

期中考參考解答及評分標準

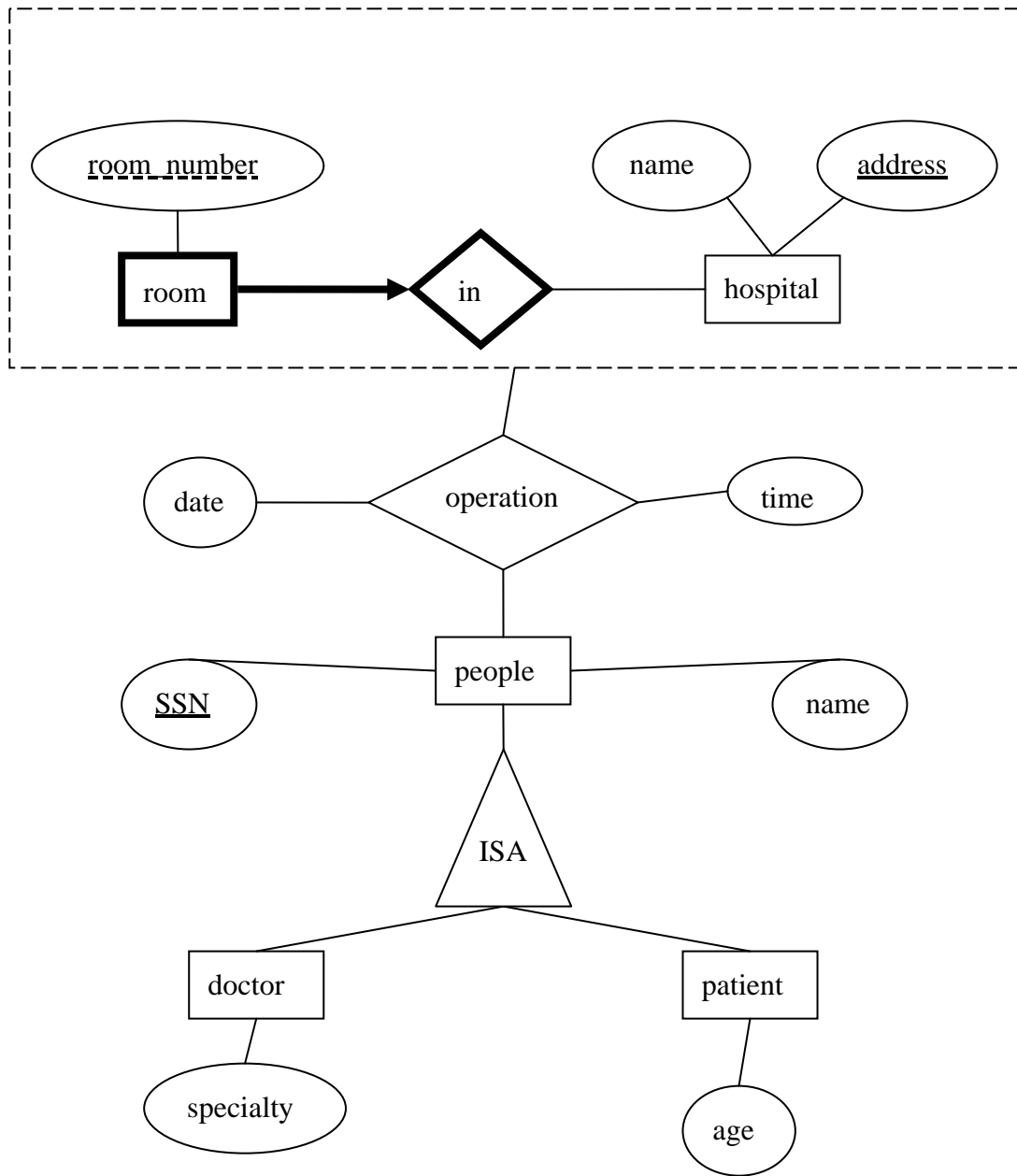
PART	負責的助教
A	許主聰
B	江建興
C	江建興
D	曾柏儂
E	江建興

如果同學們對哪些部分有任何問題，請在 office hour 時去找助教。

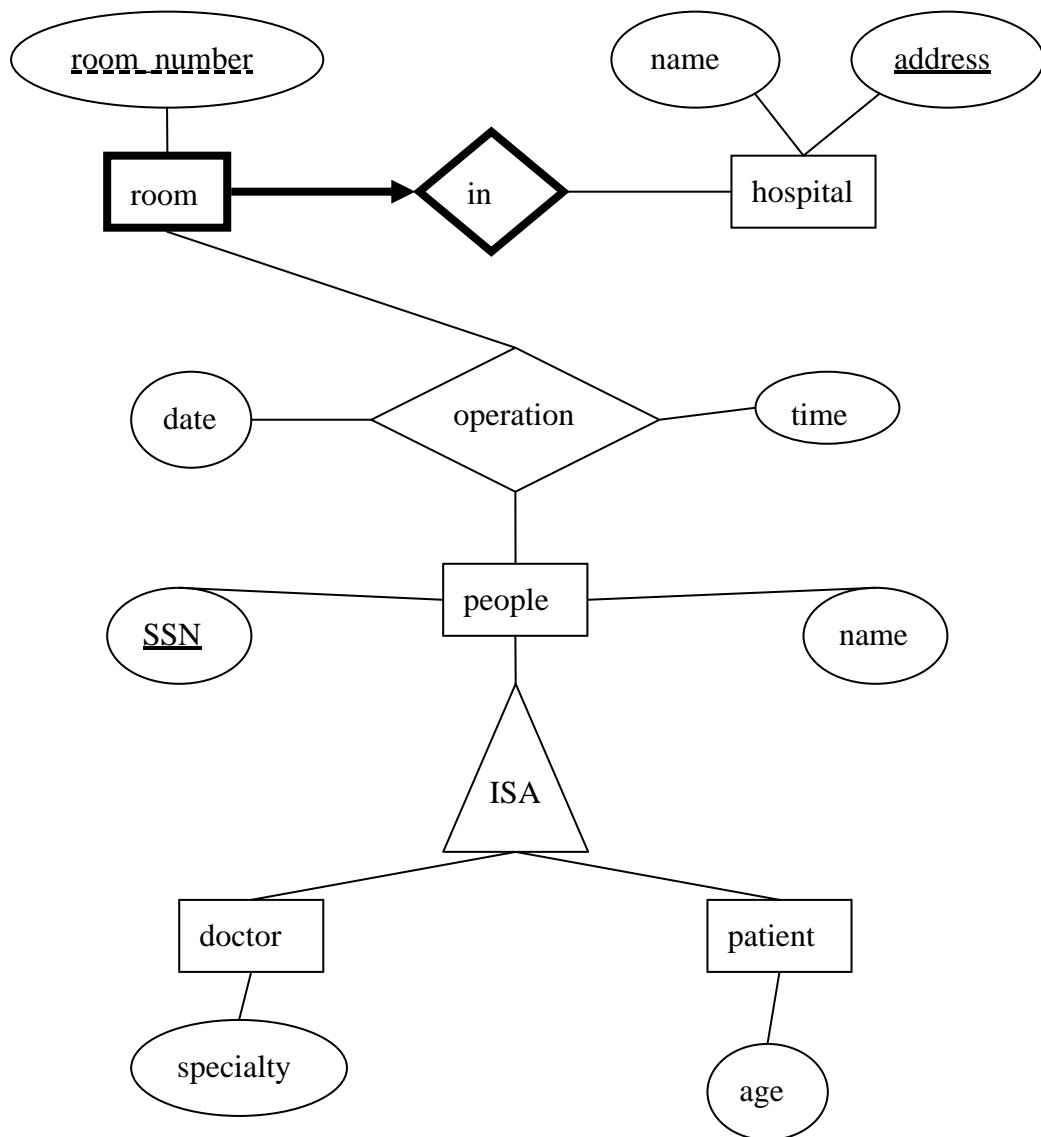
**PART A: ER Diagram(24points)**

Ans :

(1)畫圖部分(只提供 2 種版本) :



or



評分標準：

4 個 entities: room, hospital, doctor, patient 各個 2 分

2 個 relationship: in, operation 各 2 分

正確畫出 room 是 weak entity 得 2 分

對每個小錯誤(如 weak entity 的 partial key 沒有用虛線表示)，扣微量分數(以 0.5 分為基本單位)

對於題目中沒有提到的 attributes，如果在不必新增的情況下自己新增一些 attributes，會視新增多少個 attributes 以及它的必要性來選擇扣分

(2)SQL 部分

```
1.CREATE TABLE hospital (  name CHAR(50),
                           address CHAR(50),
                           PRIMARY KEY(address)
)
2.CREATE TABLE room_in (  room_number INTEGER,
                           address CHAR(50),
                           PRIMARY KEY(room_number,address),
                           FOREIGN KEY(address) REFERENCES hospital
                           ON DELETE CASCADE
)
3.CREATE TABLE people (  SSN CHAR(50),
                           name CHAR(50),
                           PRIMARY KEY(SSN)
)
4.CREATE TABLE doctor (  SSN CHAR(50),
                           specialty CHAR(50),
                           PRIMARY KEY(SSN),
                           FOREIGN KEY(SSN) REFERENCE people(SSN)
                           ON DELETE CASCADE
)
5.CREATE TABLE patient (  SSN CHAR(50),
                           age INTEGER,
                           PRIMARY KEY(SSN),
                           FOREIGN KEY(SSN) REFERENCE people(SSN)
                           ON DELETE CASCADE
)
/*
```

也接受把 3,4,5 寫成：

```
CREATE TABLE doctor (  SSN CHAR(50),
                           name CHAR(50),
                           specialty CHAR(50),
                           PRIMARY KEY(SSN)
)
CREATE TABLE patient (  SSN CHAR(50),
                           name CHAR(50),
                           age INTEGER,
                           PRIMARY KEY(SSN)
```

```
)  
*/
```

```
6.CREATE operation ( doctor_SSN CHAR(50),  
patient_SSN CHAR(50),  
address CHAR(50),  
room_number INTEGER,  
time TIME,  
date DATE,  
PRIMARY KEY(doctor_SSN, patient_SSN,  
address, room_number),
```

```
/*
```

也接受把 time 和 date 也當成 key 的一部分，即：

```
PRIMARY KEY(doctor_SSN, patient_SSN,  
address, room_number, time , date),
```

```
*/
```

```
FOREIGN KEY (doctor_SSN) REFERENCE doctor(SSN),  
FOREIGN KEY (patient_SSN) REFERENCE patient(SSN),  
FOREIGN KEY(room_number,address) REFERENCE  
room_in(room_number,address)
```

```
)
```

評分標準：

依(1)建立 tables，把 12 分平均分配

對每個小錯誤(如在建 weak entity 的 table 時，它的 foreign key 部分沒有加 ON DELETE CASCADE)，扣微量分數(以 0.5 分為基本單位)

因為 part 2 會參考到 part 1，如果 part1 中有畫錯的，轉成 SQL 時有可能會出錯，因此視情況扣分

**PART B: Relational Algebra(20 points)**

(1)

Ans :

idNumber	Type	cageNumber	name	
1	Zebra	10	Ann	
3	Monkey	10	Bob	
4	Kangaroo	30	Chris	
6	Monkey	40	Bob	

評分標準：搭配(2)，如果(2)表達正確，則(1)每少寫一個 row 扣一分

(2)

Ans :

**List each animal with people who are both its cageKeeper and typeKeeper.**

評分標準：有寫到 cageKeeper and typeKeeper is the same person 就給六分，如果方向對但表達不清楚則給一半分數

(3)

Ans :

$p(A1, Animals)$

$p(A2, Animals)$

$\pi_{A1.idNumber, A1.type, A2.idNumber, A2.type, A2.cageNumber} (\sigma_{A1.cageNumber=A2.cageNumber \text{ AND } A1.type \neq A2.type \text{ AND } A1.idNumber > A2.idNumber} (A1 \times A2))$

評分標準：

有三個條件需要考慮，分別是：

- 1.two different types
- 2.in the same cage
- 3.same animal pair should not appear twice

條件 1、2 少寫一個扣兩分，條件 3 沒寫或寫 " ≠ " 扣 3 分。

(A1 x A2)寫成(A1∞A2)扣兩分(這邊應該是用 equi join 給條件，natural join 的話會比對所有 fields，所以會得到原本的 Animals table)。

不強調語法部份，意思表達正確即可，除非太明顯錯誤或缺少才酌量扣分。

PART C: Query Equivalences (27 points)

Ans :

	Query 1	Query 2	Equivalent
1	$\pi_A(R \cup S)$	$\pi_A(R) \cup \pi_A(S)$	Yes
2	$\pi_A(R - S)$	$\pi_A(R) - \pi_A(S)$	No
3	$\sigma_{R.A=5}(R)$	$\sigma_{R.A=5}(\pi_{R.A,R,B}(R \infty S))$	No
4	$\pi_{R.A}(\sigma_{R.A=S.A}(R \times S))$	<i>select R.A from R,S where R.A=S.A</i>	Yes
5	$\pi_{R.B}(\sigma_{R.A=S.A}(R \times S))$	<i>select R.B from R,S where R.A=S.A</i>	No
6	$\pi_A(R) - \pi_{R1.A}(\sigma_{R1.A < R2.A}(\rho_{R1}(R) \times \rho_{R2}(R)))$	<i>select min(A) from R</i>	Yes
7	$\pi_A(R) - \pi_{R1.A}(\sigma_{R1.A < R2.A}(\rho_{R1}(R) \times \rho_{R2}(R)))$	<i>select max(A) from R</i>	Yes
8	<i>select B from R group by B</i>	<i>select distinct B from R</i>	Yes
9	$R/\pi_A(S)$	$\pi_A(R) - \pi_A((\pi_A(R) \times \pi_A(S)) - R)$	No

說明：

2. NO

eg.  $R\{(A,B)\} = \{(5,6)\}$  ,  $S\{(A,B)\} = \{(5,7)\}$

Query1{(A)} = {(5)} , Query2{(A)}={}

3. NO

eg.  $R\{(A,B)\} = \{(5,6)\}$   $S\{(A,B)\} = \{(5,7)\}$

Query1{(A,B)} = {(5,6)} , Query2{(A,B)}={}

5. NO

在討論 relation algebra 時，預設是會做 duplicate elimination，

而在實際的系統中預設是不會做 duplicate elimination

因為 B 不是 key，有可能會重複

eg.  $R\{(A,B)\} = \{(5,6), (6,6)\}$   $S\{(A,B)\} = \{(5,6), (6,6)\}$

Query1{(B)} = {(6)}, Query2{(B)} = {(6),(6)}

9. NO

正確的是  $R \div \pi_A(S) = \pi_B(R) - \pi_B(\pi_A(S))$  X  $\pi_B(R) - R$

評分標準：寫錯扣四分，沒寫扣三分



**PART D: SQL (16 Points)**

Ans :

(1)

```
SELECT DISTINCT v.res
FROM Votes v
WHERE (SELECT COUNT(*) FROM Votes v2 WHERE v.res=v2.res and
v2.vote='yes')>=9
AND NOT EXISTS (SELECT *FROM Members m, Votes v3
WHERE v3.country-name=m.country-name
AND m.status='permanent' AND v3.vote='no' AND v3.res=v.res)
```

Or

```
SELECT V.res FROM Votes V WHERE V.vote='yes' AND V.res NOT IN
(SELECT V1.res FROM Votes V1, Members M1 WHERE V1.vote='no' AND
M1.status='permanent'
AND V1.country-name =M1.country-name) GROUP BY V.res HAVING COUNT(*)>=9;
```

Or

```
SELECT DISTINCT v.res FROM Votes v
WHERE (SELECT COUNT(*) FROM Votes v1
WHERE v.res=v1.res and v1.vote='yes')>=9 AND
(SELECT COUNT(*) FROM Votes v2, Members m
WHERE v.res=v2.res AND v2.country-name=m.country-name
AND m.status='permanent' and v2.vote='no')=0
```

Or

```
SELECT DISTINCT v.res
FROM Members m, Votes v
WHERE (SELECT Count(*) FROM Votes v1
WHERE v.res=v1.res and v1.vote='yes')>=9 AND v.res NOT IN
(SELECT v3.res FROM Members m3, Votes v3
WHERE m3.country-name=v3.country-name
AND m3.status='permanent' AND v3.vote='no');
```

評分標準：

For a resolution to pass, you must consider two conditions:

1. At least 9 'yes' votes for the resolution.
2. None of five permanent members vote 'no' on the resolution. (i.e. If any permanent member votes 'no' on the resolution, the resolution won't be passed.)

If you consider both conditions and write query correctly, you surely get 8 points.

If you consider one of the two conditions, you will get 6 points.

With any SQL misuse or incorrect query, you will lose some points.

(2)

Ans :

```
SELECT M1.country-name, COUNT(*) AS vetos
FROM Members M1, Votes V1
WHERE M1.status='permanent'
AND M1.country-name =V1.country-name AND V1.vote='no'
GROUP BY M1.country-name
UNION
SELECT M2.country-name, 0 AS vetos
FROM Members M2
WHERE M2.status='permanent' AND M2.country-name not in
(SELECT V2. country-name
FROM Votes V2, Members M3
WHERE V2.vote='no' AND M3.country-name=V2.country-name);
```

Or

A better query:

```
SELECT M.country-name, (SELECT COUNT(*) FROM Votes V
WHERE V. country-name =M. country-name AND V.vote='no') as vetos
FROM Members M WHERE M.status='permanent'
```

評分標準：

For list each permanent member with number of vetos, you must consider two conditions:

1. List the count of 'no' votes of permanent members.
2. List permanent members who have not cast any 'no' votes with '0' in column 'vetos' .

If you consider both conditions and write query correctly, you surely get 8 points.

If you consider one of the two conditions, you will get 6 points.

With any SQL misuse or incorrect query, you will lose some points.

PART E: More SQL(16points)

(1)

Ans :

Count the number of unique tuples with A=3.

評分標準：意思對就六分，答非所問不給分(ex: 將 query 翻成白話之類的...)

(2)

Ans :

Attribute A is a unique identifier for table T.

Another answer :

For each value  $a'$  belong to T.A, all tuple( $a'$ ,  $b'$ ,  $c'$ ) are also in T,  $b' \in B'$  and  $c' \in C'$

where  $B' = \{\text{all } b' \text{ value belong to T.B under the situation that T.A = "a'"}\}$

$C' = \{\text{all } c' \text{ value belong to T.C under the situation that T.A = "a'"}\}$

In other word, for tuples (a,b,c) and (a,b',c'), the tuples (a,b,c') and (a,b',c) are also in T.

評分標準：只寫兩個答案其中之一就可得到全部分數，但多寫的如果不對就扣兩分(ex: A and C are unique)，另外，寫第二種答案如果沒有強調粗體字部份扣兩分，其他答案或答非所問則不給分(ex: if two tuples have same A, they will have a same C.))