

## 【資訊處理】

## 《資料庫應用》

## 試題評析

今年度考題份量較重，需花費較多時間解題。題型並不出傳統考題範圍，在傳統考題上作一些較為複雜的變化，考驗考生們對變化題的思考能力。一般程度考生約可得60至70分，程度較佳同學可在75分以上。

一、假設某書商之關連資料庫 (Relational Database) 中有一個表格 (Table) 儲存了「書籍」(Book) 之相關資訊。其Schema如下：

Book (Title, Type, Price, Name, Address, Publisher)

上述表格中之屬性分別代表「書籍名稱」、「書籍種類」、「書籍訂價」、「作者姓名」、「作者地址」、「出版公司」。

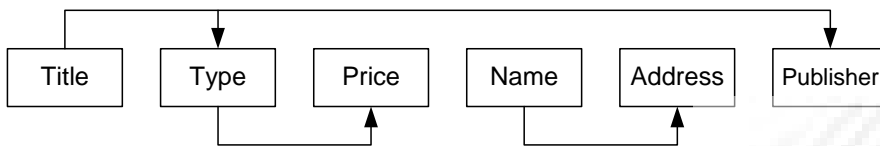
其功能依賴 (Functional Dependencies) 為 Title  $\rightarrow$  {Publisher, Type}, Type  $\rightarrow$  Price, Name  $\rightarrow$  Address

- (一) 請找出Book所有的候選鍵 (Candidate Keys)。注意：你必須以功能依賴的推論 (Inferences) 規則來證明你找出的是Candidate Keys，否則視為只憑猜測，不予計分。(5分)
- (二) 請以功能依賴的推論證明Book不是在2NF。(5分)
- (三) 請先將Book分解為滿足2NF、但可能不滿足3NF的表格Schema，你必須以功能依賴的推論證明每一個你的表格是在2NF。對每一個你的表格主鍵 (Primary Key)，請以底線標示。請注意：本小題只先要求到2NF，所以，為求資料庫的表格數精簡，你不應作不必要的分解，否則視為錯誤。(9分)
- (四) 請再將上述(三)小題中你所得答案中滿足2NF、但可能不滿足3NF的表格進一步分解為滿足3NF。最後將完整的資料庫所有表格Schema全部列出。對每一個你的表格主鍵 (Primary Key)，請以底線標示。對每一個你的表格予以適當語意出名，並以功能依賴的推論證明是分別在3NF。(11分)
- (五) 若上述(四)小題不進一步分解，那麼你預期在資料庫的日後操作上會有那種異常狀況 (Anomalies)？請舉例說明。(5分)

答：

(一)

Book(Title, Type, Price, Name, Address, Publisher)中，  
Title $\rightarrow$ {Publisher, Type}, Type $\rightarrow$ Price, Name $\rightarrow$ Address。



- (1) Title $\rightarrow$ {Publisher, Type}可分解為Title $\rightarrow$ Publisher與Title $\rightarrow$ Type。  
其中，Title $\rightarrow$ Type，又Type $\rightarrow$ Price，則根據遞移性可知Title $\rightarrow$ Price。  
Title $\rightarrow$ {Publisher, Type}，且Title $\rightarrow$ Price，則Title $\rightarrow$ {Publisher, Type, Price}。
- (2) Title $\rightarrow$ {Publisher, Type, Price}，又Name $\rightarrow$ Address，則{Title, Name} $\rightarrow$ {Address, Publisher, Type, Price}。  
{Title, Name}可決定所有Book的非鍵屬性，故候選鍵為{Title, Name}。

(二)

一關聯為2NF，若且唯若所有非鍵屬性皆完全功能相依於主鍵。

Book中，{Publisher, Type, Price}部分功能相依於主鍵的Title屬性；

Address部分功能相依於主鍵的Name屬性；

故Book非2NF。

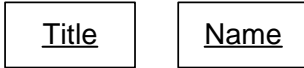
(三)

由於{Publisher, Type, Price}部分功能相依於主鍵的Title屬性，

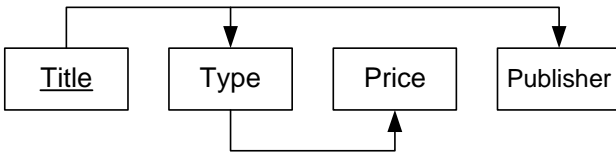
Address部分功能相依於主鍵的Name屬性，

故將Book分解為下面三關聯：

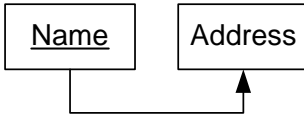
Book(Title, Name)，候選鍵為{Title, Name}。



BookTitle(Title, Type, Price, Publisher)，候選鍵為Title，所有屬性皆完全功能相依於Title，故BookTitle為2NF。



Author(Name, Address)，候選鍵為Name，所有屬性皆完全功能相依於Name，故Author為2NF。



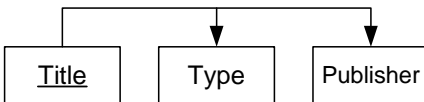
(四)

Author與Book關聯無任何遞移相依現象，故Author與Book為3NF。

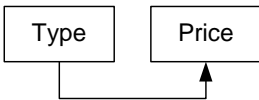
BookTitle中，Title→Type, Type→Price，則Price遞移相依於主鍵Title，故BookTitle非3NF。

BookTitle可分解為下面兩關聯：

BookTitle(Title, Type, Publisher)，所有屬性皆非遞移相依於主鍵，故BookTitle為3NF。



TypePrice(Type, Price)，所有屬性皆非遞移相依於主鍵，故TypePrice為3NF。



3NF後，各關聯整理如下：

Book(Title, Name)

BookTitle(Title, Type, Publisher)

TypePrice(Type, Price)

Author(Name, Address)

(五)由於Book關聯中的Title與Name屬性皆為其他關聯之主鍵，但該值在Book關聯中皆可重複，若更新資料時未考慮鍵值限制，將導致資料更新異常，如作者姓名更新時，需更新至各書籍之該作者資料。

二、假設某關連資料庫 (Relational Database) 中有兩個表格 (Tables) 分別儲存「員工」 (Employee)、「部門」 (Department)，其Schema如下：

屬性分別代表員工「編號」、「姓名」、「性別」、「出生年」、「專職否」、「薪水」、「所屬部門代號」，而以「編號」為其主鍵 (Primary Key)，沒有其他候選鍵 (Candidate

Key)。” Full” 的值為字元Y時表示專職，N時表示兼職。

Department (DNo, Name, Area)

屬性分別代表部門「代號」、「名稱」、「所在地區」，而以「代號」為其主鍵，沒有其他候選鍵。「名稱」值可能為「生產部」、「行銷部」等中文字串，「所在地區」值可能為「台北市」、「高雄市」等中文字串。

(一)請寫出下列查詢之SQL：(9分)

對所有專職人員，請依男女分別列出平均年齡，平均年齡欄位另取名為Average\_Year。

(二)請寫出下列查詢之SQL：(8分)

列出「生產部」的員工中最高與最低薪水的差距，差距欄位取名為Salary\_Difference。

(三)請以SQL建立一個觀點(View)，名為Work\_on\_Taipei，該View包括所有所屬部門在「台北市」的員工，欄位有「姓名」(另取名為ENAME)、「薪水」、「所屬部門名稱」(另取名為DName)。(8分)

(四)請問Work\_on\_Taipei這個View是否為Updatable? 為什麼?(5分)

答：

(一)

```
SELECT Sex, AVG(YEAR(GETDATE()) - BirthYear) AS Average_Year
FROM Employee
WHERE Full='Y'
GROUP BY Sex
```

(二)

```
SELECT MAX(Salary)-MIN(Salary) AS Salary_Difference
FROM Employee JOIN Department ON DeptNo=DNo
WHERE Department.Name='生產部'
```

(三)

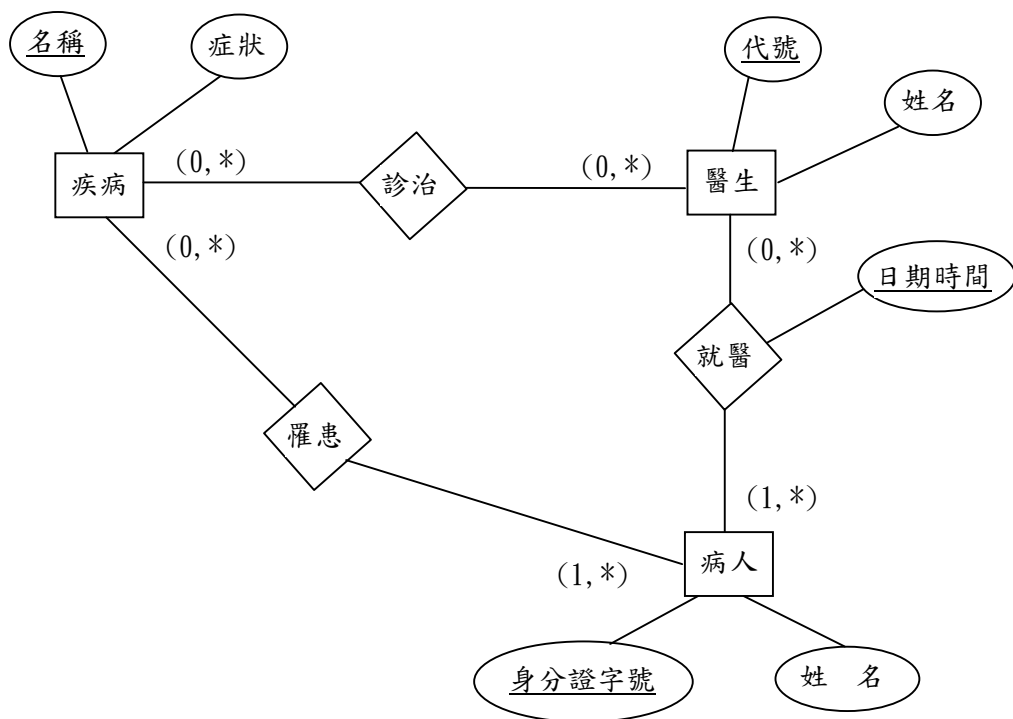
```
CREATE VIEW Work_on_Taipei AS
SELECT Employee.Name AS EName, Salary, Department.Name AS DName
FROM Employee JOIN Department ON DeptNo=DNo
WHERE Area='台北市'
```

(四)否，FROM中包含超過一個關聯，故此View不可更新。

三、圖一為實體關係圖(Entity-Relationship Diagram)，對所用符號的約略說明：其中每一實體(矩形)鄰近某關係(菱形)間均有一對(min, max)的限制，代表該實體參與該關係之min(至少)、max(至多)次數限制，\*代表沒有限制。屬性劃實底線者為Primary Key(主鍵)的一部分。

(一)請問該圖是否可抓住一般「某病人罹患某疾病由某醫生診治」的情境? 請舉例說明之。(5分)

(二)請以兩種方式，重繪實體關係圖。要求：①不得使用高於二元關係(Binary Relationship)之三元或多元關係。②所繪之圖應是最精簡的，沒有多餘、不必要的實體、關係、屬性。③兩種方式中，第一種方式只能用一般實體，而第二種方式必須採用弱實體(Weak Entity)(以雙矩形圖示)的觀念。(20分)



答：

(一)否，由於醫生與疾病間為多對多關係，無指定特定疾病由特定醫生診治。且病人與醫生亦為多對多，無法表示罹患特定疾病時，是由哪位醫生診治。

舉例如下：

病人A123456789曾罹患”喉嚨發炎”與”流行性感冒”兩疾病(病人與疾病多對多關係)；

醫生代號D001與D002皆可診治”喉嚨發炎”與”流行性感冒”兩疾病(醫生與疾病多對多關係)；

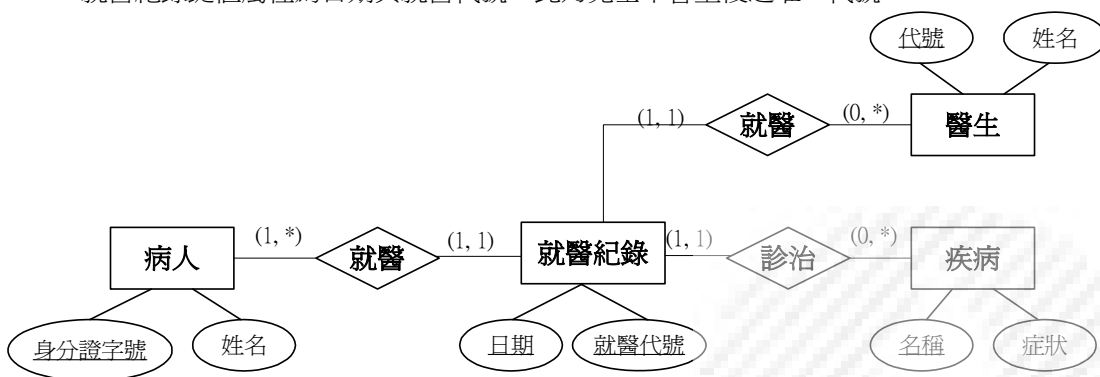
病人A123456789亦曾至醫生D001與D002就醫(醫生與病人多對多關係)。

故無法表示出病人A123456789罹患”喉嚨發炎”與”流行性感冒”時，分別由何位醫生診治。

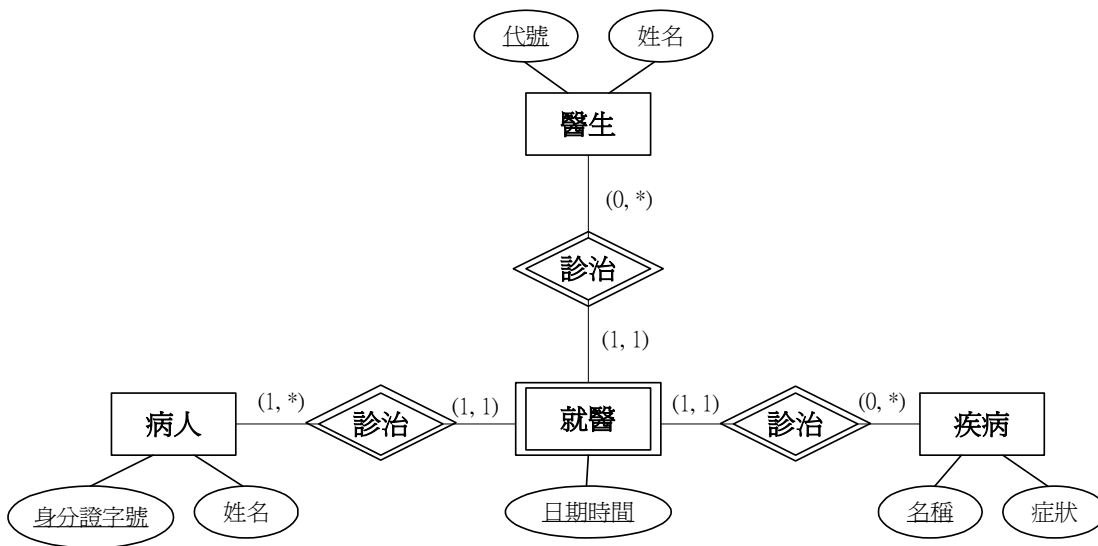
(二)

(1)以三個二元關係，加上就醫紀錄之一般個體。

就醫紀錄鍵值屬性為日期與就醫代號，此乃完全不會重複之唯一代號：



(2)以三個二元關係與弱個體表示：



四、請問假設某資料庫只有如表一日誌 (Log) 所顯示的時程 (Schedule) 執行，其中 [write\_item, T, X, old\_value, new\_value] 代表交易 (Transaction) T 對資料庫項目 X 下寫的指令，其寫之前 X 的值為 old\_value。其寫之後 X 的值為 new\_value。若 DBMS 是採立即更新技術 (Immediate Update Technique)，當回復 (Recovery) 後，各項目 (Item) A, B, C, D, E 值各為多少？(10 分)

表一

[start_transaction, T1]
[write_item, T1, A, 1, 2]
[write_item, T1, B, 0, 300]
[commit, T1]
[start_transaction, T2]
[write_item, T2, C, 100, 200]
[commit, T2]
[Start_transaction, T3]
[checkpoint]
[write_item, T3, D, 1000, 2600]
[Start_transaction, T4]
[Write_item, T4, A, 2, 22]
[write_item, T3, B, 300, 331]
[Start_transaction, T5]
[Write_item, T5, E, 12, 22]
[commit, T3]
[commit, T4]
[write_item, T5, B, 331, 336]
[write_item, T5, A, 22, 29]
[start_transaction, T6]
[write_item, T6, D, 2600, 2800]
System Crash

圖一

答：

下面表格描述進行各指令時間點，交易UNDO與REDO串列之狀態：

指令	UNDO串列	REDO串列
[start_transaction,T1]	T1	
[commit, T1]		T1
[start_transaction,T2]	T2	T1
[commit, T2]		T1, T2
[start_transaction,T3]	T3	T1, T2
[checkpoint]	T3	
[start_transaction,T4]	T3, T4	
[start_transaction,T5]	T3, T4, T5	
[commit, T3]	T4, T5	T3
[commit, T4]	T5	T3, T4
[start_transaction,T6]	T5, T6	T3, T4

System Crash前，各變數之值分別為

A=29, B=366, C=200, D=2800, E=22 (由交易最後狀態判斷)。

若採用UNDO/NO-REDO演算法，

System Crash時，T5與T6交易尚未commit，必須進行復原。

故復原所有T5與T6的寫入動作，即

[write\_item, T6, D, 2600, 2800]

[write\_item, T5, A, 22, 29]

[write\_item, T5, B, 331, 366]

[write\_item, T5, E, 12, 22]

故需將D由2800復原為2600，A由29復原為22，B由366復原為331，E由22復原為12。

即A, B, C, D, E值各為

A=22, B=331, C=200, D=2600, E=12。