User interaction

SQL> accept x number prompt 'Enter a#'

SQL> start file2

or

SQL> @ file2

file2.sql

```
prompt Type age of students
accept x number prompt 'Age : '
var y number;
execute :y := &x;
select sname from student
where age = :y;
```

Table Normalization

Table normalization deals with the issue of designing high quality tables.

Example

teachers (id, name, dob, yoe, h_degree, Salary)

Functional dependency

{yoe, h_degree} → {Salary}

{id} → {name, dob, yoe, h_degree, Salary}

Yoe	B	M	D
0	35000	37500	40000
1			
2			
3			
4			
5			80000

teachers

id	name	dob	yoe	h-degree	Salary
12	John	040890	2	BA	38000
34	peter	040870	2	BA	38000
36	Rose	110173	2	BA	38000
57	Pat	091280	10	MA	47000

yoe	h_degre	salary
0	BA	35000
1	BA	36000
2	BA	38000
⋮	⋮	⋮
10	MA	47000

id	name	dob	yoe	h_degre
12	John	040890	2	BA
34	Peter	040870	2	BA
36	Rose	110173	2	BA
57	Pat	091280	10	MA

Possible problems (issues) with bad (design) tables: redundancy, insertion anomalies, deletion anomalies, inconsistencies.

Functional Dependency

Definition: Let $R(A1, \ldots, An)$ be a relation schema. Let X and Y be two subsets of $\{A1, \ldots, An\}$. X is said to functionally determine Y (or Y is functionally dependent on X) if for every legal relation instance $r(R)$ any two tuples $t1$ and $t2$, when

$$t1[X] = t2[X] \Rightarrow t1[Y] = t2[Y]$$

Example:

R

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

	yes/no?
$A \to B$	no
$A \to C$	yes
$C \to A$	no
$A \to D$	no
$B \to D$	no
$AB \to D$	yes

$A \to B$

$\{A\} \to \{B\}$

$\{A, B\} \to \{D\}$

Identification of FDs

Trivial FDs if $Y \subseteq X \subseteq R$
then $X \to Y$

Example $R(A, B, C, D)$
$X = \{A, B, C\}$
$Y = \{B, C\}$

$ABC \to BC$
$A \to A$
$ABC \to C$
$\{A, B, C\} \to \{C\}$

Created by assertions

Teacher (id, name, ..., salary)
{yoe, h_degree} → {salary}

Analyze the semantics

Address(city, street, zipcode)
zipcode → city ✓
city, street → zipcode ?

Derive New FDs from existing FDs

$R(A, B, C)$ $F = \{A \to B, B \to C\}$
$A \to C$ can be derived from F

$F \models A \to C$ $\models$ logically implies

Definition: Let F be a set of FDs in R. The closure of F is the set of all FDs that are logically implied by F.

$$F^+ = \{ X \to Y \mid F \models X \to Y \}$$

F^+ may be derived from F.

Armstrong's Axioms

(R1) Reflexivity Rule: if $X \supseteq Y$, then $X \to Y$

(R2) Augmentation rule: $\{X \to Y\} \models XZ \to YZ$

(R3) Transitivity Rule: $\{X \to Y, Y \to Z\} \models X \to Z$

Armstrong's Axioms are sound and complete.

(R4) Decomposition rule $\{X \to YZ\} \models \{X \to Y, X \to Z\}$

proof

(1) $X \to YZ$ given

(2) $YZ \to Y$ by R1

(3) $YZ \to Z$ by R1

(4) $X \to Y$ ✓ by (1) and (2) with R3

(5) $X \to Z$ ✓ by (1) and (3) with R3

(R5) union rule

$\{X \to Y, X \to Z\} \models \{X \to YZ\}$

proof

(1) $X \to Y$ given

(2) $X \to Z$ given

(3) $XZ \to YZ$ by R2 on (1)

(4) $XY \to YZ$ by R2 on (2)

(5) $XX \to XY$ $X \to XY$ by R2 on (1)

(6) $X \to YZ$ ✓ by R3 on (5) and (4)

(R6) Pseudotransitivity rule

$\{X \to Y, WY \to Z\} \models WX \to Z$

proof

(1) $X \to Y$ given

(2) $WY \to Z$ given

(3) $WX \to WY$ by R2 on (1)

(4) $WX \to Z$ by R3 on (2) and (3)

<u>Exercise</u>

use Armstrong's Axioms to show that $\{A \to C, B \to D, CD \to E\}$

$\models AB \to E$

proof

solution given a student

Closure of Attributes

How to determine if $F \models X \to Y$ is true?

Method 1: compute F^+

if $X \to Y \in F^+$ then $F \models X \to Y$; else $F \not\models X \to Y$

Problem: computing F^+ could be very expensive

Method 2: compute X^+ the closure of X under F

X^+ denotes the set of attributes that are functionally determined by X under F

$X^+ = \{ A \mid X \to A \in F^+ \}$

Algorithm for computing X^+

Input: a set of FDs F, a set of attributes X in R

output: X^+

begin $X^+ := X$;

repeat for each FD $Y \rightarrow Z$ in F

do if $Y \subseteq X+$ then

$X+ := X+ \cup Z$

until no change to $X+$

end;

<u>Example</u>: $R(A, B, C, G, H, I)$

$F = \{ A \rightarrow B, CG \rightarrow HI, B \rightarrow H, A \rightarrow C \}$

compute $(AG)+$

$(AG)+ = AGBHCI$

AG is a candidate key

$A+ = ABHC$

$G+ = G$