

1. (10%) 關聯中的序列值必須滿足某些限制(constraint)，如果一個資料庫的所有關聯裡的序列值都滿足這些限制，我們就稱該資料庫是“一致的(consistent)”。請問，在關聯模式裡的限制有那5項？並詳細說明之。

- ✓ 定義域限制(Domain constraint):
 - 每一個關聯的每一筆序列的每一屬性值必需是其屬性定義域裡的單一值
- ✓ 關聯鍵限制(Key constraint):
 - 指的是資料庫裡的每一關聯，沒有任兩個序列值的關聯鍵值是相同的
- ✓ 實體完整限制(Entity integrity constraint):
 - 每一關聯的每一筆序列值的主鍵值不得為空值(但次要鍵值可以是空值)
- ✓ 參考完整限制指的是(Referential integrity constraint):
 - 序列值裡的外部鍵值，如果不是空值，則該值必須存在於其所參考的關聯之主鍵值裡
- ✓ 語意完整限制(Semantic integrity constraint):
 - 然而有時迷你世界裡需要設定一些額外的限制，比如：
 - 交易總金額不得少於 100 元時才可用信用卡付帳；
 - 信用卡到期日必須超過交易日六個月以上等。
 - 凡此種種皆稱為語意完整限制。

註：同學回答時舉一個例子就好。

2. (12%) SQL 提供了那三種語言，其用途分別為何？

PPT 6-4&5

- ✓ 資料定義語言 DDL(Data Definition Language)：用來定義資料庫、資料表、檢視表、索引、預存程序、觸發程序、函數等資料庫物件。
- ✓ 資料操作語言 DML(Data Manipulation Language)：用來處理資料表裡的資料。
- ✓ 資料控制語言 DCL(Data Control Language)：用來控制資料表、檢視表之存取權限，提供資料庫的安全性。

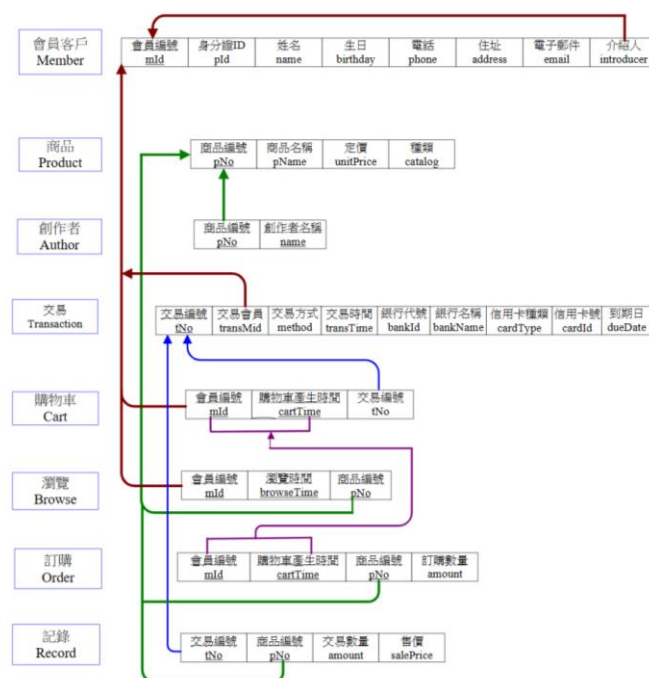
3. (12%)請分別說明第一正規式、第二正規式、第三正規式及 BCNF 之定義。

- ✓ 1NF：R 的每個屬性都是簡單且單值
- ✓ 2NF：對於 R 的每一個關聯鍵 K (i.e. 主鍵及次要鍵)，所有非鍵屬性都完全函數相依於 K
- ✓ 3NF：對於每一個關聯鍵 K，R 的每一個非鍵屬性都不可遞移函數相依於 K
- ✓ BCNF：對於 R 的每一個函數相依 $A \rightarrow B$ ，A 都是超級鍵

4. (30%) 請參考圖 1 的書店系統資料庫綱目，請用關聯代數式表達以下查詢：

- (1) 列出每一商品之編號、名稱、定價和創作者人數，請注意沒有創作者的商品資訊也要列出。
- (2) 列出每一商品的編號、名稱、定價和創作者人數。
- (3) 列出每一女性會員（身份證的第二碼為 2）的會員編號、姓名和購買總金額。
- (4) 列出購買過所有「孫燕姿」CD（孫燕姿為創作者）的會員之會員編號和姓名。
- (5) 列出總交易金額超過 800 元的交易之交易編號、會員姓名和總交易金額。
- (6) 對於每一商品，列出其在 2005 年被購買的總金額。

第1頁/共 3 頁



1.

RESULT = pNo, pName, unitPrice \mathcal{F} COUNT name (Product \bowtie Product.pNo = Author.pNo Author)

2.

若是沒有創作者的商品資訊不用列出，則如下：

RESULT = pNo, pName, unitPrice \mathcal{F} COUNT name (Product \bowtie Product.pNo = Author.pNo Author)

若是沒有創作者的商品資訊需列出，則如下：

RESULT = pNo, pName, unitPrice \mathcal{F} COUNT name (Product \bowtie Product.pNo = Author.pNo Author)

3.

T1 = Transaction * Record

FemaleMember = $\sigma_{pId \text{ LIKE } '_2\%'} \text{Member}$

RESULT = mId, name \mathcal{F} SUM salePrice (FemaleMember \bowtie mId=transMid T1)

4.

T1 = $\pi_{pNo} (\sigma_{name = \text{'孫燕姿' AND category = 'CD'}} (\text{Product} * \text{Author}))$

T2(pNo, mId) = $\pi_{pNo, transMid} (\text{Transaction} * \text{Record})$

RESULT = $\pi_{mId, name} ((T2 \div T1) * \text{Member})$

5

T1(tNo, mId, purchase) = tNo, transMid \mathcal{F} SUM salePrice (Transaction * Record)

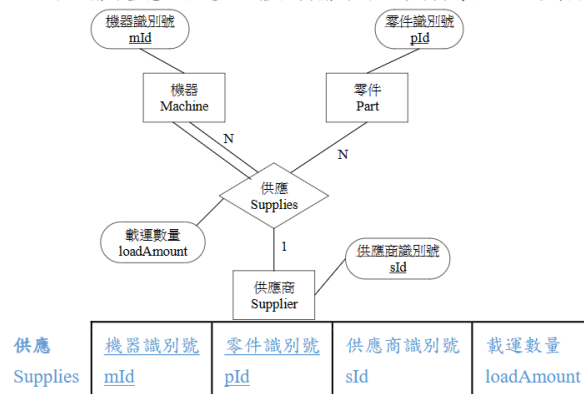
RESULT = $\pi_{tNo, mId, name} ((\sigma_{purchase > 800} T1) * \text{Member})$

6

T1 = $\sigma_{transTime \text{ LIKE } '2005\%'} (\text{Transaction} * \text{Record})$

RESULT = pNo \mathcal{F} SUM salePrice (T1)

5. (6%)針對下列 ERD 中之關係型態“供應”，產生其關聯網目（將其屬性及主鍵列出）。



6. (30%) 某公司的員工常需訂購中午的外送餐食。為方便員工，所以公司決定由你來開發一套外送餐食訂購系統，該系統存有數個有提供外送服務的餐廳和可選擇的餐點，並在每天早上開放讓公司員工上線選擇午餐要吃的餐點。為方便結帳，該系統採取預付制，也就是說員工必須先存一些錢給管理該系統的秘書，再按所點的餐點價錢扣款，扣完之前必須再存一些錢進去。具體說來，

第3頁/共 3 頁

該系統的資料需求如下：

- ✓ 員工 (Employee)：包括員工編號 (eld)、姓名 (name)、生日 (birthday)、身分證編號 (pld)、隸屬部門 (department)、存款餘額 (balance) 和電話 (phones)。其中員工編號和身分證編號均為唯一，且可能有一支以上的電話。
- ✓ 餐廳 (Store)：包括餐廳名稱 (sName)、餐廳住址 (address)、電話 (phone) 和說明 (description)。其中餐廳名稱為唯一。
- ✓ 餐點 (FoodItem)：包括餐點名稱 (fName)、餐點說明 (description)、單價 (unitPrice)、折扣說明 (discount)。餐點必然屬於一個餐廳。其中沒有任何一個屬性為唯一，但同一家餐廳的餐點其餐點名稱必不同。
- ✓ 個人餐點訂購 (Order)：包括訂購日期 (date)、訂購說明 (description) 和訂購金額 (salePrice)。一個個人餐點訂購必須屬於(剛好)一位員工，且一位員工在任何一天最多只能有一個個人餐點訂購。此外，一個個人餐點訂購會包括一個或多個餐點，餐點的數量 (num) 也必須記載。
- ✓ 存款紀錄 (Deposit)：包括存款時間 (dTime) 和存款金額 (amount)。一筆存款紀錄必然屬於一位員工，而一位員工的數個存款紀錄之存款時間必不同。

根據以上需求，目前已畫出其資料庫綱目 (如圖 2 所示)。

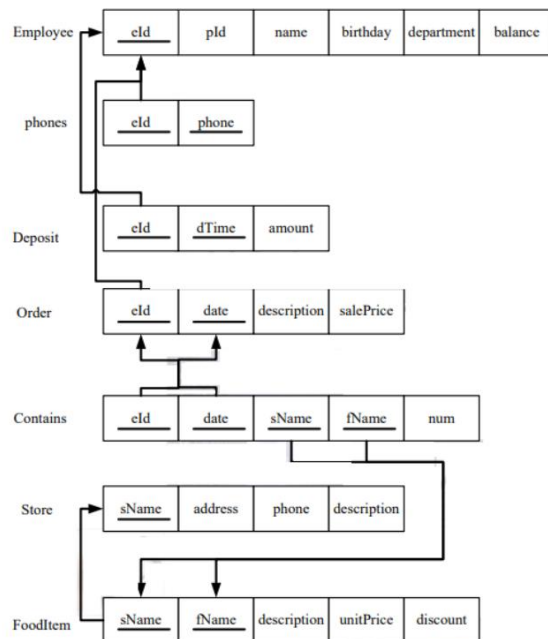


圖 2. 外送餐食系統資料庫綱目

請依外送餐食系統資料庫綱目，透過 SQL 語法，表達出以下的查詢或資料維護。

- (1) 列出有賣「紅燒牛肉麵」(為餐點名稱)的餐廳之名稱和住址。
- (2) 列出每一餐點之餐點名稱、店名和點購總次數。
- (3) 列出每一位員工的編號、姓名、餘額和在 2005 年 11 月的消費總金額。
- (4) 列出有點過所有「小蕃茄」(為一餐廳名稱)所外送餐點的員工之員工編號和姓名。
- (5) 若一餐廳不再提供某個餐點時，我們可能會想刪除該筆 **FoodItem** 記錄時，這會影響到哪些資

料表？但我不想像連鎖刪除這些參考該 FoodItem 的記錄，該怎麼辦？

- (6) 請產生一個 Trigger，每增加一筆個人餐點訂購記錄時就將 Employee 資料表裡該訂購員工記錄的 balance 欄位值減去該新增訂購記錄的 salePrice 欄位值。

1.

```
SELECT sName, address
FROM Store, FoodItem
WHERE Store.sName = FoodItem.sName AND
      fName = '紅燒牛肉麵';
```

2.

```
SELECT sName, fName, SUM(num)
FROM Contains
GROUP BY sName, fName;
```

3.

```
SELECT E.eId, name, balance, SUM(salePrice)
FROM Employee AS E LEFT OUTER JOIN Order AS O
      ON (E.eId = O.eId AND E.date = O.date
      AND to_char(date, 'yyyymm') = '200511');
```

4.

```
SELECT eId, name
FROM Employee, Contains
WHERE Employee.eId = Contains.eId AND sName = '小蕃茄';
```

1.

會影響到 Contains 資料表。可在 FoodItem 資料表上新增一個欄位“currentSupply”，代表該餐點是否目前有供應。若目前有供應，則其欄位值為'Y'，否則為'N'。因此，若一餐點要由目前供應改成不供應，則只要將其 currentSupply 欄位值改成'N'即可，而不用去刪除 Contains 資料表裡的任何記錄。

2.

```
CREATE TRIGGER ChangeBalance
AFTER INSERT ON Order
FOR EACH ROW
UPDATE Employee
SET balance = balance - New.salePrice
WHERE eId = New.eId;
```