

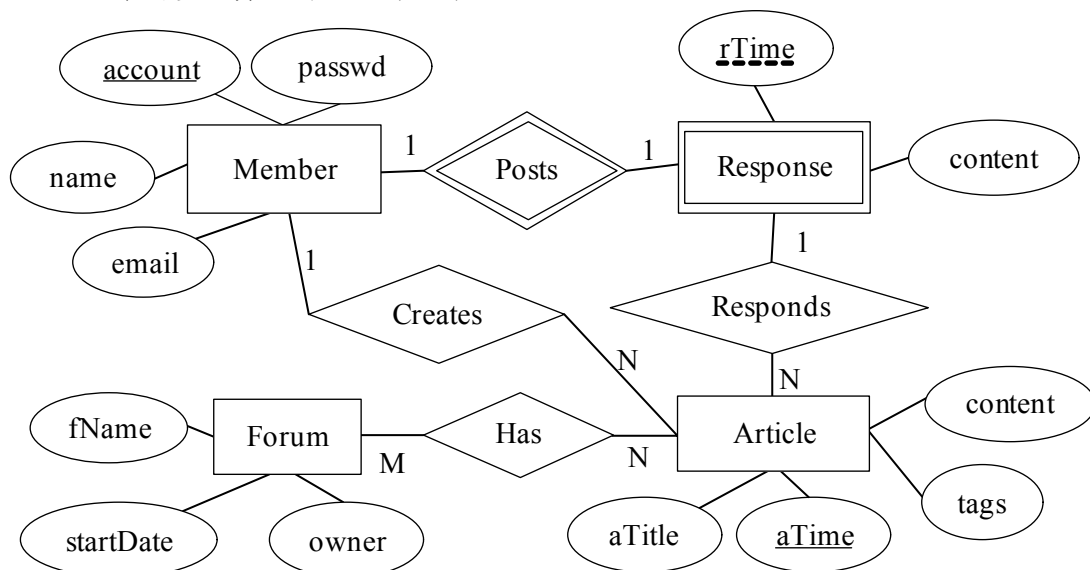
《資料庫應用》

試題評析

今年度試題題型分配平均，ERD、資料庫查詢、正規化、交易管理各出現一題，且皆為各章節之重要重點，故同學們應可掌握得不錯，一般同學應可拿到85分左右，程度較佳者可得90分以上。

- 一、假設你想設計一個討論區系統，該系統需要維護的資料包括會員（Member）、討論版（Forum）、主題文章（Article），和回應文章（Response）。資料需求如下：
- 會員（Member）：包括帳號（account）、密碼（passwd）、姓名（name），身分證字號（pId）和email。其中帳號和身分證字號均為唯一。
 - 討論版（Forum）：包括版名（fName）和開版日期（startDate），其中版名為唯一。此外，每一討論版必須剛好有一位版主（Owner），版主為會員。
 - 主題文章（Article）：包括主題（aTitle）、張貼時間（aTime）、標籤（tags）和內容（content），其中標籤可以有多个。此外，每一篇主題文章必須由一位會員張貼（Creates），且必須屬於某個討論版，其中沒有任何屬性是唯一的，但張貼會員和張貼時間合起來是唯一的。
 - 回應文章（Response）：包括張貼時間（rTime）和內容（content），一篇回應文章必須是回應一篇主題文章，但一篇主題文章可有多篇回應文章。此外，一位會員可以有多个回應文章，所回應的主題文章和張貼人也必須記載，張貼會員和張貼時間合起來是唯一的。

以下是一位同學所畫的實體關係圖（ERD）：



(一)請指出以上ERD裡不符合資料需求的部分。(16分)

(二)請修正該ERD以滿足所有的資料需求。(14分)

答：

(一)不符合需求處如下：

- 1.會員(Member)身份證字號(pId)未記錄
- 2.討論版(Forum)未記錄任何鍵值屬性，應為版名(fName)
- 3.討論版(Forum)的版主(Owner)亦為會員，故應為討論版與會員之關連，而非一屬性。
- 4.一主題文章(Article)可有多個標籤，故標籤應(tags)為多值屬性。
- 5.主題文章中，張貼會員+張貼時間加起來為唯一，故主題文章應為會員之一對多的弱個體，其中張貼時間

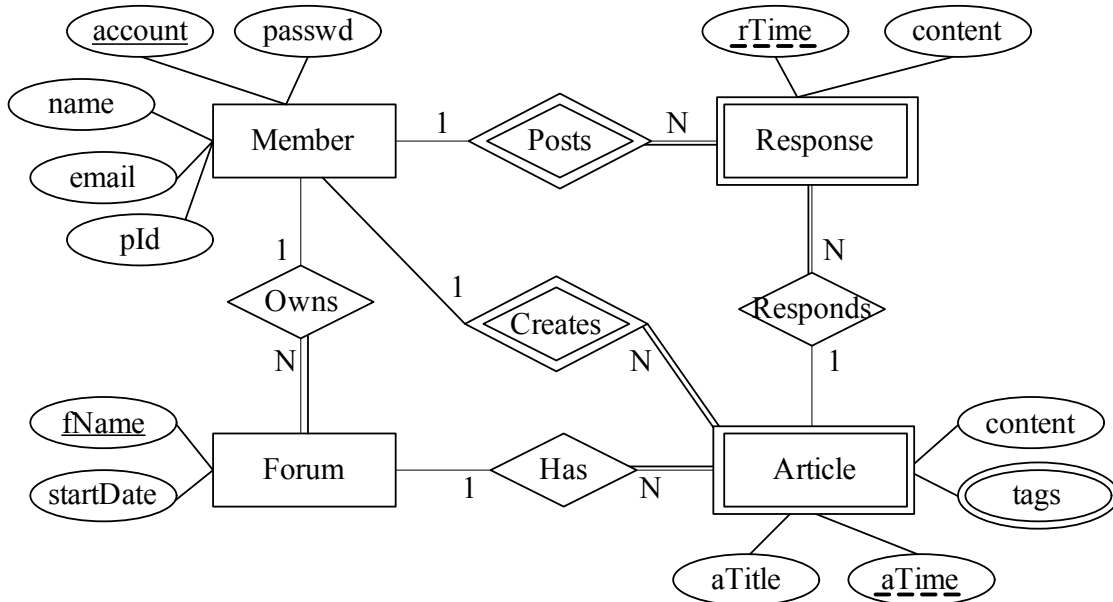
為主題文章的部分鍵。

6. 主題文章必須屬於某個討論版，故討論版與主題文章應為一對多之關連。

7. 每篇回應文章(Response)必須是回應一篇主題文章，但一篇主題文章可有多篇回應文章，故主題文章與回應文章應為一對多而非多對一之關係。

8. 一位會員可有多篇回應文章，且張貼會員和張貼時間合起來是唯一的，故回應文章應為會員之一對多之弱個體。

(二)



二、考慮以下的圖書館書籍流通系統資料庫綱目 (database schema)：

Book(callNo., isbn, title, subject)

//分類號, ISBN, 書名, 主題

Patron(pId, name, startDate)

//讀者編號, 讀者姓名, 開始日期

BorrowRecord(pId, callNo., borrowDate, returnDate)

//讀者編號, 分類號, 借閱日期, 歸還日期

請按題意表達出以下查詢：

(一) 請找出「張三」(是一讀者姓名)所借過的所有書的分類號和書名，請注意，一本書只能列出一次。

1. 用關聯代數式 (relational algebraic expression) 表達。(7分)

2. 用一個SQL敘述表達。(7分)

(二) 請列出電腦類 (即「subject= 'Computer'」) 每一本書籍的ISBN、書名和被借閱的次數。請用一個SQL敘述表達。(8分)

(三) 對於每一位總借閱次數超過10 (含) 次的讀者，列出其讀者編號和2009年的借閱總次數。(8分)

答：

(一)

1.

$$\pi_{Book.calNo, Book.title}(\sigma_{name='張三'}(Book * Book\ Record * Patron))$$

2.

```
SELECT DISTINCT Book .calNo, Book .title
FROM Book, BorrowRecord, Patron
WHERE Book.calNo = BorrowRecord.calNo AND BorrowRecord.pId = Patron.pId AND name = '張三'
```

(二)

```
SELECT isbn, title, COUNT(*)
FROM Book, BorrowRecord
WHERE Book.calNo = BorrowRecord.calNo AND subject='Computer'
GROUP BY isbn, title
```

(三)

```
SELECT pId, COUNT(*)
FROM BorrowRecord
WHERE borrowDate between '2009/01/01' AND '2009/12/31'
WHERE pId in (
    SELECT pId
    FROM BorrowRecord
    GROUP BY pId
    HAVING COUNT(*)>=10
)
```

三、考慮以下的商品交易資料庫綱目 (database schema)：

```
Product(pId, name, type, supplierId, supplierName)
//商品編號，商品名稱，商品種類，供應商編號，供應商名稱
Transaction(tId, date, payment)
//交易編號，交易日期，付款方式
TransactionDetail(tId, pId, amount, price, productDesc)
//交易編號，商品編號，商品數量，商品單價，商品敘述
```

此外，並有以下的函數相依 (functional dependency)：

```
{pId} → {name, type, supplierId}
{supplierId} → {supplierName}
{tId} → {date, payment}
{tId, pId} → {amount, price}
{pId} → {productDesc}
```

(一)請用以上函數相依推論出Product, Transaction, 和TransactionDetail的候選鍵 (candidate keys)，請注意必須用推論的方式來證明，否則不給分。(10分)

(二)以上資料庫綱目滿足第二正規式 (2NF) 嗎？滿足第三正規式 (3NF) 嗎？若不滿足，請舉出並說明所有造成不滿足的函數相依。(15分)

答：

(一)

1.Product中，{pId} → {name, type, supplierId}，{supplierId} → {supplierName}

(1)pId → supplierId，且supplierId → supplierName，故pId → supplierName

(2)由 {pId} → {name, type, supplierId} 與 pId → supplierName 可得，{pId} → {name, type, supplierId, supplierName}。

(3)由(1)、(2)，以pId為候選鍵時，pId可決定Product中所有非鍵屬性(候選鍵之唯一性)，且pId無法再移

除任何屬性(候選鍵之最小性)

故pId為關聯Product之候選鍵。

2.Transaction中，{tId}→{date, payment}

(1)以tId為候選鍵時，tId可決定Transaction中所有非鍵屬性(候選鍵之唯一性)，且tId無法再移除任何屬性(候選鍵之最小性)

故tId為關聯Transaction之候選鍵。

3.TransactionDetail中，{tId, pId}→{amount, price}，{pId}→{productDesc}

(1)pId→productDesc，故tId, pId→tId, productDesc

(2)tId, pId→amount, price且tId, pId→productDesc，故tId, pId→amount, price, productDesc。

(3)由(1)、(2)，以tId, pId為候選鍵時，tId, pId可決定TransactionDetail中所有非鍵屬性(候選鍵之唯一性)；且僅tId無法決定productDesc，僅pId無法決定amount及price(候選鍵之唯一性)，故tId, pId為關聯之候選鍵

故pId為關聯TransactionDetail之候選鍵。

(二)

1.關聯Product

(1)候選鍵僅一個屬性，不可能有部分功能相依於主鍵之情形產生，故Product為2NF。

(2)pId→supplierId，且supplierId→supplierName，故pId→supplierName中，supplierName遞移相依於主鍵pId，故Product非3NF。

2.關聯Transaction

(1)候選鍵僅一個屬性，不可能有部分功能相依於主鍵之情形產生，故Transaction為2NF。

(2)無任何遞移相依於主鍵之關連，故Transaction為3NF。

3.關聯TransactionDetail

(1)pId→productDesc，即productDesc部分功能相依於主鍵{tId, pId}，故TransactionDetail非2NF。

(2)TransactionDetail不滿足2NF，故其亦非3NF。

四、考慮以下兩支資料庫交易(database transaction)程式：

<p>T1 :</p> <p>read(X, a)</p> <p>//將資料庫裡的資料項X寫到變數a</p> <p>a=a+100</p> <p>write(a, X)</p> <p>//將變數a的值寫到資料庫裡的資料項X</p> <p>commit(T1)</p>	<p>T2 :</p> <p>read(X, b)</p> <p>//將資料庫裡的資料項X寫到變數b</p> <p>b=b-100</p> <p>write(b, X)</p> <p>//將變數b的值寫到資料庫裡的資料項X</p> <p>commit(T2)</p>
---------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------

假設資料庫裡的資料項X之初值為100，考慮以下排程(schedule)：

T1 : read(X, a)
 T2 : read(X, b)
 T2 : b = b - 100
 T1 : a = a + 100
 T2 : write(b, X)
 commit(T2)
 T1 : write(a, X)
 commit(T1)

(一)以上排程的執行結果為何？正確的排程執行結果應該是什麼？(7分)

(二)請問兩階段鎖定法 (two phase locking, 簡稱2PL) 會允許以上的排程執行嗎？請解釋。
(8分)

答：

(一)

1.

排程操作	變數/資料結果
T1: read(X,a)	X=100, a=100
T2: read(X,b)	X=100, a=100, b=100
T2: b=b-100	X=100, a=100, b=0
T1: a=a+100	X=100, a=200, b=0
T2: write(b,X)	X=0, a=200, b=0
commit(T2)	X=0, a=200, b=0
T1: write(a,X)	X=200, a=200, b=0
commit(T1)	X=200, a=200, b=0

故以上排程執行結果，X為200。

2.交易T1之操作為讀取X值，加上100後存入資料庫；T2之操作為讀取X值，減去100後存入資料庫；X之初值為100，故正確排程執行結果應為 $100+100-100=100$ ，即X=100。

(二)

若採用2PL，將readLock與WriteLock加入排程中，

T1	T2
T1: readLock(X) T1: read(X,a)	T2: readLock(X) T2: read(X,b) T2: b=b-100
T1: a=a+100	<u>T2: writeLock(X)</u> T2: write(b,X) commit(T2)
T1: write(a,X) commit(T1)	

上表中，T2寫入操作前的鎖定動作，即畫底線之writeLock(X)無法執行，因X已被T1 readLock住，但T1執行至寫入操作前仍需作X項目的writeLock；根據2PL定義，T1在擴展階段無法解除任何鎖定，故題目之排程無法順利執行完畢。