

104年公務人員升官等考試、104年關務人員升官等考試  
104年交通事業公路、港務人員升資考試試題

代號：26230

全一張  
(正面)

等 級：薦任

類科(別)：資訊處理

科 目：資料庫應用

考試時間：2小時

座號：\_\_\_\_\_

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

一、考慮以下關聯資料表：

人事資料(教授姓名，教授工作證號碼，教授研究室編號，實驗室編號，助理姓名，助理工作證號碼，助理研究室編號，學生姓名，學生證號碼)

假如該關聯資料表「人事資料」有以下限制：

- 每位教授都分配有一間研究室及一間實驗室，且僅分配有一間研究室及一間實驗室。每一間研究室頂多分配給一位教授或一位助理。每一間實驗室都頂多分配給一位教授及該教授指導的學生。
- 每位教授都有分配一位且僅有一位固定的助理，但一位助理可同時協助至少一位教授處理工作。
- 每位助理都有分配一間且僅有一間研究室。
- 學生都有一位且僅有一位指導教授，而一位指導教授可以同時指導零位以上學生。
- 每位學生也都會在其指導教授的實驗室裡。

(一)試依前述限制，描述所有可能的功能相依性 (functional dependency)。(10分)

(二)試問如何以功能相依性的規則，依前述限制來決定出關聯資料表「人事資料」的主要鍵 (primary key)。(10分)

(三)試依前述限制繪出實體關聯圖 (entity relationship diagram)。(10分)

(四)請判定上述關聯資料表「人事資料」是否有符合第三正規化之要求，如為否，試問將如何修改使其能符合第三正規化要求之最少數個關聯資料表，並標示出關聯資料表之主要鍵 (primary key)。(10分)

(五)如每一位學生有行政相關業務需協助時，得找其指導教授所分配的助理來協助。試請依據前一題已修改成符合第三正規化之要求後之關聯資料表，寫一 SQL 程式片段，來輸出每一位學生需找那位助理協助，該程式片段應輸出每一位學生的姓名及所對應助理的姓名。(10分)

(請接背面)

等 級：薦任

類科(別)：資訊處理

科 目：資料庫應用

二、以下 Loan 與 Borrower 為兩個關聯資料表。當中 Loan 記錄著每一件貸款資料；Borrower 記錄著每一位借款人資料，BorrowerLname 與 BorrowerFname 分別為借款人的姓與名字。請寫一 SQL 程式片段，依姓名順序輸出每一位借款人的姓與名字及其借款次數。(10 分)

Loan

LoanID	BorrowerID	LoanDate	DueDate	ReturnDate
123	874	12-23-2013	01-24-2014	01-24-2014
124	874	12-23-2013	01-24-2014	NULL
125	876	12-29-2013	01-28-2014	NULL
126	874	01-25-2014	02-21-2014	NULL

Borrower

BorrowerID	BorrowerFname	BorrowerLname
874	Leon	Small
875	Gary	Lowe
876	Mark	Sanchez
877	Clywd	Morgan

三、在設計關聯式資料庫時，常會對關聯資料表進行正規化。

(一)試問對關聯資料表進行正規化之優點及缺點為何？(10 分)

(二)如整個資料庫系統中對某一資料表的資料內容，只會有讀資料的動作，試問對該資料表進行正規化有何優缺點？(10 分)

四、以下為在同步處理系統裡，執行交易 (transaction) 時常被執行的動作：

read\_item(Ti, A, B)：在交易 Ti 中，讀取資料 A 的內容，並將其存入資料 B 中。

write\_item(Ti, A, B)：在交易 Ti 中，將資料 B 的內容寫入資料 A 中。

read\_lock(A)：對資料 A 進行鎖定，使其他交易無法對其進行讀取的動作。

write\_lock(A)：對資料 A 進行鎖定，使其他交易無法對其進行寫入的動作。

unlock (A)：解除對資料 A 的鎖定。

設 T1 與 T2 為系統中正在被執行的兩個交易 (transaction)，而 X 與 Y 為 T1 與 T2 中所共用的兩個變數。T1 與 T2 之程式指令如下所示，並設在執行 T1 與 T2 前，X = 20；Y = 30。請討論在執行完 T1 與 T2 後，變數 X 與 Y 會有何可能結果。(20 分)

T1	T2
read_lock(Y);	read_lock(X);
read_item (T1, Y, A);	read_item (T2, X, C);
unlock (Y);	unlock(X);
read_lock(X);	read_lock(Y);
read_item (T1, X, B);	read_item (T2, Y, D);
unlock (X);	unlock(Y);
A:=A+B;	C:=C+D;
write_lock(Y);	write_lock(X);
write_item(T1, Y, A);	write_item(T2, X, C);
unlock (Y);	unlock (X);