary	100	10k	UIC

John, Mary own acct# 100.

Details of the account should be correctly replicated in both tuples.

Equivalently: Require that whenever two tuples have identical "account numbers" then the 'bal' and 'br-name' should be same.

This is specified by a functional dependency from Acct# to bal, br-name

Written as

Acct # $\rightarrow$ bal, br-name

$r(R)$ $\quad \alpha, \beta \subseteq R$

$\alpha \rightarrow \beta$

$A \rightarrow C$ ✓

$B \rightarrow C$ ✗

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_1	c_2	d_1
a_3	b_2	c_2	d_4
a_1	b_1	c_1	d_0

$\alpha \rightarrow \beta$ is satisfied by a table

iff (for all pairs of tuples t_1, t_2 in the table

if $t_1[\alpha] = t_2[\alpha]$ then $t_1[\beta] = t_2[\beta]$)

$r(R)$

has super key α

$\alpha \rightarrow R$

Student (St-id, C-id, grade, GPA)

(1, CS480, A, 3.5)

(1, CS580, B, 3.5)

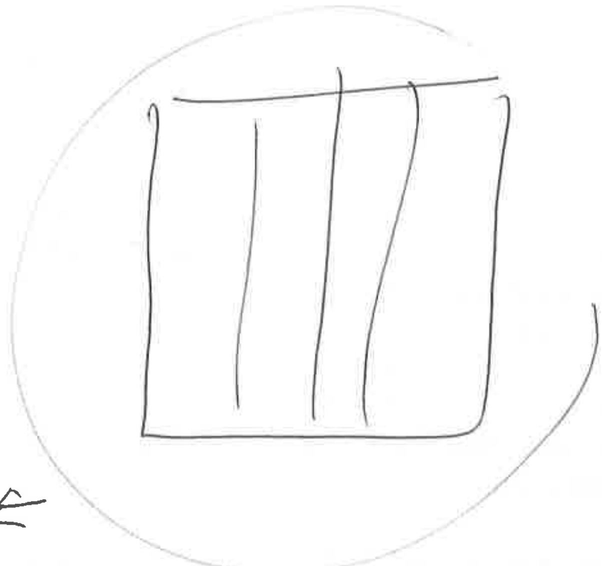
$\underbrace{\text{St-id}} \rightarrow \underbrace{\text{GPA}}$

St-id, $\underbrace{\text{C-id}}$ → grade

St-id	C-id	grade	GPA
1			
1			
1			

$r(A, B, C, D)$

$F = \{A \to B, \; B \to C\}$



$A \to C$

$F^+ = \{A \to B, \; B \to C, \; A \to C, \; A \to A$

A set of fds F
$r(R)$

$\{\alpha_1 \to \beta_1, \; \alpha_2 \to \beta_2, \; \}$

Def: A fd $\alpha_1 \to \beta_1$ is said to be implied by a set of fds F,

if whenever all the fds in F are satisfied by a table then $\alpha_1 \to \beta_1$ is also satisfied by that table

$F^+ = \{\alpha_1 \to \beta_1 \mid \alpha_1 \to \beta_1 \text{ is implied by } F\}$

Closure of F

$F \subseteq F^+$

Armstrong axioms:

A set of inference rules for deducing implied fds.

1. If $\beta \subseteq \alpha$ Then $\alpha \to \beta$ (Reflexivity)

2. If $\alpha \to \beta$ and $\beta \to \gamma$ Then $\alpha \to \gamma$ (Transitivity)

3. If $\alpha \to \beta$ Then $\alpha\gamma \to \beta\gamma$ (augmentation)

These rules are 'sound'

These rules are 'complete'

Derived rules:

4. If $\alpha_1 \to \beta_1$, $\alpha_1 \to \beta_2$ then $\alpha_1 \to \beta_1\beta_2$ (Union rule)

Assume $\alpha_1 \to \beta_1$, $\alpha_1 \to \beta_2$

$\alpha_1 \to \alpha_1\beta_2$

$\alpha_1\beta_2 \to \beta_1\beta_2$

$\alpha_1 \to \beta_1\beta_2$

5.

5. If $\alpha \to \beta\gamma$ then $\alpha \to \beta$ and $\alpha \to \gamma$ (decomposition)

easy. assume $\alpha \to \beta\gamma$

we have $\beta\gamma \to \beta$ (reflexivity rule)

Using transitivity, we get $\alpha \to \beta$.

Similarly we can get $\alpha \to \gamma$.

6. if $\alpha \to \beta$, $\beta\gamma \to \delta$ then $\alpha\gamma \to \delta$

Assume $\alpha \to \beta$ and $\beta\gamma \to \delta$

Using augmentation, we get $\alpha\gamma \to \beta\gamma$

Using transitivity we get $\alpha\gamma \to \delta$

Ex: $R = (A, B, C, G, H, I)$

$$F = \{ A \to B, \; A \to C, \; CG \to H, \; CG \to I, \; B \to H \}$$

Some fds in F^+:

$A \to BC$

$A \to H$

$CG \to HI$

$AG \to HI$

Given F.

Computing F^+:

Set Result := F

while a new fd $\alpha \to \beta$ can be deduced from Result, using rules 1, 2, 3

Result := Result $\cup \{\alpha \to \beta\}$

Closure of a set of attributes α:

Given a relation schema R, $\alpha \subseteq R$ and a set of fds F involving attributes in R,

define: $\alpha^+ = \{ A \in R \mid \alpha \to A \text{ is in } F^+ \}$

α^+ is the set of attributes that are functionally determined by α.

Ex: Student = (S_id, major, dept, College, acct #)

F = { S_id → major
major → dept
dept → college }

$(S\_id)^+$? are all the attributes functionally determined by S_id.

Significance of α^+:

α is a Super key of R iff $\alpha^+ = R$.

Algorithm to compute α^+:

result := α

while ($\exists$ an fd $\beta \to \gamma$ in F Such that $\beta \subseteq$ result)
$\gamma -$ result $\neq \phi$ and

add γ to result (i.e., result := result $\cup\ \gamma$)

'Student' example:

$(S\_id)^+$?

result := S_id
add 'major' to result
add 'dept' "
add 'college' "

Second example:

$R = (A, B, C, G, H, I)$

$F = \{ A \to B,\ A \to C,\ CG \to H,\ CG \to I,\ B \to H \}$

$(AG)^+$?

result := AG
add B
add C
add H
add I

$AG^+ = R$

Using the above approach,
we can determine if α is a Super Key.

Finding all the Candidate keys

Input : R, F

output : all the Candidate keys.

Recursive procedure keys(α) - outputs all Candidate keys in α.

```
Keys (α)
{     Boolean Change;
      change := False
      For each  A ∈ α
            if  (α - A)⁺ = R
                { chang := True;
                  keys ( α - {A}) }

      If  ¬ Change
             print (α);
}
```

To find all candidate keys: Keys (R)

Popular exam question:

Given R and F, find all candidate keys and justify your answer.

Running the above recursive procedure will take lot of time.

Instead use some simple tricks:

1. First identify all attributes that do not appear on the right hand side of a fd in F.

   All these attributes should be present in every candidate key.

2. Once you find one key, it is easier to find others.

Ex: $R = (A, B, C, G, H, I)$

$F = \{ A \to B,\ A \to C,\ CG \to H,\ CG \to I,\ B \to H \}$

'A', 'G' should be present in every candidate key. ~~G by itself is not a candidate key and hence $G^+ = G$~~

Try: AG

$(AG)^+ = ABCGHI$.

Hence AG is a candidate key. It is the only one.

Ex 2: $R = (A\ B\ C\ D\ E)$

$$F = \{ A \to BC$$
$$CD \to E$$
$$B \to D$$
$$E \to A \}$$

Every attribute appears on the right side of some fd.

Try A:

$A^+ = ABCDE$

Hence A is a candidate key

E is also a candidate key

CD is also a candidate key

BC is also a candidate key

These are the only ones.