

CS3010301 Database Systems

Spring 2013

Mid-term_solution

[1]

1. (12%) Explain the following terms briefly. Use an example if necessary.

- (3%) Entity Set.

Each entity type will have a collection of entities stored in the database

- (3%) Hierarchical Data Model.

HDM is a model that the data is organized into a tree-like structure.

- (3%) Referential Integrity Constraint.

Specified between two relations. maintains consistency among tuples in two relation.

- (3%) Union Compatible.

The two operand relation R and S must be type compatible, which have same number of attributes.

Each pair of corresponding attributes must have same or compatible domains.

(10%) Let $R(a, b, c)$ be a relation schema. Assume that the domains of the a , b , and c attributes are all integers. Let r_1 and r_2 be two instances of R . For each equation below, justify if

[2]

(1) Always true.

對同一個 instance(r1)

> 以先後順序來看：

不論先做 $\text{Select}(\sigma)$ 或 $\text{Project}(\pi)$ 不影響取出來的結果> 以 $\text{Select}(\sigma)$ 的條件來看： $c=a$ 和 $a=c$ 不影響取出結果> 以 $\text{Project}(\pi)$ 來看雖然 $\text{Project}(\pi)$ 出來的順序分別為 a, c 和 c, a

但 Fcount 是計算 tuple 數量

因此此式永遠為真

(2) Not always true.

> Ex.

r1			r2		
a	B	c	a	b	c
1	2	3	1	2	3
2	4	2	2	3	4
1	1	3	3	3	1

> 運算結果：

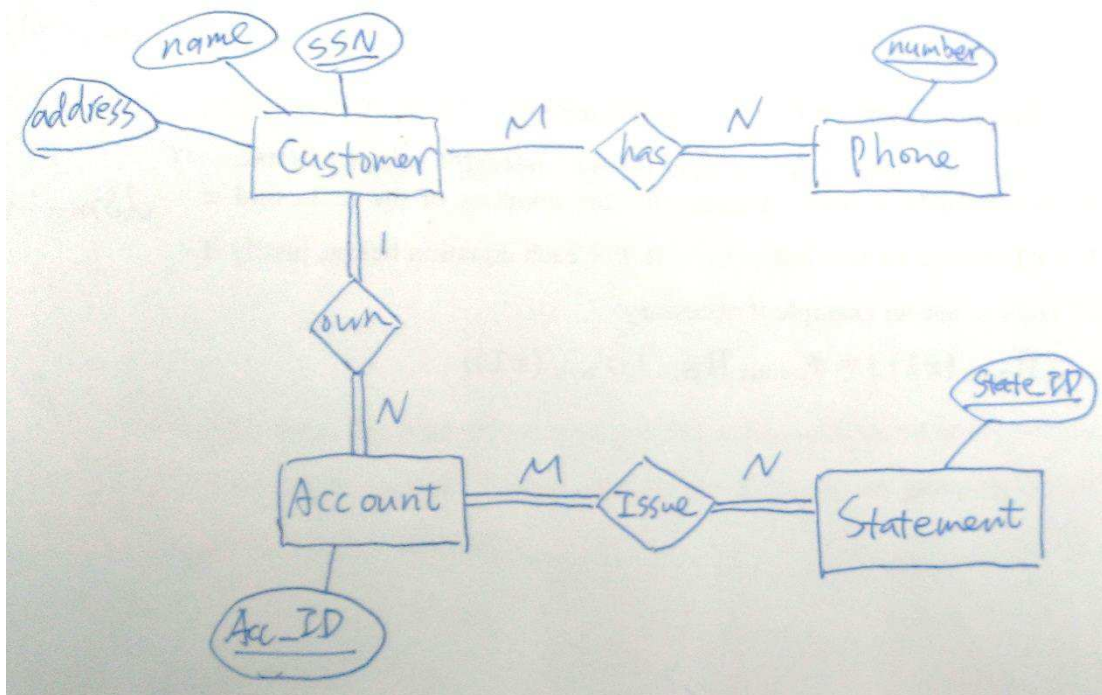
左式		右式	
c	A	a	c
2	2	2	2
		1	3

> Fcount 結果：

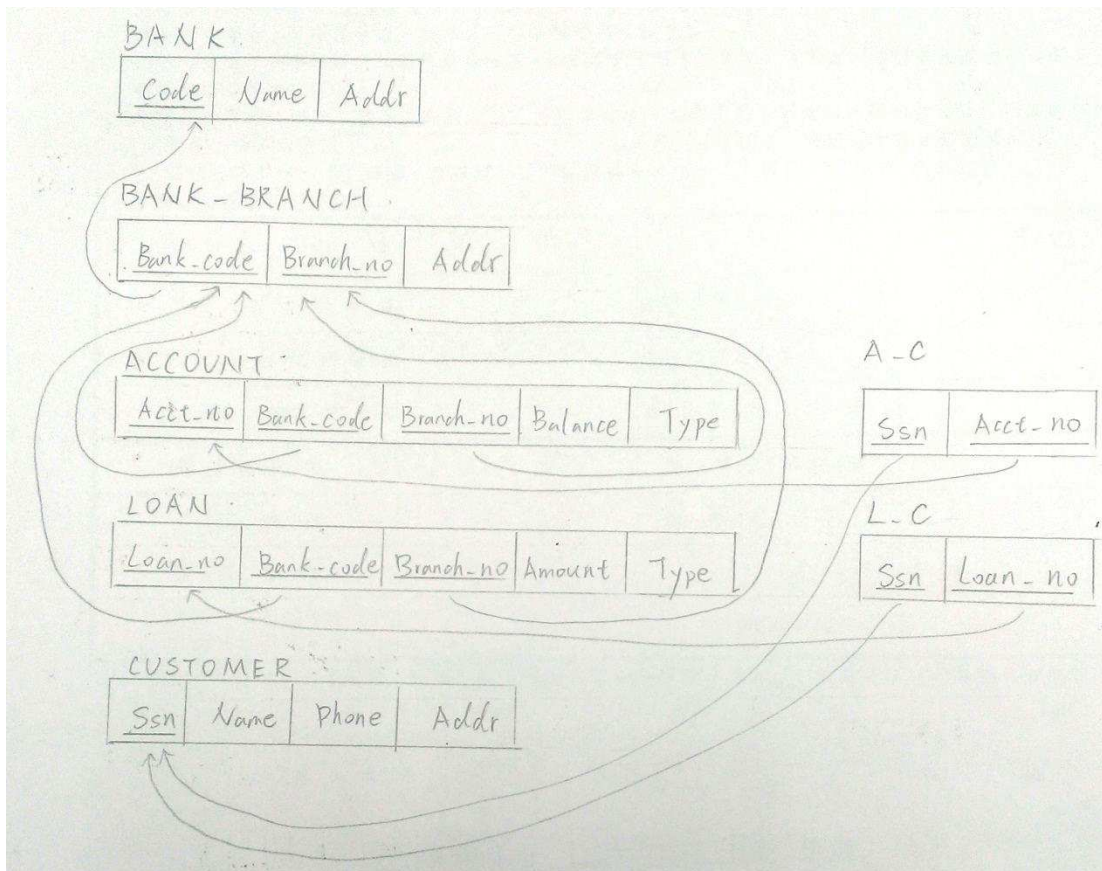
左式：1 \neq 右式：2

因此此式不會永遠為真

[3]



[4]



[5]

a,b,c

a. (5%) Write the query "Find the names of students who received grades higher than 90 in a course given by Tomas Huang" in **domain** relational calculus.

$$\{b \mid (\exists a)(\exists c)(\exists d)(\text{students}(abcd)) \text{ AND } (\exists e)(\exists f)(\exists g)(\text{registered}(efg)) \text{ AND } e=a \text{ AND } g > 90 \text{ AND } (\exists h)(\exists i)(\exists j)(\text{courses}(hij)) \text{ AND } j = \text{'Tomas Huang'} \text{ AND } h=f \}$$

b. (5%) Write the query "Find the addresses of students whose name is Frank Wang or register in Computer Vision class." in **tuple** relational calculus.

$$\{x.\text{address} \mid \text{students}(x) \text{ AND } x.\text{name} = \text{'Frank Wang'} \} \text{ OR } (\exists y)(\text{courses}(y) \text{ AND } y.\text{name} = \text{'Computer vision class'} \text{ AND } (\exists z)(\text{registered}(z) \text{ AND } z.\text{cid} = y.\text{cid} \text{ AND } z.\text{sid} = x.\text{sid})) \}$$

c. (5%) Write the query "Find the sids of students who have taken CS101 and CS301 but no other courses" in relational algebra. (CS101 and CS301 are "cid"s)

$$\begin{aligned} \text{CS101} &= \pi_{\text{sid}} (\sigma_{\text{cid} = \text{CS101}} (\text{registered})) & \text{Result}(\text{sid}) &= \text{Both} - \text{Other} \\ \text{CS301} &= \pi_{\text{sid}} (\sigma_{\text{cid} = \text{CS301}} (\text{registered})) \\ \text{Both}(\text{sid}) &= \text{CS101} \cap \text{CS301} \\ \text{Other} &= \pi_{\text{sid}} (\sigma_{\text{cid} \neq \text{CS101} \text{ OR } \text{cid} \neq \text{CS301}} (\text{registered})) \end{aligned}$$

d. (5%) Write the query "Find the sids of students who have taken all the coursed given by

d,e

d. (5%) Write the query "Find the sids of students who have taken all the coursed given by Tomas Huang" in **tuple** relational calculus.

$$\{x.\text{sid} \mid \text{registered}(x) \text{ AND } (\forall c)(\text{courses}(c) \text{ AND } c.\text{profname} = \text{'Tomas Huang'} \text{ AND } c.\text{cid} = x.\text{cid}) \}$$

e. (5%) Write the query "For each student, retrieve the name of the student, and the average grade of the student" in relational algebra. You may use grouping and/or aggregate functions for the query, if necessary.

$$\begin{aligned} \text{Grade} &= \text{sid} \int \text{Avg grade} (\text{registered}) \\ \text{Result} &= \pi_{\text{name}, \text{avgGrade}} (\text{Grade} \bowtie \text{students}) \end{aligned}$$

f

f. (5%) Write the query "Find pairs of student ids such that the student with the first sid got higher grade in some classes than the student with the second sid" in relational algebra

$\text{registered}_2 = \rho(\text{sid}_2, \text{cid}_2, \text{grade}_2) \text{ registered}$

$\text{All} = \text{registered} \times \text{registered}_2$

$\text{Result} = \pi_{\text{sid}_1, \text{sid}_2} (\sigma_{\text{sid}_1 \neq \text{sid}_2 \text{ AND } \text{cid}_1 = \text{cid}_2 \text{ AND } \text{grade}_1 > \text{grade}_2 (\text{All}))$

Page 4 of 4