

# 《資料庫應用》

## 試題評析

今年考題較偏重觀念題，沒有艱深的計算或應用題，但須觀念清楚，才可取得較高的分數。如第三題，許多答案可以一眼看出，但解題時需要脈絡清晰，寫出理由來做驗證。一般同學應可取得80分上下，程度較佳者可以超過90分。

- 一、某公司為管理員工出差，登錄有員工資料包括身分證字號、姓名、工作部門編號、薪水、員工帳號；出差資料有出差編號、出發地、抵達地、出發日、抵達日、出差費。已知一個員工只在一個部門做事且只有一份薪水登錄在一個帳號；不是每一位員工都會出差。一件出差，只有一種行程和出差費，有一到多個員工參與，行程一樣，出差費也一樣。假設關連設計如下：
- 員工（身分證字號、姓名、工作部門編號、薪水、員工帳號）  
 出差（身分證字號、出差編號、出發地、抵達地、出發日、抵達日、出差費、員工帳號）
- (一) 以上述關連為例，舉例說明超級鍵（superkey）和外部鍵（foreign key）的性質。（10分）
- (二) 說明上述關連設計在資料更新和刪除時有何嚴重缺點？（10分）要如何改正，以維持一致性。（5分）

**答：**

(一)

1. 超級鍵(superkey)指可以唯一決定該關連之值組的屬性子集，但與候選鍵不同的是，超級鍵不需為最小之屬性子集。以員工關連為例，(身分證字號)、(身分證字號, 姓名)、(身分證字號, 員工帳號)、(身分證字號, 姓名, 員工帳號)，上述這些屬性子集皆可稱為員工的超級鍵，因這些屬性集合皆不會重複，可辨識員工個體。
2. 外部鍵(foreign key)為一關聯中的屬性子集，而此屬性子集在其他關聯中扮演主鍵(候選鍵)的角色。例如出差關連的身分證字號欄位，便為該關連的一個外部鍵，其值參考員工關連的主鍵身分證字號欄位。

(二) 一件出差可有一到多個員工參與，上述設計會造成之缺點，例如：

1. 更新一件出差編號的出發地、抵達地、出發日等資訊時，因資料有多筆，造成有的改到有的沒改到；
2. 刪除一員工的出差資訊，可能同時刪除了該件出差的詳細資訊，造成刪除過多資訊。
3. 改正方式應將關連進行正規化，以避免上述異常現象。例如將出差關連分割為：出差(出差編號，出發地，抵達地，出發日，抵達日，出差費)；員工出差(身分證字號，出差編號)。

## 【高分閱讀】

唐箏，《資料庫應用講義第一回》，頁36-37及補充筆記。

二、假設某公司資料庫有供應商資料、零件資料、供應零件資料：

Supplier (SID, SNAME, ADDRESS) // 供應商 (供應商代號, 供應商姓名, 地址)

Parts (PID, TYPE, PRICE) // 零件 (零件編號, 零件類別, 價格)

Offer (SID, PID) // 供應零件 (供應商代號, 零件編號)

註：劃底線的屬性為主鍵

- (一) 以關連代數運算子 {selection  $\sigma$ 、project  $\Pi$ 、equal-join  $\bowtie$ 、division /} 進行查詢“找出供應所有零件的供應商姓名”。(10分)
- (二) 以SQL表示查詢“找出那些價格最高的零件編號PID和其價格PRICE”。(10分)
- (三) 以SQL表示查詢“找出每個零件類別，其零件數和其平均價格”。(5分)

答：

(一)  $\Pi_{SNAME}(\text{Supplier} \bowtie_{\text{Supplier.SID=Offer.SID}}(\text{Offer} / \Pi_{PID}(\text{Parts})))$ 

(二)

```

SELECT  PID, PRICE
FROM    Parts
WHERE   PRICE=
(      SELECT  MAX(PRICE)
      FROM    Parts
      )

```

(三)

```

SELECT  TYPE, COUNT(*), AVG(PRICE)
FROM    Parts
GROUP BY TYPE

```

## 【高分閱讀】

唐箏，《資料庫應用講義第二回》，頁96。

三、已知關聯  $R = \{A, B, C, E, G, H\}$  為1NF，其functional dependency $F = \{A \rightarrow B, B \rightarrow C, AB \rightarrow CEG, H \rightarrow A\}$ 。

(一)何者可作R的主鍵 (primary key)？要說明理由才可得分。(5分)

(二)簡化集合F成為M使得M含最少數量的functional dependency且  $F^+ = M^+$ 。(5分)

(三)若將R分解成R1 (A, B, C, E, G) 與R2 (H, A)，則是無失誤合併的分解嗎？要說明理由驗證才可得分。(5分)

(四)將R分解為無失誤合併且保留F的3NF關聯，註明各關連的主鍵，要注意控制屬性在各關連的重覆出現。(10分)

答：

(一)R的主鍵為H。因所有屬性中，只有H屬性不為任何功能相依的因素，故其必為部分鍵。

 $H \rightarrow A, A \rightarrow B$ ，故  $H \rightarrow B$  $H \rightarrow B, B \rightarrow C$ ，故  $H \rightarrow C$  $H \rightarrow A, H \rightarrow B$ ，故  $H \rightarrow AB$ 。且  $AB \rightarrow CEG$ ，故  $H \rightarrow CEG$ 由上述關係可得， $H \rightarrow ABCEG$ ，即H可決定關連中所有屬性，故其可作為R的主鍵。

(二)依下述步驟化簡：

1.右邊只有一個相依因素：

 $A \rightarrow B$  $B \rightarrow C$  $AB \rightarrow C$  $AB \rightarrow E$  $AB \rightarrow G$  $H \rightarrow A$ 

2.去除多餘的決定因素

因  $A \rightarrow B$  故  $A \rightarrow AB$ ，則  $AB \rightarrow C$ 、 $AB \rightarrow E$ 、 $AB \rightarrow G$  可化簡為  $A \rightarrow C$ 、 $A \rightarrow E$ 、 $A \rightarrow G$ 

故功能相依為

 $A \rightarrow B$  $B \rightarrow C$  $A \rightarrow C$  $A \rightarrow E$  $A \rightarrow G$  $H \rightarrow A$ 

3.去除多餘的FD

因 $A \rightarrow B$ 且 $B \rightarrow C$ ，故 $A \rightarrow C$ ，故 $A \rightarrow C$ 為多餘，可刪除

4.最簡功能相依 $M^+$ 為

$M^+ = \{ A \rightarrow B, B \rightarrow C, A \rightarrow E, A \rightarrow G, H \rightarrow A \}$

(三)將 $R$ 分解為 $R_1(A, B, C, E, G)$ 與 $R_2(H, A)$ 為無失誤合併的分解。

因關連 $R$ 中，

$A \rightarrow B, B \rightarrow C$ ，故 $A \rightarrow C$

$A \rightarrow E$

$A \rightarrow G$

故 $A \rightarrow BCEG$ ，因此 $R_1(A, B, C, E, G)$ 中， $A$ 為關連 $R_1$ 之主鍵。

$R_2(H, A)$ 中， $H \rightarrow A$ ，因此 $H$ 為關連 $R_2$ 之主鍵。

$H \rightarrow A$ ，且 $A \rightarrow BCEG$ ，故 $R_1$ 及 $R_2$ 可無失誤合併為 $H \rightarrow ABCEG$ 。

(四)因關連 $R$ 中， $H \rightarrow A, A \rightarrow BEG, B \rightarrow C$ ，故 $BCEG$ 遞移相依於主鍵 $H$ ，可先將關連 $R$ 分解為

$R_1(A, B, C, E, G)$ ，其中 $A \rightarrow B, B \rightarrow C, A \rightarrow E, A \rightarrow G$

$R_2(H, A)$ ，其中 $H \rightarrow A$

$R_1$ 中， $A \rightarrow B, B \rightarrow C$ ，故 $C$ 遞移相依於主鍵 $A$ ，可再將 $R_1$ 分解為

$R_1(A, B, E, G)$ ，其中 $A \rightarrow B, A \rightarrow E, A \rightarrow G$

$R_3(B, C)$ ，其中 $B \rightarrow C$

因此，3NF後的關連如下：

$R_1(A, B, E, G)$ ，其中 $A \rightarrow B, A \rightarrow E, A \rightarrow G$

$R_2(H, A)$ ，其中 $H \rightarrow A$

$R_3(B, C)$ ，其中 $B \rightarrow C$

#### 【高分閱讀】

唐筭，《資料庫應用講義第二回》，頁4。

四、多數資料庫系統會定期執行checkpoint，以提高毀損回復(recovery)的執行成效。

(一)說明資料庫系統是如何執行checkpoint？(5分)

(二)在有checkpoint機制配合下，針對在checkpoint後，毀損前，已committed和uncommitted的交易(transaction)，系統如何進行延遲更新(deferred-update)的回復？(10分)

(三)在有checkpoint機制配合下，針對在checkpoint後，毀損前，已committed和uncommitted的交易(transaction)，系統如何進行立即更新(immediate-update)的回復？(10分)

答：

(一)系統會定期將一個檢查點(checkpoint)強迫寫入至日誌(log)，並將所有已提交的交易中所有寫入(write)動作真正寫入資料庫中。在系統發生錯誤時，檢查點可確定哪些交易在發生錯誤前已提交了。檢查點的寫入代表此點之前所有已提交的交易，在系統毀損、錯誤時，不需要被重新執行(redo)。Checkpoint步驟如下：

- 1.暫停所有交易動作。
- 2.將所有在主記憶體中，已委任(committed)的交易操作，強制寫入系統日誌。
- 3.寫入一個check point至系統日誌，並將日誌強制寫入到磁碟(資料庫)中。
- 4.所有交易繼續。

(二)延遲更新(Deferred update)技術是一種NO-UNDO/REDO演算法。

- 1.uncommitted的交易，所有更新動作不會影響到資料庫內容，因此不需UNDO，可忽略此交易之復原動作。
- 2.在checkpoint後committed的交易，需進行REDO動作，確認此交易動作真正影響資料庫。
- 3.在checkpoint前已committed的交易，交易動作已真正反應至資料庫，可忽略不需進行任何動作。

(三)立即更新(Immediate update)技術是一種UNDO/REDO或UNDO/NO-REDO演算法。以下採UNDO/REDO為例。

- 1.uncommitted的交易，所有更新動作會真正影響到資料庫內容，因此需要UNDO操作。
- 2.在checkpoint後committed的交易，需進行REDO動作，確認此交易動作真正影響資料庫。

3.在checkpoint前已committed的交易，交易動作已真正反應至資料庫，可忽略不需進行任何動作。

【高分閱讀】

- 1.唐筭，《資料庫應用講義第三回》，頁4。
- 2.唐筭，《資料庫應用講義第三回》，頁4、頁37-38、及頁42-43。



# 高點 · 高上高普特考