

## 第二章 關聯式資料庫

1

### 章節大綱

2

- 資料庫管理系統常見的三種模型。
- 關聯式資料庫管理系統規則建議。
- 認識關聯。
- 資料的完整性。
- 資料庫的正規化。
- 關聯式代數。
- 結構化查詢語言SQL。

## 資料庫管理系統常見的三種模型

3

- 將資料抽象化的過程需要有效地利用塑模而建成模型。
- 資料流程圖模型（**Data flow diagram, DFD**）。
  - 以處理為主體。
- 實體關係圖模型（**Entity relation diagram, ERD**）。
  - 以資料為主體。
- 統一塑模語言模型（**Unified modeling language, UML**）。
  - 以物件導向為主體。

## 關聯式資料庫管理系統規則建議

4

- **1970**年代，**Codd**首先發表關聯式資料庫模型，建議關聯式資料庫管理系統的十二項設計規則。
  - 資訊規則。
  - 保證存取規則。
  - 空值處理。
  - 關聯模組的動態線上目錄。
  - 提供通用的子語言規則。
  - 視域更新規則。
  - 高階**Insert**、**Delete**、**Update**。
  - 實體資料獨立性。
  - 邏輯資料獨立性。
  - 完整性獨立性。
  - 分散式獨立性。
  - 無破壞性規則。

## 關聯式模型及術語

5

- 關聯式模型是將資料庫由表格及其關聯組成的集合。
- 值組（**Tuple**）：資料表中每一列的資料，稱之為值組，或稱之為資料錄（**Record**）、實例（**Instance**）。
- 在資料表中的值組沒有一定的排列順序。因此，以下二表格被視為相等。

姓名	身分證號碼	年紀	國文成績
李一一	R111111111	12	89
王心心	R222222222	19	66

姓名	身分證號碼	年紀	國文成績
王心心	R222222222	19	66
李一一	R111111111	12	89

6

- 屬性（**Attribute**）：資料表中的欄位，稱之為屬性。亦稱之為欄位（**Field**）或行（**Column**）。
- 定義域（**Domain**）：代表特定屬性的記錄資料，有一定的表示範圍，符合此範圍的資料為合法的資料，才能記錄。
- 關聯（**Relation**）：資料以資料表（**Table**）來呈現，此資料表稱之為關聯或實體（**Entity**）。
- 關聯網要（**Relational schema**）：資料表與資料表間有相通的屬性欄位，透過此欄位兩個資料表就有所關聯關係。而描述出關聯關係的結構，稱之為關聯網要。

7

- 主鍵 (**Primary key**) : 由一個以上的屬性所組成。對於值組有唯一性。因此，有主鍵的資料表值組不能重複。
- 外部鍵 (**Foreign key**) : 由一個以上的屬性所組成。指向其它資料表中的主鍵，其值只能是所參考的另一個資料表之主鍵欄位值，可用來確定資料的參考完整性。

8

- 使用關聯式模組為基礎的資料庫管理系統，稱之為「關聯式資料庫管理系統」。
- 商業軟體：
  - Oracle、MS SQL Server、MS Access、...
- 開放原始碼軟體：
  - My SQL、PostgreSQL、...

## 認識關聯

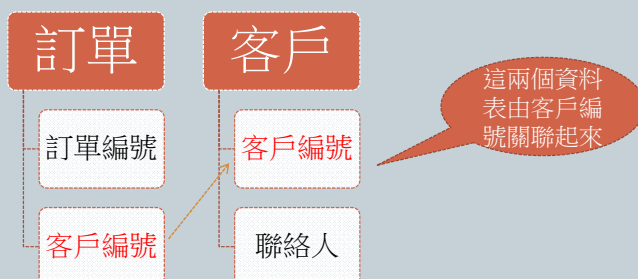
9

- 資料庫中每一資料表即是一種關聯。不同的資料表間，需建立資料表間的關聯，此架構是為關聯網要。
- 關聯之間的建立主要是利用外部鍵與主鍵來達成。
  - 注意：關聯式資料庫中，主鍵是每一資料表中最重要的一項，此處不可為空值，而為了保持資料的完整性，亦不可刪除。因為資料表會互相參考，需靠主鍵來找。

## 關聯

10

- 例如：下圖的訂單資料表與客戶資料表，便因為客戶編號欄位而產生關聯：



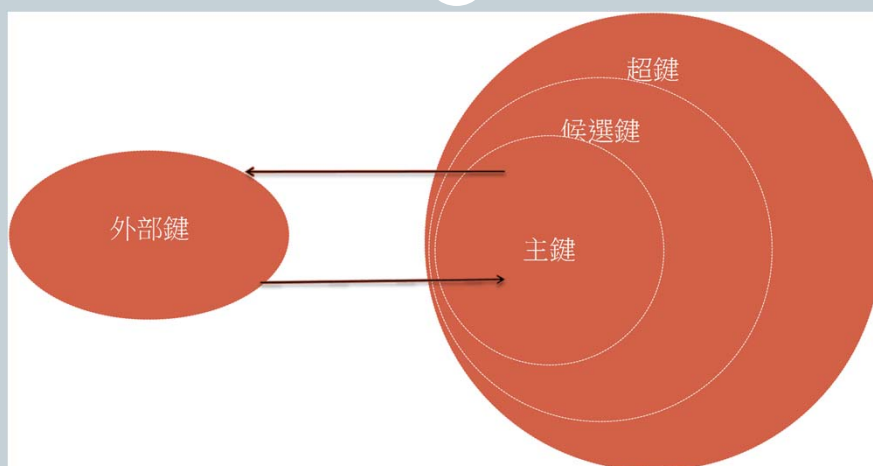
## 不同鍵的關係 ( 1 )

11

- 超鍵 ( **Super key** ) : 在一資料表中, 由一個以上的屬性所組成的集合。若此集合使其值組有唯一性, 稱此集合為「超鍵」。
- 候選鍵 ( **Candidate key** ) : 在一資料表中, 為所有超鍵中的屬性值最少的最短鍵, 具有成為主鍵資格的鍵。
- 主鍵 ( **Primary key** ) : 在一資料表中, 由使用者從候選鍵中指定其中一候選鍵成為最主要的主鍵。
- 外來鍵 ( **Foreign key** ) : 用來達成資料表間的參考關係。由子關聯的鍵值 (外部鍵) 對應到父關聯的候選鍵或主鍵。子關聯外部鍵的值域與父關聯候選鍵或主鍵的值域相同。

## 不同鍵的關係 ( 2 )

12



## 資料的完整性

13

- 資料的完整性 (**Data Integrity**) 是用來確保資料的可靠性。避免輸入無用的資料，造成錯誤資訊，誤導決策。
- 關聯式資料庫中的完整性限制有：
  - 值域完整性限制 (**Domain integrity constraint**)：值域完整性限制是指每一個屬性值，皆被限制在特定的範圍或資料型態中。
  - 鍵值完整性限制 (**Key integrity constraint**)：在一關聯中，鍵值不可為空值 (**Null value**)，因在關聯中是以鍵值做為分辨資料的唯一性。
  - 實體完整性限制 (**Entity integrity constraint**)：在同一關聯中，更值組具為唯一性，亦即，不可有相同重複的鍵值屬性值。
  - 參考完整性限制 (**Referential integrity constraint**)：用於關聯與關聯間的關聯性限制，維護兩關聯值組的一致性。
  - 使用者自訂完整性限制 (**User-defined integrity constraint**)：依照使用者的需求，自行限制資料的輸入，以確保資料的完整性。

## 資料表的關聯基數對應種類

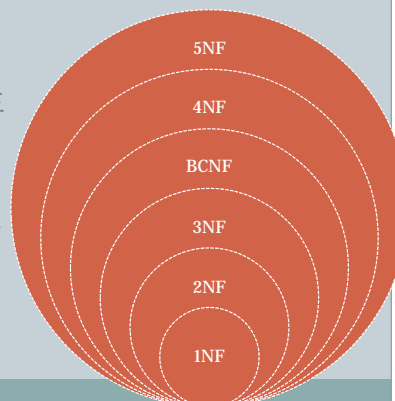
14

- 1 : 1，一對一對應關係。
- 1 : N，一對多對應關係。
- M : 1，多對一對應關係。
- M : N，多對多對應關係。

## 資料庫的正規化

15

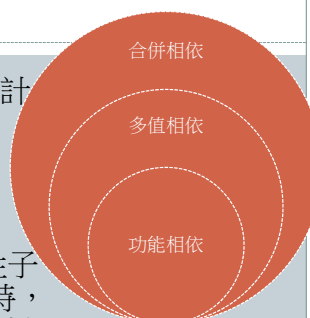
- 資料表正規化的目的：
  - 減少重複性、減少儲存空間。
  - 避免發生資料庫新增、刪除和更新的異常情況。
  - 易讀易擴充。
  - 簡化完整性限制的實行。
- 正規化處理的過程：
  - 第一正規化形式 → 第二正規化形式 → 第三正規化形式 → BCNF 正規化形式 → 第四正規化形式 → 第五正規化形式。
- 低階正規化：
  - 第一正規化形式、第二正規化形式、第三正規化形式、BCNF 正規化形式。
- 高階正規化：
  - 第四正規化形式、第五正規化形式。



## 相依性

16

- 相依性（Dependency）：用來分析關聯設計是否良好。
  - 功能相依性（Functional dependency）。
  - 多值相依性（Multivalued dependency）。
  - 合併相依性（Join dependency）。
- 功能相依性：令關聯為  $R$ ， $X$ 、 $Y$  為  $R$  的屬性子集合。對於  $R$  中所有的  $X$  值可唯一決定  $Y$  值時，則「 $Y$  功能相依於  $X$ 」或「 $X$  功能決定  $Y$ 」，以  $X \rightarrow Y$  表示。
  - 若  $X$  為候選鍵，則所有屬性  $Y$  功能相依於  $X$ 。
  - 此時， $X$  是為決定因素（Determinant）， $Y$  為相依因素（Dependent）。
  - 在  $R$  中，當  $X \rightarrow Y$ ，且  $X$  非候選鍵，必存在重複性的資訊。在資料庫操作中可能有潛在的問題。





## 最小功能相依性

17

- 在正規化前需求找出最小功能相依集合S。目的為：
  - 找出候選鍵及主鍵。
  - 正規化。
- 最小功能相依性（**Irreducible functional dependency**）：功能相依集合S符合以下三個條件，則稱S為最小功能相依集合。
  - 在S中的每一個功能相依FD，被功能決定的屬性亦即相依因素，只能有一個屬性。
  - 在S中沒有重複或多餘的功能相依。
  - 在S中的每一個功能相依FD，功能決定的屬性亦即決定因素的屬性，不可移除。
- 找出最小功能相依集合的方法：
  - 分解法。
  - 化簡功能決定屬性數目。
  - 去除多餘的功能決定屬性。
  - 刪除重複及遞移的功能相依。
  - 刪除不必要的功能相依。

## 投影

18

- 投影（**Projection**）：為轉換關聯的方式，將一關聯的屬性分割成一些屬性的子集合，每個屬性子集合都包含原先關聯的主鍵。將每個屬性子集合都設定成一新的關聯。
- 投影後新的關聯能取得原來聯連的資訊，是為無損投影（**Lossless projection**、**lossless decomposition**）。無損投影的分解過程是可逆的，即可以由分解後的關聯組合回原先的關聯。
- 鍵屬性（**Key attribute**）：候選鍵的屬性子集合，稱為鍵屬性，否則為非鍵屬性。
- 完全功能相依性（**Full functional dependency**）：在一功能相依 $X \rightarrow Y$ 中，從決定因素X中移去任一屬性，造成功能相依不存在，則稱Y完全功能相依於X。否則為Y部份功能相依於X（**Partial functional dependency**）。
- 遞移相依性（**Transitive dependent**）：在一功能相依 $X \rightarrow Y$ 中，存在非鍵屬性子集合Z導至 $X \rightarrow Z$ 且 $Z \rightarrow Y$ 。則Y遞移相依於X。

## 低階正規化

19

- 第一正規化形式 (**First normal form, 1NF**) :
  - 關聯R的所有屬性其基礎值域只包含單一純量值。
- 第二正規化形式 (**Second normal form, 2NF**) :
  - 關聯R是1NF，且所有非鍵屬性，完全功能相依於主鍵。
- 第三正規化形式 (**Third normal form, 3NF**) :
  - 關聯R是2NF，且所有非鍵屬性，非遞移相依於主鍵。
- **BCNF**正規化形式 (**Boyce-Codd normal form, BCNF**) :
  - 關聯R唯一的決定因素為候選鍵。

## 高階正規化 ( 1 ) 第四正規化形式

20

- 多值相依性 (**Multivalued dependency, MVD**) :
  - 給定一關聯R，令X、Y、Z為R的屬性子集合。對於所有值組，給定一值組的(X, Z)值，其值組上的Y值只相依於X，不相依於Z。則Y多值相依於X。
  - 以 $X \twoheadrightarrow Y$ 表示。
  - 多值相依為功能相依的特例。多值相依一定是功能相依，但功能相依不一定是多值相依。
- 第四正規化形式 (**Fourth normal form, 4NF**) :
  - 在關聯R中，存在屬性子集合X、Y，滿足多值相依 $X \twoheadrightarrow Y$ ，則R的所有屬性皆功能相依於X。

## 高階正規化 ( 2 ) 第五正規化形式

21

- **N度分解 ( N-decomposition )**：一關聯R可投影成N個關聯，且所有N個投影後的關聯合併後成為關聯R。但關聯R無法投影成N-1個關聯且R為此N-1投影關聯的合併時，稱之為N度分解。最常見的分解為二度分解。
- **合併相依性 ( Join dependency, JD )**：
  - 給定一關聯R，X1、X2、...、Xn為R的任意子屬性集合。若關聯R可用此N個屬性子集合作N度分解，則關聯R滿足合併相依。
  - 以\*{X1, X2, ..., Xn}表示。
  - 多值相依為合併相依的特例。合併相依一定為多值相依，多值相依不一定是合併相依。
- **第五正規化形式 ( Fifth normal form, 5NF )**：
  - 關聯R隱含候選鍵的合併相依。即關聯的每一子屬性集合分解，都要包含同一個候選鍵。

## 各種正規化的目的

22

- 第一正規化：去除多值屬性。
- 第二正規化：去除非鍵屬性的部份相依。
- 第三正規化：去除所有非鍵屬性的遞移功能相依。
- **BCNF**正規化：去除所有決定因素不是候選鍵的功能相依。
- 第四正規化：去除不是功能相依的多值相依。
- 第五正規化：去除不是候選鍵的合併相依。
- 較高階的正規化較可避免資料變動時的異常。但是，較高階的正規化需要較多的外部鍵參考，過度的外部鍵參考將降低資料庫的效能。

## 正規化形式的判定

23

- 無正規化：有多值屬性。
- 第一正規化形式：非候選鍵存在功能相依。
- 第二正規化形式：在功能相依中，存在非鍵屬性的遞移相依。
- 第三正規化形式：存在非鍵屬性的決定因素。
- **BCNF**正規化形式：存在不是功能相依的多值相依。
- 第四正規化形式：存在三度分解以上的合併相依，即存在非鍵屬性的合併相依。
- 第五正規化形式：非以上情況。

## 關聯式代數

24

- 關聯式代數的目的：由給定的關聯中，透過代數運算，建立所要的關聯。
- 八種基本關聯代數運算子：
  - 限制（Restrict）、選擇（Select）：取得符合條件的值組。
  - 投影（Project）：刪除指定屬性之後所得到的關聯。
  - 乘積（Product）：兩個關聯的所有值組對結果。
  - 聯集（Union）：包含兩個關聯中的所有值組。
  - 交集（Intersection）：兩個關聯同時出現的值組。
  - 差集（Difference）：出現在第一個關聯，但不出現在第二個關聯的值組。
  - 合併（Join）：合併後的關聯為由合併前的兩個關聯各取一值組組合而成，其在共同屬性上有共同的值。
  - 除法（Divide）：第一個關聯為二元關聯，第二個關聯為一元關聯，經運算後得到二元關聯中某屬性的值組。

## 選擇

(25)

- 選擇符合條件的組。
- 右方上面表格是原始表格，右方下面表格是篩選後的表格。

屬性A	屬性B	屬性C	屬性D	屬性E
A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4

屬性A	屬性B	屬性C	屬性D	屬性E
A1	B1	C1	D1	E1
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4

## 投影

(26)

- 選擇符合條件的組。
- 右方上面表格是原始表格，右方下面表格是投影後的表格。

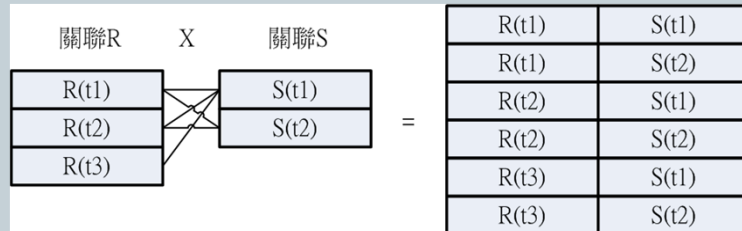
屬性A	屬性B	屬性C	屬性D	屬性E
A1	B1	C1	D1	E1
A2	B2	C2	D2	E2
A3	B3	C3	D3	E3
A4	B4	C4	D4	E4

屬性A	屬性C	屬性D	屬性E
A1	C1	D1	E1
A2	C2	D2	E2
A3	C3	D3	E3
A4	C4	D4	E4

## 乘積

27

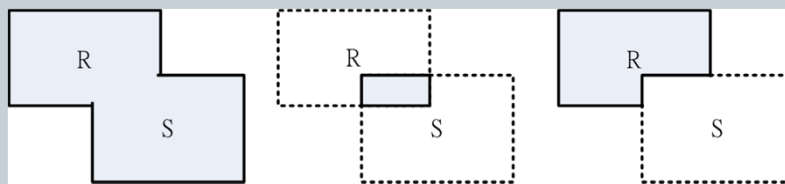
- 下圖為兩個關聯的乘積。



## 聯集、交集與差集

28

- 下圖由左至右分別是聯集、交集與差集。



## 合併

29

- 內部合併 (Inner join) :
- 外部合併 (Outer join) :
  - 左外部合併 (Left outer join) 。
  - 右外部合併 (Right outer join) 。
  - 完全外部合併 (Full outer join) 。

## 結構化查詢語言SQL

30

- SQL (Structured query language) 語言為關聯式資料庫的標準語言。
- SQL1 (SQL-86) 。
- SQL2 (SQL-92) 。
- SQL3 (SQL-99) 。
- SQL4 (SQL-2003) 。

## 本章回顧 ( 1 )

31

- 資料庫管理系統常見的三種模型：
  - 資料流程圖模型 (Data flow diagram, DFD)。
  - 實體關係圖模型 (Entity relation diagram, ERD)。
  - 統一塑模語言模型 (Unified modeling language, UML)。
- 資料庫管理系統的十二項設計規則。
  - 資訊規則。
  - 保證存取規則。
  - 空值處理。
  - 關聯模組的動態線上目錄。
  - 提供通用的子語言規則。
  - 視域更新規則。
  - 高階Insert、Delete、Update。
  - 實體資料獨立性。
  - 邏輯資料獨立性。
  - 完整性獨立性。
  - 分散式獨立性。
  - 無破壞性規則。

## 本章回顧 ( 2 )

32

- 值組 (Tuple)：資料表中每一列的資料，稱之為值組，或稱之為資料錄 (Record)、實例 (Instance)。
- 屬性 (Attribute)：資料表中的欄位，稱之為屬性。亦稱之為欄位 (Field) 或行 (Column)。
- 定義域 (Domain)：代表特定屬性的記錄資料，有一定的表示範圍，符合此範圍的資料為合法的資料，才能記錄。
- 關聯 (Relation)：資料以資料表 (Table) 來呈現，此資料表稱之為關聯或實體 (Entity)。
- 關聯網要 (Relational schema)：資料表與資料表間有相通的屬性欄位，透過此欄位兩個資料表就有所關聯關係。而描述出關聯關係的結構，稱之為關聯網要。



## 本章回顧 ( 3 )

33

- 關聯式資料庫中的完整性限制：
  - 值域完整性限制 ( Domain integrity constraint ) 。
  - 鍵值完整性限制 ( Key integrity constraint ) 。
  - 實體完整性限制 ( Entity integrity constraint ) 。
  - 參考完整性限制 ( Referential integrity constraint ) 。
  - 使用者自訂完整性限制 ( User-defined integrity constraint ) 。
- 資料表正規化的目的：
  - 減少重複性、減少儲存空間。
  - 避免發生資料庫新增、刪除和更新的異常情況。
  - 易讀易擴充。
  - 簡化完整性限制的實行。
- 正規化處理的過程：
  - 第一正規化形式 → 第二正規化形式 → 第三正規化形式 → BCNF 正規化形式 → 第四正規化形式 → 第五正規化形式。

## 本章回顧 ( 4 )

34

- 相依性 ( Dependency )：用來分析關聯設計是否良好。
  - 功能相依性 ( Functional dependency ) 。
  - 多值相依性 ( Multivalued dependency ) 。
  - 合併相依性 ( Join dependency ) 。
- 功能相依性：令關聯為  $R$ ， $X$ 、 $Y$  為  $R$  的屬性子集合。對於  $R$  中所有的  $X$  值可唯一決定  $Y$  值時，則「 $Y$  功能相依於  $X$ 」或「 $X$  功能決定  $Y$ 」，以  $X \rightarrow Y$  表示。
- 在正規化前需求找出最小功能相依集合  $S$ 。目的為：
  - 找出候選鍵及主鍵。
  - 正規化。
- 完全功能相依性 ( Full functional dependency )：在一功能相依  $X \rightarrow Y$  中，從決定因素  $X$  中移去任一屬性，造成功能相依不存在，則稱  $Y$  完全功能相依於  $X$ 。否則為  $Y$  部份功能相依於  $X$  ( Partial functional dependency ) 。
- 遞移相依性 ( Transitive dependent )：在一功能相依  $X \rightarrow Y$  中，存在非鍵屬性子集合  $Z$  導至  $X \rightarrow Z$  且  $Z \rightarrow Y$ 。則  $Y$  遞移相依於  $X$ 。

## 本章回顧 ( 5 )

35

- 低階正規化形式：
  - 第一正規化形式 (First normal form, 1NF)。
  - 第二正規化形式 (Second normal form, 2NF)。
  - 第三正規化形式 (Third normal form, 3NF)。
  - BCNF正規化形式 (Boyce-Codd normal form, BCNF)。
- 高階正規化形式：
  - 第四正規化形式 (Fourth normal form, 4NF)。
  - 第五正規化形式 (Fifth normal form, 5NF)。
- 各種正規化的目的：
  - 第一正規化：去除多值屬性。
  - 第二正規化：去除非鍵屬性的部份相依。
  - 第三正規化：去除所有非鍵屬性的遞移功能相依。
  - BCNF正規化：去除所有決定因素不是候選鍵的功能相依。
  - 第四正規化：去除不是功能相依的多值相依。
  - 第五正規化：去除不是候選鍵的合併相依。

## 本章回顧 ( 6 )

36

- 正規化形式的判定：
  - 無正規化：有多值屬性。
  - 第一正規化形式：非候選鍵存在功能相依。
  - 第二正規化形式：在功能相依中，存在非鍵屬性的遞移相依。
  - 第三正規化形式：存在非鍵屬性的決定因素。
  - BCNF正規化形式：存在不是功能相依的多值相依。
  - 第四正規化形式：存在三度分解以上的合併相依，即存在非鍵屬性的合併相依。
  - 第五正規化形式：非以上情況。
- 關聯式代數的目的：由給定的關聯中，透過代數運算，建立所要的關聯。
- 八種基本關聯代數運算子：
  - 限制 (Restrict)、選擇 (Select)：取得符合條件的值組。
  - 投影 (Project)：刪除指定屬性之後所得到的關聯。
  - 乘積 (Product)：兩個關聯的所有值組對結果。
  - 聯集 (Union)：包含兩個關聯中的所有值組。
  - 交集 (Intersection)：兩個關聯同時出現的值組。
  - 差集 (Difference)：出現在第一個關聯，但不出現在第二個關聯的值組。
  - 合併 (Join)：合併後的關聯為由合併前的兩個關聯各取一值組組合而成，其在共同屬性上有共同的值。
  - 除法 (Divide)：第一個關聯為二元關聯，第二個關聯為一元關聯，經運算後得到二元關聯中某屬性的值組。

## 參考文獻 ( 1/2 )

37

• **Fundamentals of Database Systems**

資料庫系統概論

- Ramez Elmasri & Shamkant B. Navathe
- Pearson
- Brief Edition

• **Database Design, Application Development & Administration** 資料庫管理

- Michael V. Mannino
- Mc Graw Hill
- Third Edition

## 參考文獻 ( 2/2 )

38

- 陳祥輝。資料庫系統理論與實務。旗標。台北市。2007。
- 曾守正。資料庫理論與實務。第3版。華泰文化。台北市。2010。
- 桂思強。資料庫理論與實務。旗標。台北市。2012。