

《資料庫應用》

- 一、資料模式 (Data Model) 很多，就其特性大致可以歸類為三類：高階 (High-Level, 或觀念 (Conceptual)) 資料模式、低階 (Low-Level, 或實體 (Physical)) 資料模式、表示 (Representational, 或實作 (Implementation)) 資料模式。請分別說明這三類資料模式為何？並為每一類資料模式各舉至少2個資料模式的例子。(20分)

試題評析	本題考High Level-conceptual、Low level-Physical、Representation Data Model三種資料模式，高階資料模式離使用者最近，也最容易了解。實作、低階則離使用者角度越來越遠。這三個資料模式有上課的同學應該不太容易被考倒，但要各舉兩個資料模式的例子可能就有點難了。沒法完整舉出兩個資料模式的例子的同學不用太灰心，因為別人應該也寫不出來，把握基本分才是考上的關鍵。
考點命中	《高點·高上資料庫講義》第一回，唐箏編撰，頁10-28。

答：

- (一) High Level-conceptual Data Model：
用使用者角度以及想法來描述資料，此模型連一般使用者也看得懂。
例如：entity relationship model, data structure diagram
- (二) Low level-Physical Data Model：
描述資料如何存入電腦中。這樣的模型主要是給電腦專家看的，不是給一般使用者。
例如：Data organization, Data architecture
- (三) Representation Data Model：
介於 High level 與 Low level data model 之間。這種模型一般使用者也可勉強了解，但又不離底層的軟體實作太遙遠。
例如：Relational Data Model, Network Model
- 二、某出版社要用關聯式資料模式建立可記錄其出版書籍的資料庫，設計了包括可記錄書籍的資料表 (Table) BOOK及記錄作者的資料表AUTHOR。BOOK中包含BookId (書籍編號)、Title (書名)、PubDate (出版日期)、Category (分類) 等屬性 (Attribute) 欄位，且BookId是其主鍵 (Primary Key, PK)；AUTHOR中包含AuthId (作者編號)、Name (姓名)、Birthday (出生年月日)、Tel (電話) 等屬性欄位，且AuthId是其主鍵。即資料庫中包含
BOOK(BookId, Title, PubDate, Category)
AUTHOR(AuthId, Name, Birthday, Tel)
兩個資料表。請依序回答下列問題：
- (一) 如果每本書可以有幾位作者，每一位作者也可以出版好幾本書，請對前項資料庫設計進行增修，使得出版社的資料庫可記錄每本書的作者，也可記錄每個作者所出版的書。作答時可仿效題目的描述方式說明您的設計，注意必須以英文命名明確表達資料表名稱及屬性欄位名稱，並明確指出主鍵及外來鍵 (Foreign Key, FK)，外來鍵並須指出與其對應的主鍵。(10分)
- (二) 承續並運用前一子題之資料庫增修，請寫出可查得屬於「資料庫」分類 (Category) 的所有書籍之SQL命令，此命令必須列出書籍的BookId, Title及作者人數，並依BookId由小而大依序列出。(15分)
- (三) 請運用子題(一)之資料庫增修寫出可查得曾經與名字叫「張山峰」的作者共同著作過的作之SQL命令，列出其姓名及電話。(15分)

試題評析	本題是關聯資料庫的基本題，大概只有第三小題要細心才能完整拿到分數。第三小題要注意SQL輸出可能會有重複，要加DISTINCT。而且要把張山峰從輸出中拿掉。
------	---

考點命中 《高點·高上資料庫講義》第二回，唐箏編撰，頁90-110。

答：

(一)需要另一個資料表，來表示BOOK and AUTHOR的多對多關係。此資料表取名為

BOOK_AUTHOR (BookID, AuthId)

PK： {BookID, AuthId}

FK： BookId references BOOK(BookId)

AuthId references AUTHOR(AuthId)

(二)SELECT B.BookID, B.Title, COUNT(A.AuthId) AS 作者人數

FROM BOOK B, AUTHOR A, BOOK_AUTHOR C

WHERE B.BookId = C.BookId AND A.AuthId = C.AuthId AND B.Category='資料庫'

GROUP BY B.BookId, B.Title

ORDER BY B.BookId ASC

(三)SELECT DISTINCT A.Name, A.Tel

FROM BOOK B, AUTHOR A, BOOK_AUTHOR C

WHERE B.BookId = C.BookId AND A.AuthId = C.AuthId AND B.BookId

IN (SELECT D.BookId

FROM BOOK D, AUTHOR E, BOOK_AUTHOR F

WHERE D.BookId = F.BookId AND E.AuthId = F.AuthId AND E.Name = '張山峰')

AND A.Name NOT IN ('張山峰')

三、某企業要建立自己的電子商務平台銷售自家商品，分析統整所有商品後，將商品分為若干大分類，每一大分類又分為若干次分類，每一次分類又分為若干細分類。假設此分類系統非常理想，每一個細分類號都是唯一的，且不會有同一個細分類同屬於不同次分類的情形；每一個次分類號也都唯一，且不會有同一個次分類同屬於不同大分類的情形，一個商品也只會歸類為單一細分類，不會同時歸類到不同細分類。依據這樣的分類系統，該公司的資料庫設計師設計出關聯式資料庫的資料表PRODUCT來儲存所有商品的資料，其中包含MajorCate（大分類號）、SubCate（次分類號）、FineCate（細分類號）、SeqNo（商品編號）、ProdName（商品名稱）、Price（單價），且任何一個商品的MajorCate、SubCate、FineCate所代表的該商品的各級分類號都是明確的，SeqNo是商品所屬細分類之流水號，在同一細分類中不會重複，但在不同細分類間會有相同的SeqNo。並以 (MajorCate, SubCate, FineCate, SeqNo) 為主鍵。也就是資料表PRODUCT為

PRODUCT (MajorCate, SubCate, FineCate, SeqNo, ProdName, Price) 請檢視此一設計，指出其設計錯誤或不良之處，並在不更動屬性設計的前提下修正之。(20分)

試題評析

為了避免資料庫的新增、更新、刪除異常，衍生出第一、第二、第三正規化的方式。本題在考什麼情況下會產生異常，基本上有讀懂正規化原因的同學都能順利作答。老師在此要再次強調、讀資料庫這科要把為什麼要這麼做給弄懂，不要死背。

考點命中

《高點·高上資料庫講義》第二回，唐箏編撰，頁1-14

答：

由題目可知{FineCate, SeqNo}即可形成主鍵。如果將MajorCate, SubCate 也放入主鍵的集合中，會產生新增、刪除異常。例如，當資料庫已經有一筆 MajorCate=1, FineCate=1, SeqNo=1的資料，如像題目同時用四個屬性當主鍵，則可以再新增一筆MajorCate=2, FineCate=1, SeqNo=1的資料。但這筆資料在此資料庫設計的邏輯中是不會發生，因為FineCate=1, SeqNo=1已經可以唯一決定產品，只會對應到一個MajorCate。而這筆錯誤的資料在有四個主鍵的情況下反而可以新增。如果將主鍵屬性減少到只有 {FineCate, SeqNo}，則這筆錯誤的資料將無法新增。

修正方法：將主鍵由{MajorCate, SubCate, FineCate, SeqNo}改成 {FineCate, SeqNo}

四、兩階段鎖定 (Two-Phase Locking) 技術可以用來做並行控制 (Concurrency Control)，請詳細說明符合兩階段鎖定協定的交易應遵循的規範為何？時程 (Schedule) 是指多個交易 (Transaction) 並行 (Concurrency) 執行時，各交易內的操作 (Operation) 間的執行順序。請以此觀點說明若所有交易都遵循兩階段鎖定協定撰寫，則這些交易並行執行時，時程必定是什麼時程？並說明為何遵循這種時程執行可以達到並行控制的目的？(20分)

試題評析	本題考Two-Phase Locking，有基本概念的同學應該都會寫。在此提醒同學未來在讀Two-Phase Locking 時，更應該要注意的是Two-Phase Locking是會產生死結的，必須要用Conservative 2PL才能避免死結。
考點命中	《高點·高上資料庫講義》第三回，唐箏編撰，頁4-24

答：

- (一)Two-Phase Locking 限定每個交易中所有的鎖定動作，包括 Read-lock 與 Write-lock，必須在所有解除鎖定 Unlock 動作之前。
- (二)遵守 Two-Phase Locking 的排程，如能成功執行完畢，保證皆為可序列化(serializable)
- (三)Two-Phase Locking 因為每個交易把所有的鎖定動作都集中在最前面，所有的解鎖動作都集中在最後面，而把所有跟其他交易可能有衝突的 operations 放在中間；中間的 operations 在執行時，因為資料已經鎖定，不會穿插跟其他交易有衝突的 operations。而在 lock 釋放後，因為此時缺乏 lock 保護，不會再執行可能會跟其他交易發生衝突的 operation，因此可以保證可序列化，達到並行控制的目的。

【版權所有，重製必究！】