

《資料庫應用》

| | |
|-------------|---|
| 試題評析 | 今年高考資料庫應用考題屬於十分中規中矩的命題，考題主要圍繞在資料庫設計、資料庫正規化之實作與資料庫交易管理上，考題難易度適中，並沒有過分簡易或刁難考生的考題出現。值得注意的是今年最後一題又考了分散式資料庫的考題，參照往年試題，可以發現NoSQL、分散式資料庫系統的考題逐漸成為考試中常見的題目類型。由於這兩個主題對於傳統資料庫考題來說相對新穎，理論內容也相對較深，若未來成為必考試題則題目難度可能會相較於往年有明顯的差異。 |
|-------------|---|

一、國立ABC大學的校務行政電腦化，其中有關教務資料，擬採用資料庫系統，經系統分析後，得到如下需求（Requirements）：

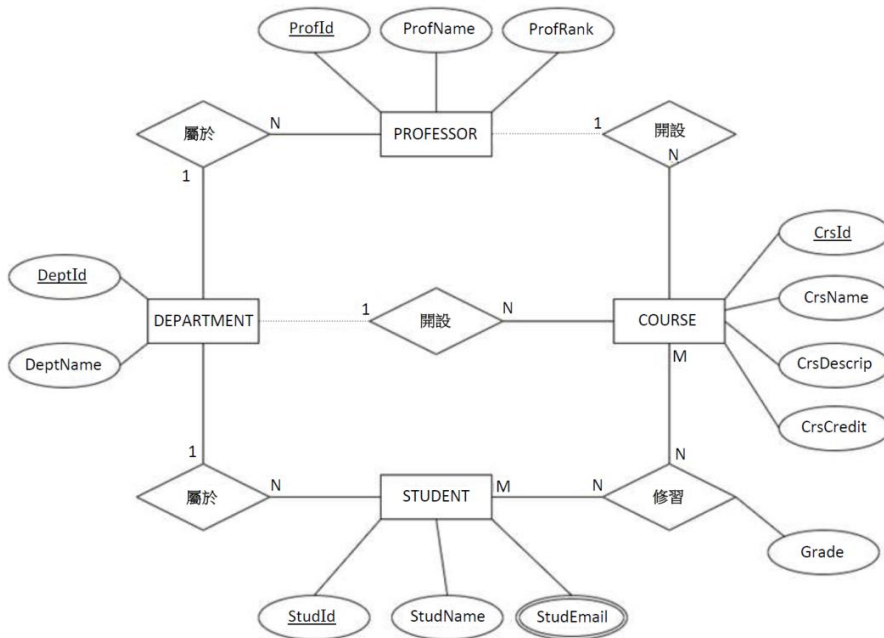
1. ABC大學由許多學系（DEPARTMENT）組成，每一學系有系代碼（DeptId）與系名稱（DeptName），系代碼可區別不同學系。
2. 各學系會開許多課程（COURSE），每一課程有課程代碼（CrsId）、課程名稱（CrsName）、課程說明（CrsDescrip）與課程學分數（CrsCredit），課程代碼可區別不同課程。
3. 每一學系有許多學生（STUDENT），每個學生有學生代碼（StudId）、學生姓名（StudName）與學生電子信箱（StudEmail），有學生會有多個電子信箱，學生代碼可區別不同學生。
4. 每一學系由許多教授（PROFESSOR）組成，每一教授有教職員工代碼（ProfId）、姓名（ProfName）與職稱（ProfRank），教職員工代碼可區別不同教授。
5. 每個學系會開授許多課程，每一課程僅由一學系開授。
6. 每個學系會有許多學生，每一學生僅隸屬於一學系。
7. 每個學系會有許多教授，每一教授僅隸屬於一學系。
8. 每一教授可以開授許多門課程，每一門課程僅由一個教授開授。
9. 每一學生可以修習許多課程，每一課程也可由許多學生來修習，每一學生修習的課程有一成績（Grade）。

請設計此教務關聯式資料庫綱要（Relational Database Schema），並標示出此資料庫關聯（Relations）的主鍵（PrimaryKey）與外來鍵（Foreign Key）。（25分）

| | |
|-------------|--|
| 試題評析 | 本題為資料庫設計實作試題，題目為典型的由系統分析需求要求同學製作出相對應的資料庫綱要，題目本身實作上並不困難，但十分講究同學對於閱讀題目需求的細心程度以正確達成其要求。 |
| 考點命中 | 《高點·高上資料庫應用講義》第一回，黃浩哲編撰，頁25-30。 |

答：

【版權所有，重製必究！】



DEPARTMENT(DeptId, DeptName)
 COURSE(CrsId, CrsName, CrsDescrip, CrsCredit, DeptId, ProfId)
 STUDENT(StudId, StudName, StudEmail, DeptId)
 PROFESSOR(ProfId, ProfName, ProfName, DeptId)
 STUDENT_COURSE(CrsId, StudId, Grade)
 STUDENT_Email(StudId, StudEmail)

其中STUDENT_COURSE之CrsId、StudId，STUDENT_Email之StudId亦為外鍵。

- 二、給予一關聯網要EMP-DEPT (EmpId, EmpName, EmpBdate, EmpAddr, DeptNum, DeptName, DmgrId)，主鍵 (Primary Key) 為{EmpId}，此關聯網要記錄員工參與部門的相關資料，員工有員工編號 (EmpId)、員工姓名 (EmpName)、員工生日 (EmpBdate) 與員工地址 (EmpAddr)，部門有部門編號 (DeptNum)、部門名稱 (DeptName) 與部門經理編號 (DmgrId)，而且給予一組功能依附性 (Functional Dependencies) {{EmpId}→{EmpName, EmpBdate, EmpAddr}，{DeptNum}→{DeptName, DmgrId}}，關聯網要EMP-DEPT是否為2NF？如不是，請將EMP-DEPT正規化至2NF，然後正規化至3NF，並論述分割 (Decompose) 的理論基礎。(25分)

| | |
|------|--|
| 試題評析 | 此題為資料庫正規化試題，主要考驗同學對於常見的資料庫第二、第三階正規化的了解程度，本題所給功能相依內容並不會太複雜，謹記2NF/3NF定義即可正確答題。 |
| 考點命中 | 《高點·高上資料庫應用講義》第一回，黃浩哲編撰，頁47-51。 |

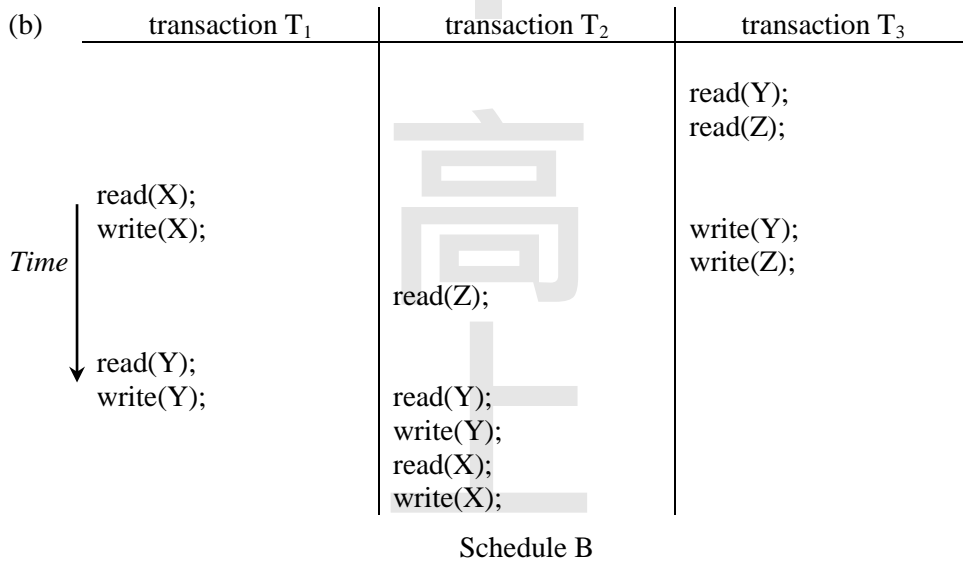
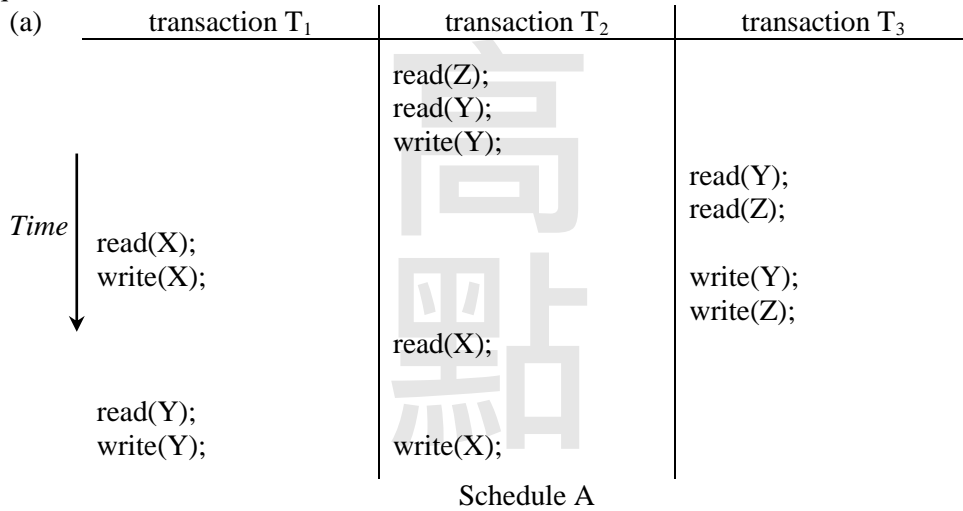
答：

由2NF定義：一關聯為2NF若且唯若關聯中所有屬性皆完全相依於候選鍵，及關聯屬性不存在部分功能相依，又由於本關聯網要EMP-DEPT僅有單一主鍵{EmpId}，因此不可能存在「部分相依」可能，因此本關聯符合2NF。

3NF：由於功能相依{DeptNum}→{DeptName, DmgrId}顯然違反3NF中所有非鍵值屬性僅與主鍵相關之定義，因此須將與主鍵無關之功能相依拆解至其他關聯：

EMP-DEPT(EmpId, EmpName, EmpBdate, EmpAddr, DeptNum)
 R'(DeptNum, DeptName, DmgrId)

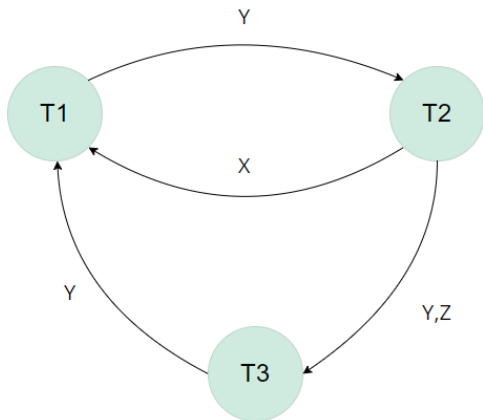
三、給予下列二個行程 (Schedules) A與B，請用一圖形演算法，利用行程中的讀 (Read) 與寫 (Write) 動作 (Operations) 構成圖形，圖形邊 (Edge) 上標示讀寫的資料項目 (Data Items)，以此演算法論述A與B兩行程是否具序列性 (Serializability)？如具序列性，請寫出對等序列行程 (Equivalent Serial Schedule)。(25分)



| | |
|-------------|------------------------------------|
| 試題評析 | 本題為資料庫交易管理中的序列化問題，考驗同學對於資源順序圖的掌握能力 |
| 考點命中 | 《高點·高上資料庫應用講義》第二回，黃浩哲編撰，頁53-57。 |

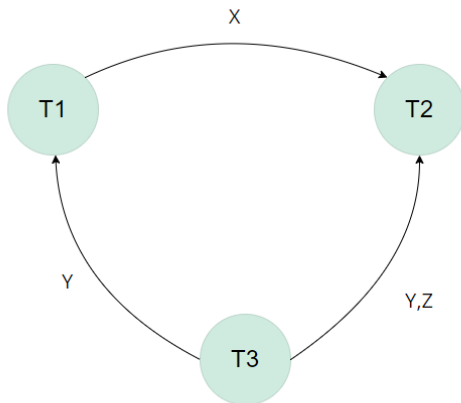
答：

(一)Schedule A 資源優先順序圖：**【版權所有，重製必究！】**



由於資源順序圖存在迴圈，故Schedule A不可序列化

(二)Schedule A 資源優先順序圖：



由於資源順序圖不存在迴圈，故Schedule B可序列化，對等序列行程： $\{T3 \rightarrow T1 \rightarrow T2\}$

四、分散式資料庫為一個分散在電腦網路的許多在邏輯上相關資料庫的集合，請畫出分散式資料庫系統三層主從伺服器架構（Three-tier Client-server Architecture），並論述其運作原理，分散資料的管理具有不同層次（Levels）的透明度（Transparency），請論述三種透明度及相關技術。（25分）

試題評析

本題為分散式資料庫觀念考題，題意為兩項分散式資料庫中常見的主題，題目難度不高，同學將考題要求之主題詳加論述即可得分。

考點命中

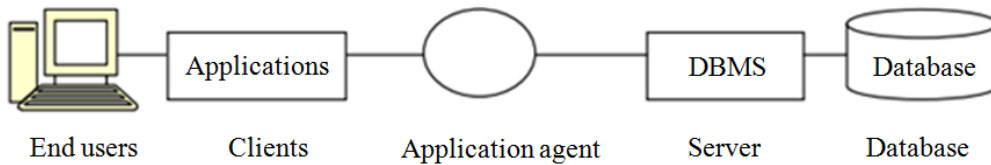
《高點·高上資料庫應用講義》第二回，黃浩哲編撰，頁16-19、23。

答：

將應用程式代理者(Application agent)置於Client與Server中間，存放企業邏輯(Business logic)，以處理Client與Server間往來的業務，可整合後端不同的Server，以統一的方式呈現內部的資料。

(1) 用戶端(Client)：負責使用者介面的呈現(User Interface)。

- (2) 伺服器端(Database Server)：負責資料庫資料的存取。
- (3) 應用程式伺服器(Application Server)：負責企業邏輯的規劃，及Client-Server 間的溝通。



分散式資料庫之透明度：分散式資料庫系統在邏輯上可視為單一資料庫系統，但實際上資料卻是分散在網路的多個地方，所謂的「透明度」即指使用者並不會認為他正在存取分散儲存的資料；對於使用者來說，感覺仍然是一個完整的資料庫，分散式資料庫共有三種透明度(Transparency)：

1. **位置透明度(Location Transparency)**：使用者或程式不需知道資料實體儲存位置與路徑，使用者下的指令與集中式的下法一樣，舉例：若使用者實際位置在台北市，欲查詢位在其他縣市之某圖書館館藏，使用者僅需下 `Select * from Books` 而非實際執行的 `Select * From Location.Books`。
2. **資料分割透明度(Fragmentation Transparency)**：使用者不需知道資料如何分割，亦可以存取原本正確的資料，舉例：使用者欲查詢完整系統之圖書館藏，無需執行各分館資料之 `Union` 僅須執行簡單查詢 `Select * from Books` 即可。
3. **重複透明度(Replication Transparency)**：資料庫因為某些需求可能被複製成多份，使用者存取時不需要知道資料的任何備份，且資料會同步更新於各個副本，舉例：使用者對於更新重覆存放在不同地點資料所下的指令應與集中式一致，例如：`Update Books where id = 3` 而各分散系統分別下更新指令。

高上

【版權所有，重製必究！】