

《資料庫應用》

一、已知一個資料庫，其關連 (relation) 的屬性 (attribute) 之間，有以下的關係。有底線者為該關連的主鍵屬性 (key attribute)。

$R(\underline{r1}, r2)$

$S(\underline{s1}, s2)$

$T(\underline{s1}, \underline{t1}, t2)$ 其中s1就是relation S的s1。

$U(\underline{u1}, \underline{u2}, u3, u4)$ 其中u1 references r1, u2 references s1。

$V(\underline{r1}, v1)$ 其中r1就是relation R的r1。

除了上述所附註說明的屬性之外，其餘屬性互相之間並沒有任何關係。根據以上資料，試推測此資料庫的實體－關係綱目圖 (ER schema diagram) 該是如何？請繪出其ER schema diagram，並說明原因。圖中需註明所有能推知的基數比 (cardinality ratio)，如1:1、1:N、M:N等等。(15分)

答：

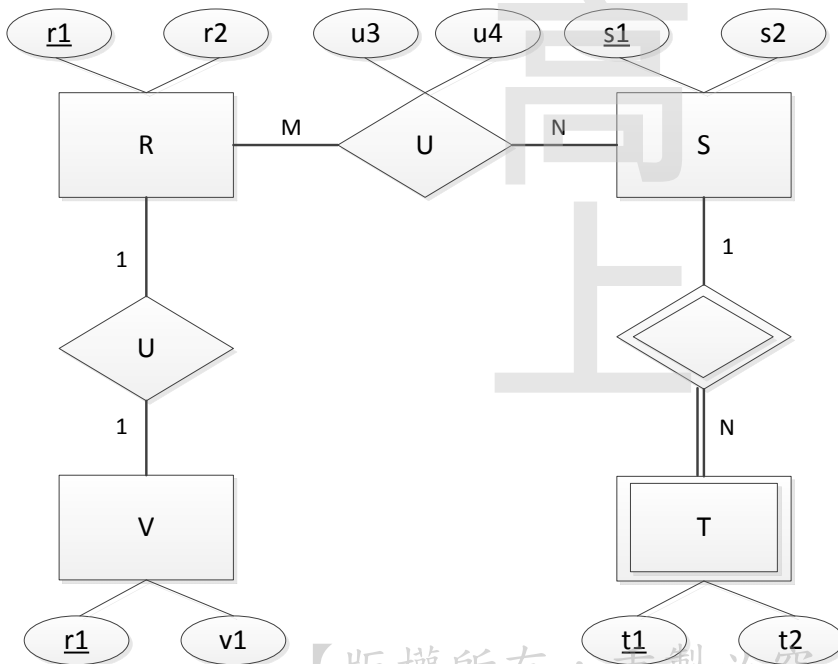
(一)R、S兩關聯，為標準單一主鍵搭配其他非鍵屬性，因此其應為一般個體。

(二)關聯T，主鍵s1, t1中，s1參考關聯S的主鍵，另包含t2一般屬性，因此其應為依附於S的弱個體，其中t1為部分鍵。

(三)關聯U，主鍵u1, u2分別參考關聯R的r1與S的s1，因此其應為關聯R與S之多對多關係。

(四)關聯V，主鍵r1，參考至R的r1，因此其應為一個一般個體，同時與關聯R為一對一之關係。

推測之ER圖如下：



【版權所有，重製必究！】

二、試回答以下有關DBMS user的問題：

(一)Tool developer主要是做什麼工作，試舉例說明。(5分)

(二)Parametric end user是如何使用系統，試舉例說明。(5分)

(三)Application programmer是做什麼工作，試說明之。(5分)

答：

(一) Tool developer

資料庫相關工具，通常是獨立購買，例如：資料塑模工具、資料庫績效監控工具、測試案例產生工具等，而 Tool Developer 即是負責建構這些相關工具者。

(二) Parametric end user

為資料庫的終端使用者，工作主要為查詢更新資料庫、或存取資料庫以產生所需報表。通常透過應用程式達到此目的。例如：負責進行交易處理、訂單操作及查詢等。

(三) Application programmer

應用程式的開發人員，通常在系統分析師撰寫出需求規格後，進行應用程式開發。例如：建立資料庫應用程式開發人員，利用程式開發工具或指定的程式語言，呼叫資料庫系統提供的「資料庫存取函式庫」(Database Access Library) 建立所需的資料庫應用程式。

三、某關連 $R(A, B, C, D, E, F, G)$ 的屬性之間有以下的關係。其中 “ \rightarrow ” 是 functional dependency。

$$AB \rightarrow CE, F \subseteq E, D \subseteq B, G \rightarrow B$$

(一) 試推導 (inference) 出屬性之間的 functional dependencies，並指出整個關連的 key attribute 應為何者。(10分)

(二) 請將此關連作第二正規化 (second normal form)，指明各關連的 key attribute 是何者。(5分)

(三) 請繼續將(二)的結果作第三正規化 (third normal form)，指明各關連的 key attribute 是何者。(5分)

答：

(一)

1. 因 $F \subseteq E$ ，故 $E \rightarrow F$ 。同理， $D \subseteq B$ ，故 $B \rightarrow D$ 。因此關聯 R 的 functional dependencies 可歸納為： $AB \rightarrow CE$ ， $E \rightarrow F$ ， $B \rightarrow D$ ， $G \rightarrow B$ 。

2. Key attributes：關聯中，僅為決定因素不為相依因素的屬性子集 (A, G) 中， G 可決定 B ， B 或 AB 又可決定其他 $CDEF$ ，因此 (A, G) 為關聯候選鍵。即題目所述 key attributes 為 (A, G) 。

(二) 關聯 R 中， $G \rightarrow B$ 故 $AG \rightarrow AB$ ，且 $AB \rightarrow CE$ ，因此 $AG \rightarrow CE$ 。其中 B, D 部分功能相依於主鍵 (A, G) 的 G 屬性。故 2NF 後將關聯分解為：

$$R_1(\underline{A}, \underline{G}, C, E, F), \text{ 其中 } AG \rightarrow C, AG \rightarrow E, E \rightarrow F$$

$$R_2(\underline{G}, B, D), \text{ 其中 } G \rightarrow B, B \rightarrow D$$

(三) R_1 中， $AG \rightarrow E$ ， $E \rightarrow F$ ，故 F 遞移相依於主鍵； R_2 中， $G \rightarrow B$ ， $B \rightarrow D$ ，故 D 遞移相依於主鍵。因此 3NF 後分解為：

$$R_1(\underline{A}, \underline{G}, C, E), \text{ 其中 } AG \rightarrow C, AG \rightarrow E$$

$$R_2(\underline{G}, B), \text{ 其中 } G \rightarrow B$$

$$R_3(\underline{E}, F), \text{ 其中 } E \rightarrow F$$

$$R_4(\underline{B}, D), \text{ 其中 } B \rightarrow D$$

四、已知有一手機製造公司資料庫綱目 (database schema) 如下：

工程師 (編號, 名字, 年齡, 性別, 薪水)

產品 (代號, 名目, 年銷售量, 售價)

設計 (編號, 代號)

請用 SQL 回答下列問題：

(一) 列出公司所有男工程師的平均薪水。(5分)

(二) 對於沒有參與任何一項產品設計的工程師，或者所參與設計的產品年銷售量低於 1 萬隻的，列出這些工程師的名字和年齡。(10分)

(三)針對年銷售量高於20萬隻的每一項產品，且參與設計此產品的工程師數目不超過5人的，列出其產品名目和工程師的平均薪水。(10分)

答：

(一)

```
SELECT  AVG(薪水)
FROM    工程師
WHERE   性別='男'
```

(二)

```
SELECT  名字, 年齡
FROM    工程師
WHERE   編號 NOT IN (SELECT 編號 FROM 設計) OR 編號 IN (SELECT 編號 FROM 設計, 產品
WHERE   設計.代號=產品.代號 AND 年銷售量<10000)
```

(三)

```
SELECT  名目, AVG(薪水)
FROM    產品, 工程師, 設計
WHERE   產品.代號=設計.代號 AND 設計.編號=工程師.編號 AND 產品.年銷售量>200000
GROUP BY 名目
HAVING COUNT(工程師.編號)<=5
```

五、Tr1和Tr2是二個交易 (transaction)。某次執行程序 (schedule) 如下：假設X和Y在硬碟裡的初始值是 $X = 1000$ ， $Y = 1000$ 。

Time	Transaction Tr1	Transaction Tr2
t1	read X	
t2		read Y
t3	$X = X - 1000$	
t4		read X
t5	write X	
t6		$Y = X + 1000$
t7		write Y

(一)請問按以上執行方式，此執行程序 (schedule) 是否為serializable? (5分)

(二)如果Tr1和Tr2按Two-phase locking protocol執行，請說明系統會如何進行? (10分)

(三)如果此程序是按time-stamping protocol執行，且Tr1在Tr2之後才進入系統 (亦即， $TimeStamp(Tr1) > TimeStamp(Tr2)$)。說明上面程序的執行結果是如何。(10分)

答：

(一)是，該排程僅一項T2 read X與T1 write X之衝突，故該排程為serializable。

(二)若採Two-phase locking protocol，所有鎖定動作必須在所有解除鎖定之前，因此Tr2在read X前先read-lock X後，Tr1則無法繼續鎖定，因此Tr1的write X動作需要等待Tr2的write Y動作執行後，完成擴展階段，進入收縮階段，並解除X鎖定後，Tr1的write X動作才可繼續進行。

(三)若採用time-stamping protocol，且Tr1較晚進入系統，則僅Tr2 read X時，有比Tr2晚進入系統的交易(Tr1)有存取過X，但亦僅為讀取操作；其餘操作皆無較該交易晚進入系統之其他交易有存取過相同資料項目。因此該排程的執行順序可順利完成。

Time	Transaction Tr1	Transaction Tr2
t1	read X (讀取X=1000)	
t2		read Y (讀取Y=1000)
t3	X=X-1000	
t4		read X (讀取X=1000)
t5	write X (寫入X=0)	
t6		Y=X+1000
t7		write Y (寫入Y=2000)

執行結果：X=0, Y=2000。

高點 · 高上

【版權所有，重製必究！】