



最新

PHP + MySQL + AJAX 網頁程式設計

第 8 章

MySQL 資料庫入門

本著作僅授權老師於課堂使用，切勿置放在網路上播放或供人下載，除此之外，未經授權不得將全部或局部內容以任何形式重製、轉載、變更、散佈或以其他任何形式、基於任何目的加以利用。

著作權所有 © 旗標出版股份有限公司

本章重點

- 8 - 1 資料庫基礎
- 8 - 2 資料庫的正規化分析
- 8 - 3 建立資料庫與資料表
- 8 - 4 MySQL 資料庫的存取權限

MySQL 資料庫入門

- 在 PHP 動態網頁的應用中，有相當多情況需要儲存資料。
- 例如帳號密碼、產品規格...等，此時除了將資料儲存在檔案之外，為了提高讀寫的效率與程式設計上的簡便，我們會採用資料庫系統，將資料的讀寫都交由資料庫系統來處理。
- 本章會先講解資料庫系統的基本觀念，然後逐步說明 MySQL 的使用方式。

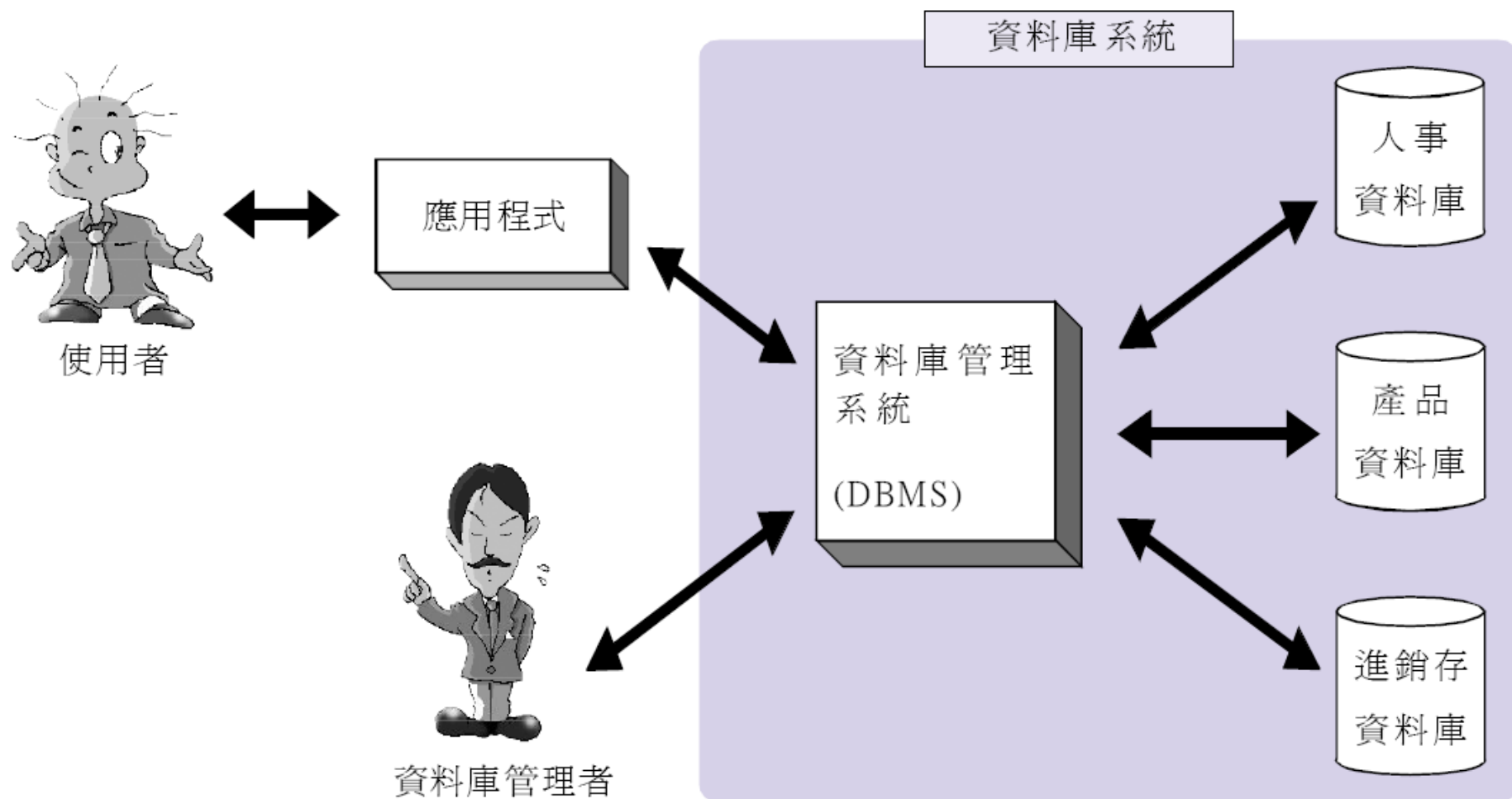
8 - 1 資料庫基礎

- 資料庫系統簡介
- 關聯式資料庫
- 關聯式資料庫的優點
- 資料表的 Primary key 與 Foreign key
- 資料型別

資料庫系統簡介

- **資料庫系統** (Database System) 是電腦化的資料儲存系統, 使用者則透過各種應用程式來存取其中的資料。
- 資料庫系統又可分為兩個部份：**資料庫** (Database) 與 **資料庫管理系統** (DataBase Management System, DBMS)。

資料庫系統簡介



資料庫系統簡介

- **資料庫**是儲存資料的地方。一個**資料庫系統**中可以有多個**資料庫**，每個**資料庫**都是一組經過整理好的資料集合。
- 一般，我們會將**資料庫**想像成是一個存放資料的容器，但**資料庫**的真實型態其實是一個個的電子檔案 (file)。
- **資料庫管理系統**則是指管理資料庫的軟體，它們負責使用者與資料庫之間的溝通，如存取資料庫中的資料、以及管理資料庫的各項事務等。

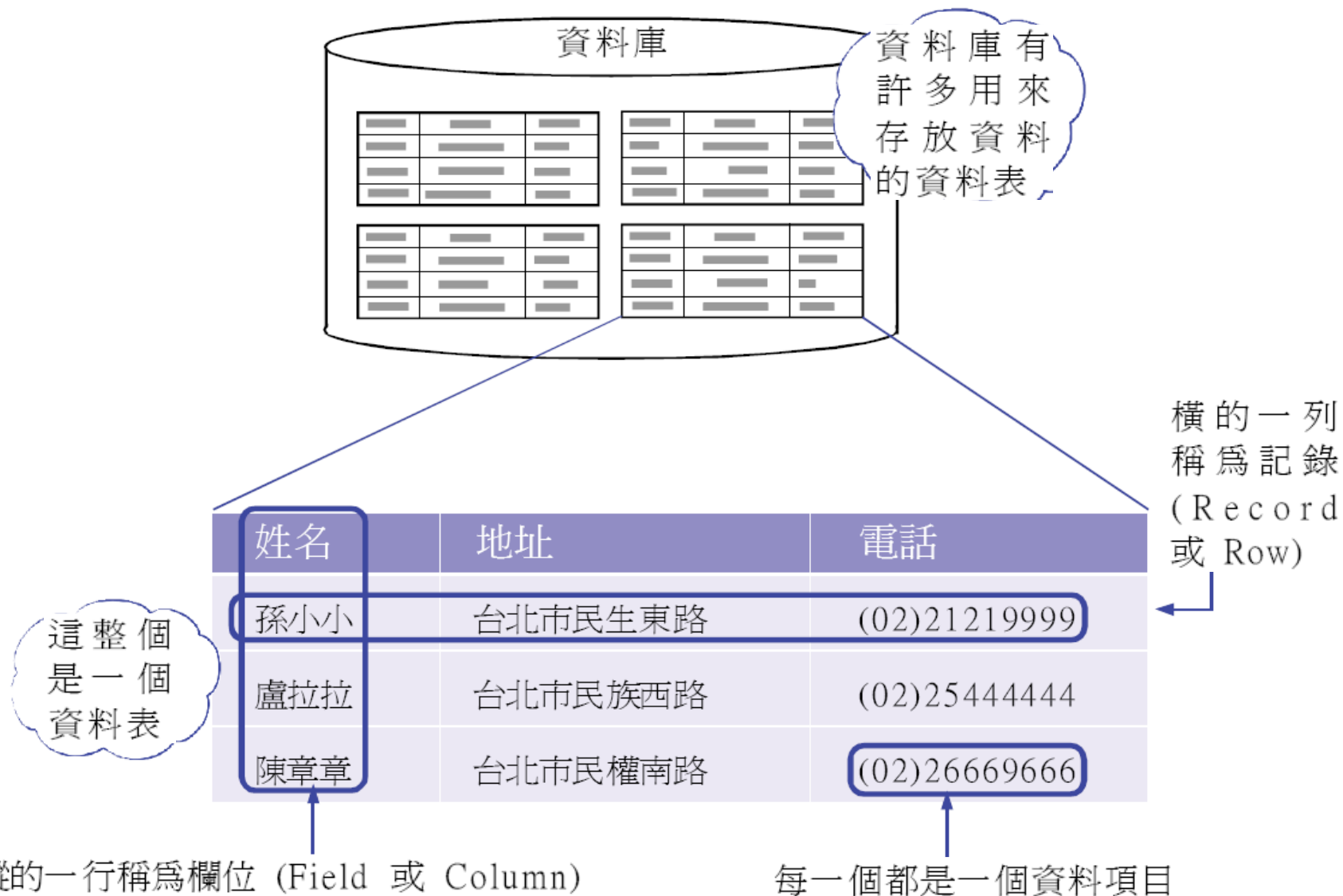
資料庫系統簡介

- 如 Microsoft 的 Access, 還有許多用在大型資料庫系統上的 Microsoft SQL Server、Oracle、SyBase、Informix、MySQL、PostgreSQL ... 等皆是**資料庫管理系統**。
- 稍後, 我們會對 MySQL 做較詳細的介紹。

關聯式資料庫

- 資料庫依照其儲存架構的不同，可以分為許多種類，較常用的有**階層式**、**網狀式**、**關聯式**、**物件導向式**等 4 種。
- 其中關聯式資料庫是目前最廣泛使用的種類，本書介紹的 MySQL 也是屬於關聯式資料庫。
- 關聯式資料庫是由**資料表** (Table) 所組成，資料會分類儲存在各個資料表中。

關聯式資料庫



關聯式資料庫

- 假如我們要從以上的資料表尋找 "盧拉拉" 的地址, 則是由橫向的『盧拉拉』與縱向的『地址』, 交相關聯而得來:

姓名	地址	電話
孫小小	台北市民生東路	(02)21219999
盧拉拉	台北市民族西路	(02)25444444
陳章章	台北市民權東路	(02)26669666

- 除了儲存在資料表行與列會有所關聯, 關聯式資料庫裡面的資料表之間通常也會互有關聯。

關聯式資料庫

- 透過資料表的關聯，我們可以從一個資料表中的欄位，找到另一個資料表中相關的資料：

訂單序號	日期	客戶編號	是否付款
1	2005/7/1	6	1
②	2005/7/1	③	1
3	2005/7/3	2	0

客戶資料表

經由客戶編號欄的關聯，可知道
訂單序號 2 的客戶為好看書店

客戶編號	客戶名稱	聯絡人	性別	地址
1	十全書店	陳圓圓	女	台北市
2	大發書店	陳季暄	女	台北市
③	好看書店	趙飛燕	女	台中市

訂單資料表

關聯式資料庫的優點

- 從上面**訂單**與**客戶**資料表的例子中，可以看到現實生活中一筆資料存入關聯式資料庫時，可能會被分割放進多個資料表。
- 雖然說將所有資料都放在一個資料表中，資料庫一樣可以運作，不過如此便失去了關聯式資料庫的優點。
- 關聯式資料庫有什麼優點呢？下面將為您說明。

節省儲存空間

- 因為資料庫中有相當多的資料會產生重複的情況，如果每一次都要輸入相同的資料，則容易浪費磁碟儲存的空間，例如：

書籍名稱	作者姓名	分類	價格
Windows 實用秘笈	施施研究室	Windows	450
Windows 網路通訊秘笈	施施研究室	Windows	480
Windows 系統秘笈	施施研究室	Windows	490
PhotoShop 特效魔術師	施施研究室	影像處理	490
抓住你的 PhotoShop	施施研究室	影像處理	580

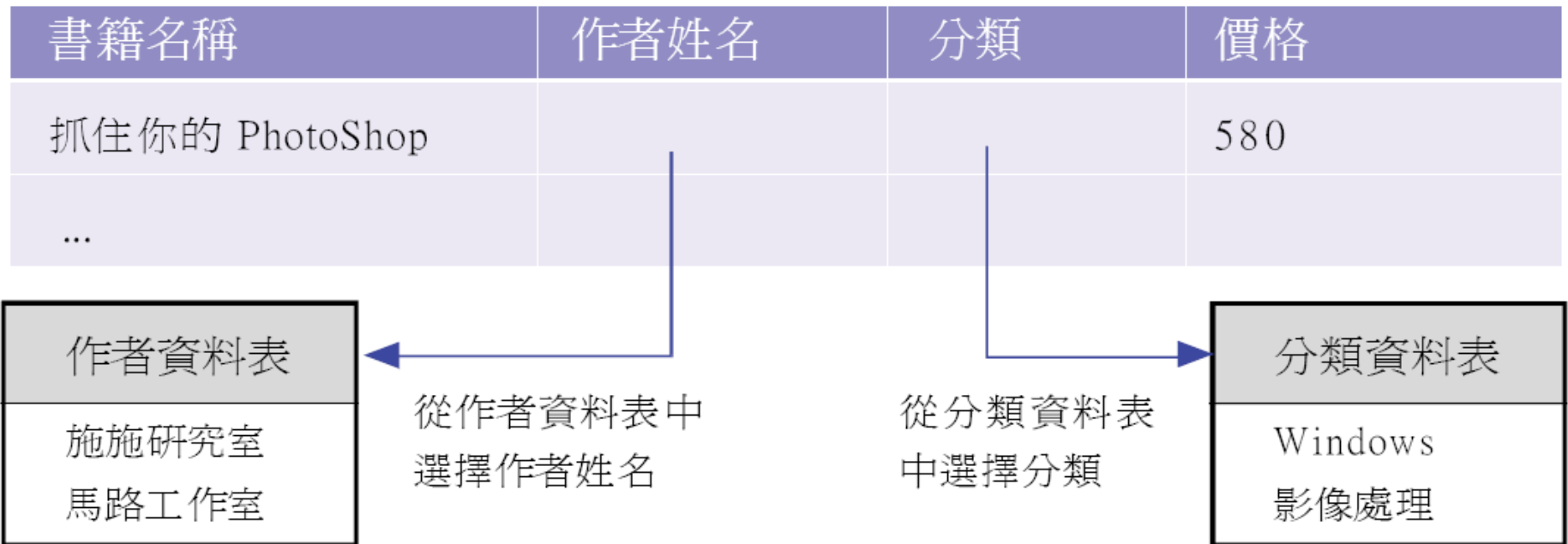
節省儲存空間

- 很明顯地，在**作者姓名**欄位與**分類**欄位中，有相當多的資料是重複的，例如 "施施研究室" 輸入 5 次、"Windows" 輸入 3 次，而 "影像處理" 則輸入 2 次。
- 如果一個資料庫中有數千或上萬筆記錄，這些重複輸入的資料所造成的磁碟空間浪費就很可怕了。
- 若是我們將**作者姓名**與**分類**欄位抽離，另外獨立成**作者資料表**與**分類資料表**，並建立這 3 個資料表間的**關聯**。

節省儲存空間

- 那麼在**作者資料表**中，各作者名稱只需記錄一次，而在**分類資料表**中每種分類也只需記錄一次即可。
- 當**書籍資料表**需要使用到作者名稱或分類時，則可以經由**關聯**，到**作者資料表**與**分類資料表**中選取。

節省儲存空間



- 想想看，若是原本有 500 本書的作者姓名是 "施施研究室"，那就要在資料表中記錄 500 次的 "施施研究室"。

節省儲存空間

- 而分割資料表並建立關聯之後，只要在**作者資料表**中記錄一次"施施研究室"就好了，這對於空間的節省不可謂不大呀！

減少輸入錯誤

- 同樣的資料經常重複輸入時，難免會漏打或是打錯字，使得原本應該是相同的資料，卻變成 2 筆不同的資料：

書籍名稱	作者姓名	分 類	價格
Windows 實用秘笈	施施研究室	Windows	450
Windows 網路通訊秘笈	施施研究室	Windows	480
Windows 系統秘笈	施施研究示	Windows	490
...			

輸入錯誤

減少輸入錯誤

- 以後當用 "施施研究室" 字串來查詢資料時，**Windows 系統秘笈** 這一本恐怕就查不到了。
- 然而若是使用關聯式資料庫，則**作者姓名**這一欄的資料實際上是來自於**作者資料表**。
- 因此只要確定**作者資料表**中的 "施施研究室" 這筆記錄是正確的，就不需要重複輸入 "施施研究室"，自然就減少輸入錯誤的機會。

方便資料修改

- "方便資料修改" 也是分割資料表一個重要的優點！
- 如果有一天要將 "施施研究室" 改為 "旗旗研究室", 在沒有分割資料表的狀況下需要一筆一筆記錄去修改, 相當耗費時間與精力。
- 若有適當的分割資料表, 則只要將**作者資料表**的 "施施研究室" 改為 "旗旗研究室" 即可。

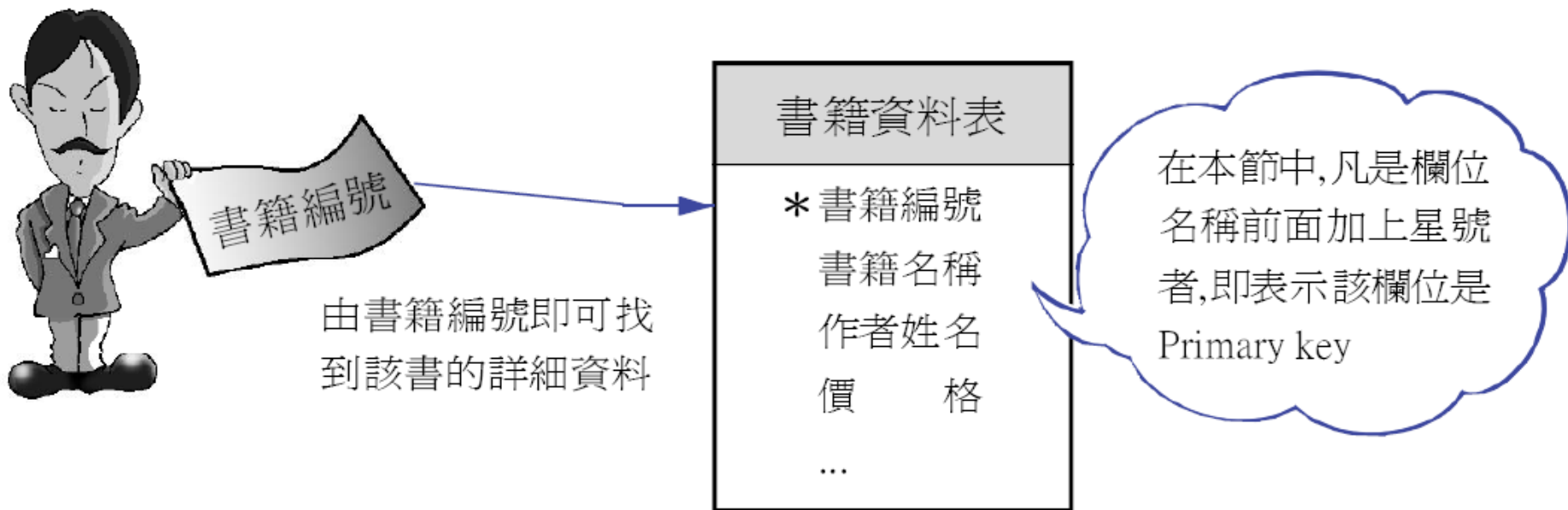
資料表的 Primary key 與 Foreign key

- 資料表之間的關聯是由所謂的鍵 (Key) 來建立的。
- Key 可分為兩種：一種是 Primary key, 另一種是 Foreign key, 以下分別說明。

Primary key

- Primary key 是用來辨識記錄的欄位，具有唯一性，且不允許重複。
- 例如在**書籍資料表**中加入**書籍編號**欄位，給每本書一個唯一的編號。
- 那麼這個**書籍編號**欄位就可用來當作 Primary key，使用者即可依據此 Primary key (主鍵) 找到某特定書籍的詳細記錄。

Primary key



- 雖然資料表不一定要有 Primary key, 但一般都建議最好要有。
- 不過資料表中並不是每一個欄位都適合當做 Primary key。

Primary key

- 例如書籍資料表的作者姓名，因為可能會遇到同名同姓的人，所以就不具有唯一性了，當然也就不適合做 Primary key。
- 通常每個資料表只有一個欄位設定為 Primary key，但有時候可能沒有一個欄位具有唯一性，此時可以考慮使用兩個或多個欄位組合起來做為 Primary key。
- 請看下面的範例。

Primary key

訂購者編號	訂單編號	書籍名稱	數量	單價
100011	1	COOL3D 使用手冊	100	390
100011	2	抓住你的 PhotoImpact	200	390
100011	3	Linux 實務應用	150	620
100200	1	Windows Server 架站實務	80	450
100200	2	BIOS 玩家實戰	80	299
...				

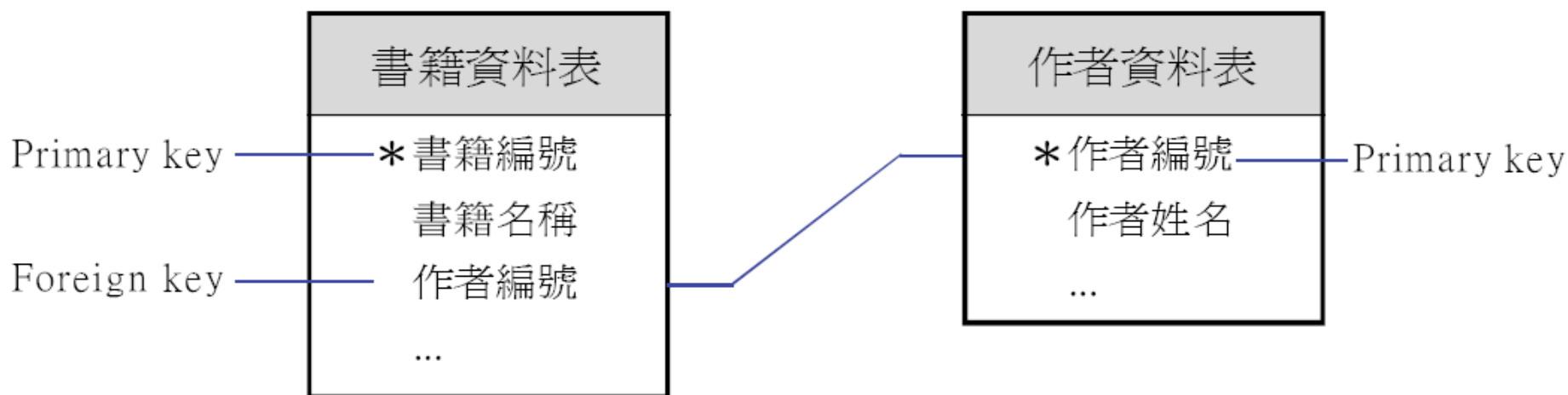
- 上例中好幾筆記錄具有相同的**訂購者編號**或**訂單編號**，使得沒有一個欄位具有唯一性，因此找不到一個單獨的欄位來當 Primary key。

Primary key

- 其實這可說是資料表設計的問題, 若我們在設計欄位時, 不管由哪一個訂購者所下的每一筆訂單, 都給一個唯一的編號時, 就可以用**訂單編號**欄位來做 Primary key 了。
- 但在不修改資料表的設計的狀況下, 我們發現, 其實將**訂購者編號**與**訂單編號**這兩個欄位組合起來也具有唯一性 — 因為同一個訂購者不會有 2 個相同的訂單號碼。
- 因此我們可以將這 2 個欄位同時設為 Primary key, 那麼就具有辨識唯一一筆記錄的特性了。

Foreign key

- 在關聯式資料庫中，資料表之間的關係是藉由 Foreign key 來建立的，例如：



- 書籍編號與作者編號欄位，分別是書籍資料表與作者資料表的 Primary key。

Foreign key

- 為了建立兩資料表之間的關聯，則在書籍資料表需要有一個欄位參考或對應到作者資料表的 Primary key, 所以便在書籍資料表中設置了作者編號欄位，此欄位便是 Foreign key。
- 並不是每一種關聯式資料庫都支援 Foreign Key, 例如本書介紹的 MySQL, 其預設使用的 MyISAM 儲存引擎便無法建立 Foreign Key。
- 雖然無法建立，但是我們仍可自行在程式中依照其關聯來讀寫資料。

Primary Key 和 Foreign Key 的名稱一定要相同嗎？

- 前文所述書籍資料表與作者資料表中的作者編號欄位，前者是 Foreign key，後者是 Primary key。
- 這兩個欄位的資料型別、寬度等屬性必須相同，但名稱不一定要一樣，只是我們習慣上都會取相同的名稱。
- 另外，Foreign key 中的資料可以重複（例如多本書作者可能是同一人），這點和 Primary key 不同。

資料型別

- 前面提到資料庫的資料表是由欄位與記錄所組成，例如一個**書籍**資料表如下：

書籍編號	書籍名稱	價格	入庫日期
1	Linux 實務應用	500	2007/07/30
2	Windows 使用手冊	450	2007/07/07

- 因為資料類型的不同，上面 3 個欄位必須設定不同的**資料型別** (Data Type, phpMyAdmin 中翻譯為**型態**)：

書籍編號	書籍名稱	價格	入庫日期
數字	字串	數字	日期時間

資料型別

- 資料型別關係到該欄位要儲存的資料種類與未來的運作，必須謹慎設定。
- 例如**書籍名稱**如果錯設為數字型別，便無法寫入非數字的文字；或者**價格**欄位若錯設為字串型別，資料庫系統便無法依需求進行總和、平均、最大值...等計算。
- 本書介紹的 MySQL 目前共有 3 大類的資料型別：字串、數字、日期時間，以下摘要介紹各類別中常用的資料型別。

字串型別

- 下表中, n 表示可儲存的最大字元數 (稱為長度), L 表示實際儲存的字元數:

型別	佔用的空間大小	範圍
CHAR(n)	n 字元	0 ~ 255 字元
VARCHAR(n)	若 $n \leq 255$, 則為 L 字元 + 1 byte 若 $n > 255$, 則為 L 字元 + 2 bytes	0 ~ 65535 字元

- 使用字串型別時, 如果使用英美語系的編碼 (MySQL 中的編碼取決於字元集的設定), 則通常 1 個字元只佔用 1 bytes。

字串型別

- 但是如果使用 Big5、UTF8...等多位元語系, 1 個字元可能是 1 ~ 3 個 bytes。
- CHAR 具有固定長度的特性, 而 VARCHAR 的長度則會視儲存的字元數而變動。
- 舉例來說, 假設在字元型別的欄位中存入 3 個字元, 若該欄位設定為 CHAR (8), 仍然會佔用 8 個字元的空間, 如果是 VARCHAR (8), 則只會佔用 3 個字元 + 1 byte 空間。
- 不過, 因為 VARCHAR 是變動的長度, 所以存取的速度會比 CAHR 慢。

數字型別

- 數字型別可細分為整數型別與浮點數型別，分別用來存放整數與小數，以下將分別說明：

- **整數**：下表中，n 為顯示寬度：

型別	佔用的空間大小	範圍
TINYINT(n)	1 Bytes	-128~127 (SIGNED, 有正負號) 0~255 (UNSIGNED, 無正負號)
SMALLINT(n)	2 Bytes	-32768~32767 (SIGNED) 0~65535 (UNSIGNED)
MEDIUMINT(n)	3 Bytes	-8388608~8388607 (SIGNED) 0~16777215 (UNSIGNED)

數字型別

INT(n)	4 Bytes	-2147483648~2147483647 (SIGNED) 0~4294967295 (UNSIGNED)
BIGINT(n)	8 Bytes	-9223372036854775808~ 9223372036854775807 (SIGNED) 0~18446744073709551615 (UNSIGNED)

- 整數型別可以設定顯示寬度, 用來設定要顯示的位數。
- 例如設定 INT (3), 表示其顯示時的寬度至少為 3 位數, 如果位數不足則 MySQL 會自動補上空白, 所以存入 "1" 時會顯示 "__1" (_ 代表空白)、存入 "56" 時會顯示 "_56"。

數字型別

- 請注意, 顯示寬度與範圍大小並無相關, INT (3) 欄位依然可以存入 "12345" 這個值, 顯示時也會是 "12345"。
- **浮點數**：下表中, m、d 用來定義有效位數：

型別	佔用的空間大小	範圍
FLOAT(m,d)	4 Bytes	-3.402823466E+38~-1.175494351E-38, 0, 1.175494351E-38~3.402823466E+38
DOUBLE(m,d)	8 Bytes	-1.7976931348623157E+308 ~ -2.2250738585072014E-308,

數字型別

		0, 2.2250738585072014E-308 ~ 1.7976931348623157E+308
DECIMAL(m,d)	不一定	不一定(參見後面說明)

- 浮點數型別能夠使用 FLOAT (m, d)、DECIMAL (m, d) 的方式來定義有效位數, 其中 m 代表最大長度 (整數加小數), d 表示小數長度。
- 故 DECIMAL (5, 2) 可以儲存 XXX.dd 這種 2 位小數的 5 位數字。如果沒有定義, DECIMAL 預設使用 DECIMAL (10, 0)。

數字型別

- 請注意, **FLOAT** 內部運作是以近似值的方式來儲存數字, 所以 **FLOAT** 可能會在某些數字失去精確度。
- 而 **DECIMAL** 則是精確儲存數字, 所以如果用於財務、成績...等用途, 建議使用 **DECIMAL** 型別。
- 使用數字型別時, 若指定 **UNSIGNED** 屬性, 則表示無負數值。
- **UNSIGNED** 會影響整數型別的大小範圍, 但是對於浮點數則不影響範圍, 僅能限定該欄位使用正數。

數字型別

- 除了 UNSIGNED 屬性以外, 數字型別還可以設定 ZEROFILL 屬性, 可以在不足顯示寬度或有效位數的部分補上 0。
- 例如將 "123" 存入帶有 ZEROFILL 屬性的 INT (5) 欄位, 取出時會變成 "00123"。
- 將 "12.3" 存入帶有 ZEROFILL 屬性的 DECIMAL (7, 3) 欄位, 取出時會變成 "0012.300"。

日期時間型別

- 如果要儲存日期或時間，可以使用以下型別：

型別	佔用的空間大小	範圍
DATE	3 bytes	1000-01-01 ~ 9999-12-31
DATETIME	8 bytes	1000-01-01 00:00:00 ~ 9999-12-31 23:59:59
TIME	3 bytes	-838:59:59 ~ 838:59:59
YEAR	1 bytes	1901 ~ 2155

8 - 2 資料庫的正規化分析

- 關聯式資料庫不管設計得好壞，都可以儲存資料，但是存取效率上可能會有很大的差別。
- 想提升關聯式資料庫的效率，在設計資料庫的時候，可以利用**正規化** (Normalization) 的方法來協助我們修改資料表的結構。

何謂正規化

- **正規化**到底在做什麼？
- 其實簡單的說, **正規化**就是要讓資料庫中重複的資料減到最少, 讓我們能夠快速地找到所要的資料, 以提高關聯式資料庫的效能。
- E.F. Codd 博士的關聯式資料庫正規化理論, 將正規化的步驟歸納成幾個階段, 讓我們有具體可循的方法來建全資料表的結構。

何謂正規化

- 資料庫的正規化共可分為**第一階正規化** (1st Normal Form, 1NF)、**第二階正規化** (2NF)、**第三階正規化** (3NF)、**BCNF** (Boyce-Codd Normal Form)、**第四階正規化** (4NF)、**第五階正規化** (5NF) 等多個階段。
- 不過對於一般資料庫設計來說, 通常只要執行到第三階段正規化即可, 其他更高階的正規化只有在特殊的情況下才用得到。
- 因此本章也只介紹到第三階段正規化的正規化。

第一階正規化

- 正規化的過程是循序漸進的，資料表必須在滿足第一階正規化的條件之下，才能進行第二階正規化。
- 也就是說，第二階正規化必須建立在符合第一階正規化的資料表上，依此類推。
- 因此，第一階正規化是所有正規化的基礎。

第一階正規化的規則

- **第一階正規化** (1st Normal Form, 以下簡稱 1NF) 有以下幾個規則：
 - (1) 資料表中必須有 Primary Key, 而其他所有的欄位都『相依』於 Primary Key。
 - 『相依』是指一個資料表中, 若欄位 B 的值必須搭配欄位 A 才有意義, 就是「B 欄位相依於 A 欄位」。
 - 舉例來說, 某一**員工**資料表如下。

第一階正規化的規則

* 員工編號	姓名	地址
1032	孟庭訶	台北市杭州南路一段 15-1 號 19 樓
1039	楊咩咩	台北市杭州南路一段 15-1 號 19 樓

- 上表的**員工編號**欄為 Primary Key, 做為唯一辨識該筆記錄的欄位。
- 對此資料表來說, **地址**欄必須要相依於**員工編號**欄才有意義。
- 由於 "孟庭訶" 和 "楊咩咩" 的地址都相同, 所以不能以**地址**做為 Primary Key, 否則就無法從地址來區別是哪一個人了！

第一階正規化的規則

- 同樣地, **姓名**欄也必須相依於**員工編號**欄。
- (2) 每個欄位中都只儲存單一值, 例如同一筆記錄的**姓名**欄位中不能存放 2 個人的姓名。
- (3) 資料表中沒有意義相同的多個欄位, 例如**姓名 1**、**姓名 2**...等重複的欄位。
- 反之, 若資料表的欄位不符合以上規則, 則稱為『非正規化』的資料表。

建構 1NF 資料表的方法

- 首先我們來看一個非正規化的**訂單**資料表：

訂單編號	客戶名稱	員工編號	負責業務員	書號	書籍名稱	數量
OD101	十全書店	1032	孟庭訶	F5301	Linux 實務應用	20
				F5120	XOOPS 架站王	60
				F5662	威力導演	30
OD103	愛潤福量販店	1039	楊咩咩	F5662	威力導演	80

- 對於不具 1NF 形式的**訂單**資料表，我們可將重複的資料項分別儲存到不同的記錄中，並加上適當的 Primary Key (標示 * 符號者為 Primary Key)，產生如下的**訂單**資料表。

建構 1NF 資料表的方法

* 訂單編號	* 書號	員工編號	負責業務員	客戶名稱	書籍名稱	數量
OD101	F5301	1032	孟庭訶	十全書店	Linux 實務應用	20
OD101	F5120	1039	孟庭訶	十全書店	XOOPS 架站王	60
OD101	F5662	1039	孟庭訶	十全書店	威力導演	30
OD103	F5662	1039	楊咩咩	愛潤福量販店	威力導演	80

- 如此一來，雖然增加了許多筆記錄，但每一個欄位的長度及數目都可以固定。
- 而且我們可用**訂單編號**欄加上**書號**欄做為 Primary Key，那麼在查詢某本書的銷售數量時，就非常地方便快速了。

第二階正規化

- 將上述**訂單**資料表執行 1NF 之後，應該很容易察覺：我們輸入了許多重複的資料。
- 如此，不但浪費儲存的空間，更容易造成新增、刪除或更新資料時的異常狀況。
- 所以，我們必須接著進行第二階正規化，來消除這些問題。

第二階正規化的規則

- **第二階正規化** (2nd Normal Form, 以下簡稱 2NF) 有以下幾個規則：
 - (1) 必須符合 1NF 的格式。
 - (2) 各欄位與 Primary Key 間沒有『部分相依』的關係。
- 『部分相依』只有在 Primary Key 是由多個欄位組成時才會發生, 它是指某些欄位只與 Primary Key 中的部分欄位有相依性, 而與另一部分的欄位沒有相依性。

第二階正規化的規則

- 以前例的**訂單**資料表來說, 其 Primary Key 為**訂單編號** + **書號**欄位。
- 但**客戶名稱**欄只和**訂單編號**欄有相依性 (一筆訂單只對應一家客戶), 而**書籍名稱**欄只和**書號**欄有相依性 (一個書號只對應一本書):

書籍名稱只和書號有相依性

訂單資料表

訂單編號	書號	員工編號	負責業務員	客戶名稱	書籍名稱	數量
OD101	F5301	1032	孟庭訶	十全書店	Linux 實務應用	20
OD103	F5662	1039	楊咩咩	愛潤福量販店	威力導演	80

客戶名稱只和訂單編號有相依性

建構 2NF 資料表的方法

- 要除去資料表中的部分相依性, 只需將部份相依的欄位分割成另外的資料表即可。
- 例如我們將**訂單**資料表分割成 3 個較小的資料表 (標示 "*" 號的欄位為 Primary Key) :

出貨記錄表

* 訂單編號	* 書號	數量
OD101	F5301	20
OD101	F5120	60
OD101	F5662	30
OD103	F5662	80

建構 2NF 資料表的方法

書籍資料表

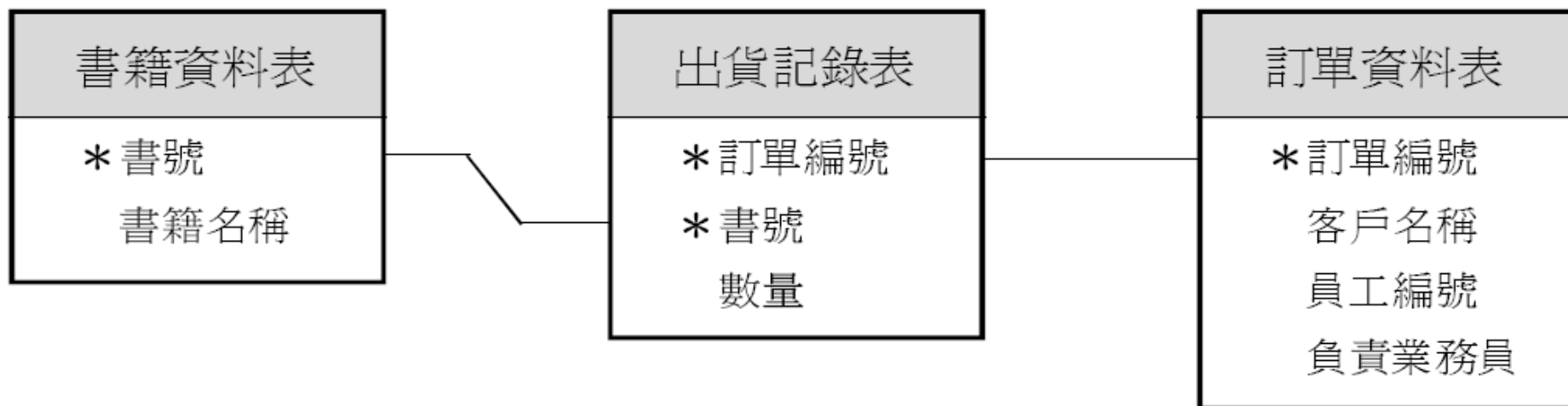
* 書號	書籍名稱
F5301	Linux 實務應用
F5120	XOOPS 架站王
F5662	威力導演

訂單資料表

* 訂單編號	客戶名稱	員工編號	負責業務員
OD101	十全書店	1032	孟庭訶
OD103	愛潤福量販店	1039	楊咩咩

建構 2NF 資料表的方法

- 分成 3 個資料表後, 便去除了原本資料表的 "部份相依性", 我們來看看資料表間的關聯可以更容易明白:



第三階正規化

- 經過 2NF 後的資料表，其實還存在一些問題：
 - 在**訂單**資料表中，如果有新進業務同仁 "陳圓圓"，在該員尚未安排負責客戶之前，我們無法輸入該員的資料。
 - 若要刪除**十全書店**的這家客戶，勢必會將負責該客戶的業務同仁 "孟庭訶" 一併刪除。
- 基於上述理由，我們必須再執行第三階正規化。

第三階正規化的規則

- **第三階正規化** (3rd Normal Form, 以下簡稱 3NF) 有以下幾個要件：
 - (1) 符合 2NF 的格式
 - (2) 各欄位與 Primary Key 間沒有『間接相依』的關係
- 『間接相依』是指二個欄位間並非直接相依，而是借助第三個欄位來達成資料相依的關係。
- 例如 A 相依於 B；而 B 又相依於 C，如此 A 與 C 之間就是間接相依的關係。

第三階正規化的規則

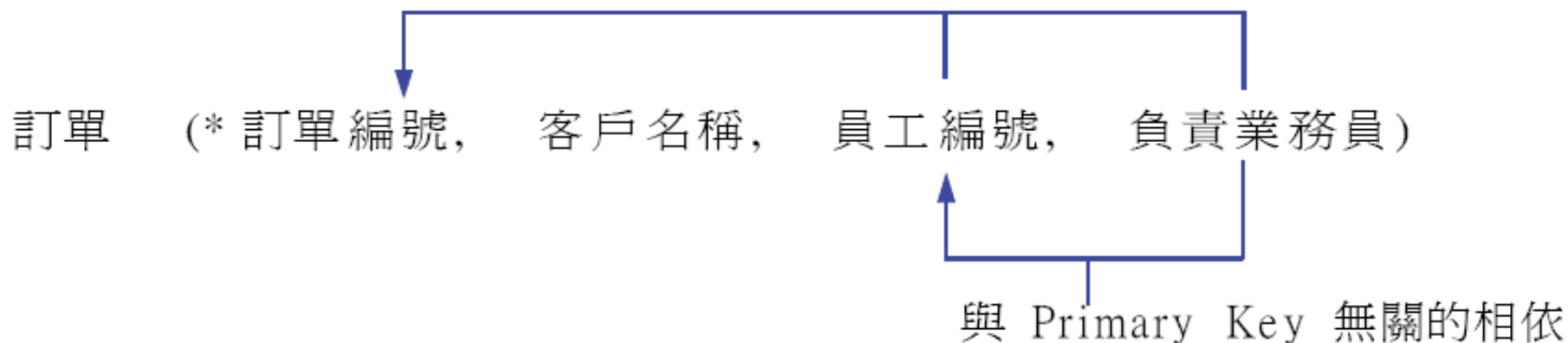
- 要找出各欄位與 Primary Key 間的間接相依性，最簡單的方式就是看看資料表中有沒有『與 Primary Key 無關的相依性』存在。
- 例如在前例**訂單**資料表中：

訂單資料表

* 訂單編號	客戶名稱	員工編號	負責業務員
OD101	十全書店	1032	孟庭訶
OD103	愛潤福量販店	1039	楊咩咩

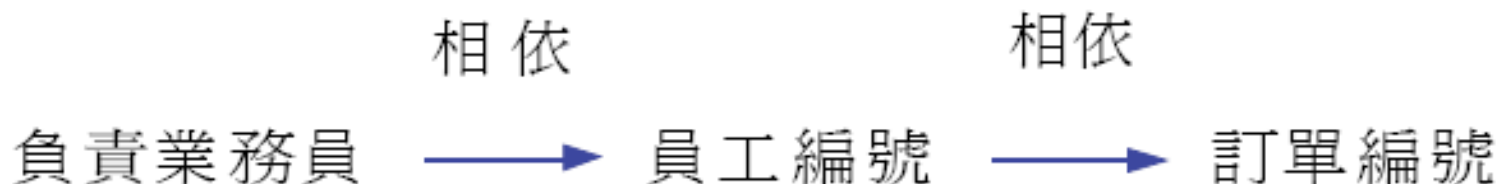
第三階正規化的規則

- 由於每筆訂單都會有一位業務員負責，所以 **員工編號** 欄和 **負責業務員** 欄都相依於 **訂單編號** 欄。
- 但 **負責業務員** 又同時相依於 **員工編號** 欄，而這個相依性是與 Primary Key 完全無關的：



第三階正規化的規則

- 事實上，它們之間的相依關係為：



- 由此可知，**負責業務員**與 Primary Key 存在著無關的相依性，也就是有『間接相依』的關係存在。

建構 3NF 資料表的方法

- 要除去資料表中的『間接相依性』，其方法和除去『部分相依性』完全相同。
- 例如**訂單**資料表可再分割成兩個資料表：

訂單資料表

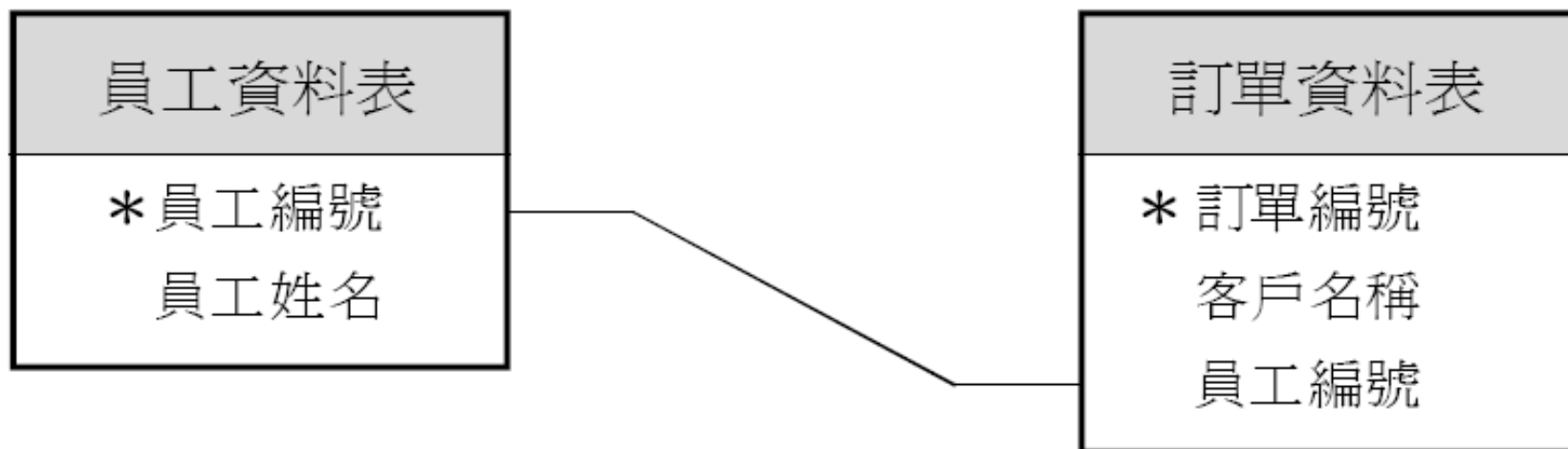
* 訂單編號	客戶名稱	負責員工編號
OD101	十全書店	1032
OD103	愛潤福量販店	1039

員工資料表

* 員工編號	員工姓名
1032	孟庭訶
1039	楊咩咩

建構 3NF 資料表的方法

- 我們來看看這 2 個資料表的關聯：



- 這樣負責業務員的『間接相依性』便被去除了。

與直覺式的分割技巧做比較

- 當您設計資料庫一段時間，累積了經驗及技術後，您便可依照自己的經驗，以直覺的方式對資料表執行最佳化。
- 底下是兩種方法在功能上的對照：

正規化	功能相同的直覺式分割法
1NF：有主鍵 欄位中只有一個單一值 沒有意義相同的重複欄位	無
2NF：除去『部分相依性』	分割『欄位值一再重複』的欄位
3NF：除去『間接相依性』	分割『與主鍵無關』的欄位

8 - 3 建立資料庫與資料表

- 瞭解了資料庫的基本觀念後, 就可以開始學習如何在 MySQL 中建立資料庫與資料表。
 - MySQL 資料庫的字元集與校對
 - 建立資料庫
 - 建立資料表
 - 新增、瀏覽、編輯與刪除資料
 - 備份與回復資料庫

MySQL 資料庫的字元集與校對

■ 認識字元集與校對

- **字元集** (character set) 指的是文字的編碼方式，而**校對** (collation) 則是字元資料的排序方式。
- 這兩個設定關係到要儲存的文字。
- 例如要存放繁體中文的字串時，應該使用 UTF8 或 Big5 編碼，如果不小心使用日文編碼儲存資料，日後以 UTF8 或 Big5 編碼讀取資料庫時便會導致亂碼。

MySQL 資料庫的字元集與校對

- 所以在建立資料庫與資料表之前，應該先瞭解 MySQL 資料庫的字元集與校對方式，才能在建立時指定正確的編碼設定。
- 目前 MySQL 支援相當多種語文的編碼，請依照 1 - 5 節的說明登入 MySQL 的管理程式 — phpMyAdmin，在首頁如下操作即可觀看 MySQL 支援的所有字元集與校對。

MySQL 資料庫的字元集與校對

如果目前位置不在首頁，
請按此鈕即可回到首頁

2 點此連結



1 向下拉動滑動桿

MySQL 資料庫的字元集與校對

各字元集下有不同的校對

這些是字元集

校對	說明
armSCII8 (ARMSCII-8 Armenian)	
armSCII8_bin	美式英語, 二進制碼
armSCII8_general_ci	美式英語, 大小寫不相符
ascii (US ASCII)	
ascii_bin	西歐語文 (多語言), 二進制碼
ascii_general_ci	西歐語文 (多語言), 大小寫不相符
big5 (Big5 Traditional Chinese)	
big5_bin	繁體中文, 二進制碼
big5_chinese_ci	繁體中文, 大小寫不相符
binary (Binary pseudo charset)	
binary	二進制碼
cp1250 (Windows Central European)	

校對的名稱是以其所屬的字元集做為開頭

MySQL 資料庫的字元集與校對

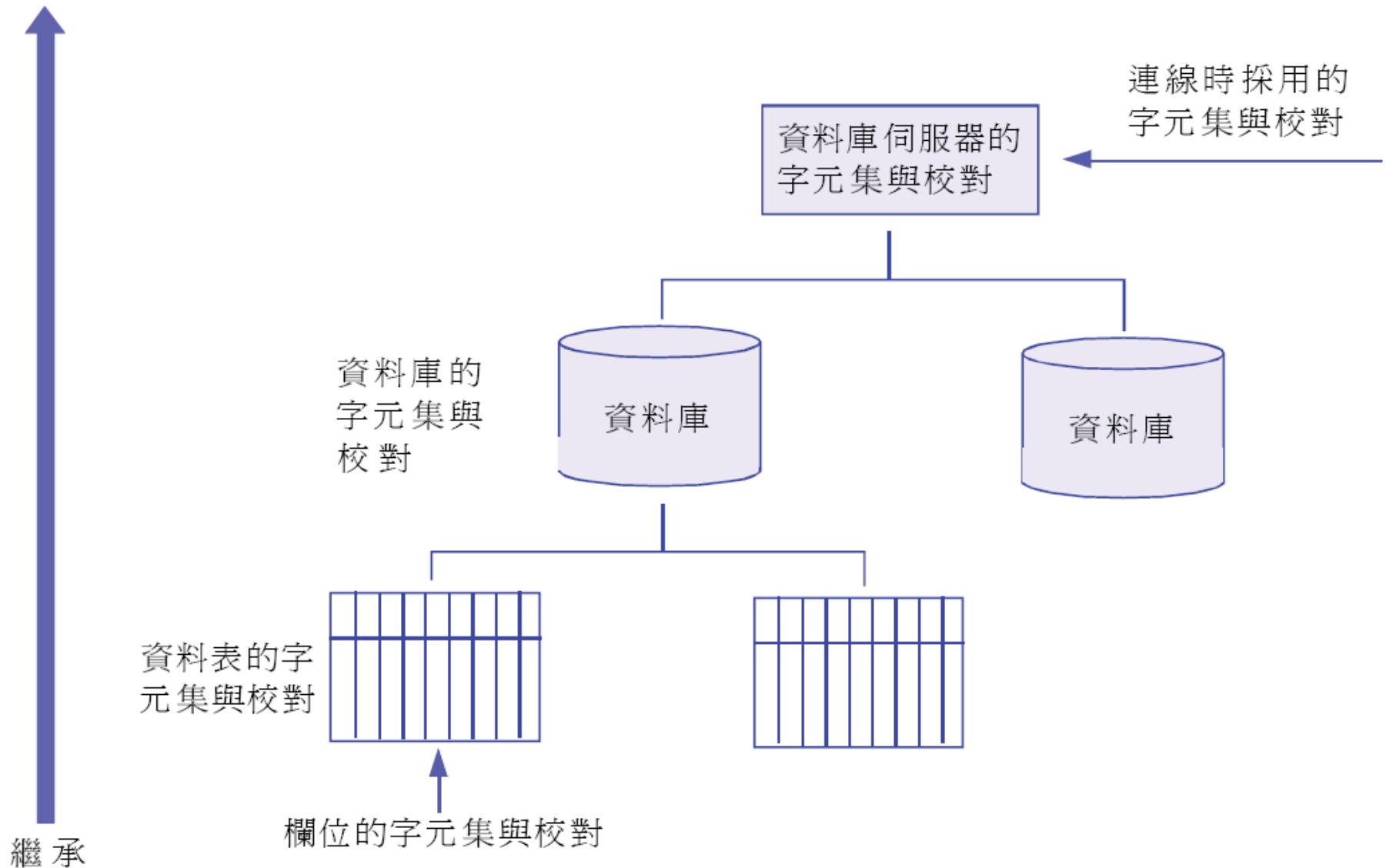
- 前面曾經提到，為了避免 Big5 編碼可能造成的衝碼問題，所以本書統一採用 UTF8 編碼。
- 故隨後建立資料庫及資料表時，我們會使用 "utf8" 字元集與 "utf8_unicode_ci" 校對。
- 因為校對的名稱是以其所屬的字元集做為開頭，所以使用 phpMyAdmin 建立資料庫與資料表時，phpMyAdmin 只會要求我們指定校對的方式。
- phpMyAdmin 會自行依照指定的校對幫我們設定字元集。

MySQL 資料庫的字元集與校對

■ 字元集與校對的層級

- 為了提供最大的彈性, 在 MySQL 中, 可以在伺服器、資料庫、資料表、欄位與連線等不同層級設定字元集與校對 (如下頁圖所示)。
- 除了連線時採用的字元集與校對是獨立設定以外, 至於伺服器、資料庫、資料表、欄位等層級, 預設下層會繼承上一層的設定。
- 例如資料庫設定為 "utf8_unicode_ci" 校對, 其下的資料表如果沒有特別設定, 就會繼承採用 "utf8_unicode_ci" 校對。

MySQL 資料庫的字元集與校對



MySQL 資料庫的字元集與校對

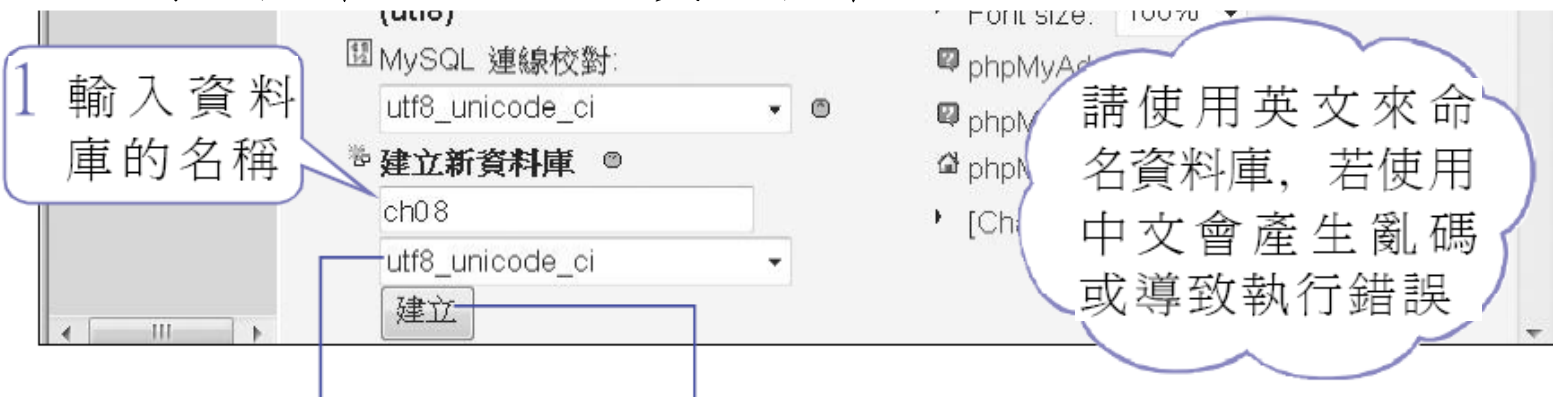
- 若未特別設定, MySQL 預設會採用 "latin1" 字元集與 "latin1_swedish_ci" 校對。
- 所以為了正確儲存中文資料, 建議您建立資料庫或資料表時, 務必明確指定採用 "utf8" 或 "big5" 字元集。
- 除了伺服器與資料庫內可以設定字元集與校對, 當連線 MySQL 存取資料時, 也應該設定連線採用相符的字元集與校對。

MySQL 資料庫的字元集與校對

- 例如要存取字元集為 "utf8" 的資料表時，若連線採用 "latin1" 字元集，取出與儲存的資料都會成為亂碼，必須設定連線採用 "utf8" 字元集，才能正確存取資料。
- 關於伺服器、資料庫、資料表、欄位等層級如何設定與修改字元集與校對，會在下一節說明，至於連線層級的設定方法，請參考 10 - 2 節。

建立資料庫

- 接下來我們將說明如何在 MySQL 中，以 phpMyAdmin 建立資料庫與資料表。
- 請依照 1 - 5 節的說明，以 root 帳號登入 MySQL 的管理程式 — phpMyAdmin，在首頁如下操作即可建立資料庫：



- 2 選擇資料庫的校對。本書使用 UTF8 編碼，所以請選擇 utf8_unicode_ci
- 3 按建立鈕

建立資料庫

進入此連結中可以
更名資料庫或修改
字元集與校對設定

資料庫建
立成功



關於 SQL 語法的用途，請參考 9-1 節

按此連結可以刪除資料庫

建立資料表

- 建立了資料庫之後，便可以在資料庫中建立資料表。例如筆者想要建立一個資料表，用來儲存留言板資料，所以如下操作建立資料表：

1 在此列示窗中選擇要建立資料表的資料庫

The screenshot shows the phpMyAdmin interface. On the left, the 'Database' dropdown menu is open, showing 'ch08 (0)' selected. The main area displays the 'Create new table' form for the 'ch08' database. The form has two input fields: 'Name' (labeled '名稱') with the value 'guestbook' and 'Columns' (labeled '欄位數目') with the value '4'. There is an 'Execute' button at the bottom right. A callout bubble points to the 'Database' dropdown menu with the text: '請使用英文來命名資料表，若使用中文則會產生亂碼或導致執行錯誤'.

2 輸入資料表的名稱

3 輸入要建立的欄位數目

4 按此鈕

建立資料表

5 輸入各欄位的名稱
(可使用英文或中文)

7 如果使用字串型別，
可以在此指定長度。
此欄位功能請參考後
面的詳細說明。

此處可指定欄位與資料
表要使用的字元集與校
對，請保留空白，繼承資
料庫的設定使用 UTF8

欄位	型態	長度/集合	校對
留言編號	INT		
姓名	VARCHAR	20	
留言	VARCHAR	256	
日期時間	DATETIME		

資料表注解文字:

儲存引擎: InnoDB

校對: utf8_general_ci

儲存 或新增 1 執行

6 選擇各欄位
的資料型別

等一下設定好各欄
位後，按儲存鈕即
可建立資料表

儲存引擎在
此採用預設
值即可

建立資料表

9 請先如圖設定其他欄位

屬性	Null	預設值 ²	附加
UNSIGNED	not null		auto_increment
	not null	無名氏	
	null		
	not null		

8 向右拉曳滑動桿


建立資料表

- 以下分別說明上述各欄位設定的用途：
 - **長度**：若是整數型別，可以在此欄位指定顯示寬度；如果是浮點數型別，可設定有效位數；若為字串型別，則可指定長度。
 - **屬性**：如果是數字型別，可以在此欄位設定是否有正負號，預設有正負號，若選擇 **UNSIGNED** 表示無正負號。至於 **ZEROFILL** 屬性請參考 8 - 12 頁。
 - **Null**：設定是否允許 Null，也就是新增記錄時，該欄位是否可以不輸入資料。

建立資料表

- **預設值**：設定欄位的預設值。若設定了預設值，當新增記錄時，如果沒有給予該欄位資料，預設會填入此處設定的值。
- **附加**：若為數字型別，是否讓該欄位自動編號。
- 預設不自動編號，若選擇 **auto_increment**，新增記錄時，MySQL 會以累加的方式自動選擇編號，設定為該欄位的值。
- 請注意，**每個資料表只能有一個欄位設定為 auto_increment，而且該欄位必須是 Primary Key**。

建立資料表

- ：如果選擇此圖示下的單選鈕，表示該欄位要設定為 Primary Key。
- 輸入各欄位的名稱與設定後，按**儲存**鈕即可建立資料表：

若按此連結可瀏覽
資料表內的記錄

此連結可觀看或修改資
料表的各欄位的設定

按此連結可
以新增記錄



成功建立資料表

這個連結可以
更名資料表或
修改校對設定

如果按這兩個連結，將會清
空資料表內的所有資料，
或是直接刪除此資料表

MySQL 的儲存引擎

- MySQL 內建了多種儲存引擎，儲存引擎指的是資料存放在資料表中的機制與結構，所以每一種儲存引擎所建立的資料表具有不同的特性。
- 例如 MyISAM 引擎具備相當快的讀取效能，但是缺少 Foreign Key、交易 (Transaction) 功能。
- 相對的，InnoDB 引擎則具有 Foreign Key、交易 (Transaction) 功能，不過多數情況下讀取時的效能不如 MyISAM 引擎。

MySQL 的儲存引擎

- 本書採用 MySQL 5.0 版，原本其預設的儲存引擎為 MyISAM，不過 WAMP5 修改了其設定，所以在 WAMP5 架構的環境下，預設會使用 InnoDB 引擎建立資料表。
- 如果您想要瞭解 MySQL 有哪些儲存引擎，以及各儲存引擎的特點，請參考 <http://dev.mysql.com/doc/refman/5.0/en/storage-engines.html>。

新增、瀏覽、編輯與刪除資料

- phpMyAdmin 除了提供資料庫的管理介面外，也提供了相當好用的資料輸入與編輯介面。
- 如果您想要使用 phpMyAdmin 輸入資料，請如下操作：

因為我們已經設定留言編號欄位為 auto_increment, MySQL 會自動為此欄位輸入編號，所以不必自行輸入

2 按新增連結

3 在此處輸入資料

型態	函數	Null	值
int(10) unsigned	<input type="text"/>		<input type="text"/>
varchar(20)	<input type="text"/>		無名氏
varchar(256)	<input type="text"/>	<input type="checkbox"/>	測試一下
datetime	<input type="text"/>		2007-08-20 15:34:30

新增、瀏覽、編輯與刪除資料

1 在左邊欄位選擇要輸入資料的資料庫與資料表

位	型態	函數	Null	值
id	int(10) unsigned			
姓名	varchar(20)			
留言	varchar(256)		<input checked="" type="checkbox"/>	
日期時間	datetime			

儲存為新記錄 然後 返回 執行 重置

如果希望按執行鈕後仍顯示此畫面，以便繼續輸入其他資料，請在此選擇新增一筆記錄

按執行鈕即可新增資料

按此圖示會開啓一個月曆視窗，如果您要輸入多筆資料，可繼續在此輸入

新增、瀏覽、編輯與刪除資料

- 除了新增資料外，您也可以如下在 phpMyAdmin 內瀏覽、編輯與刪除資料：

2 按瀏覽連結



新增、瀏覽、編輯與刪除資料

顯示: 30 開始列數: 0

顯示為: 水平 示欄名: 100

依鍵名排序: 不適用 執行

<T>	留言編號	姓名	留言	日期時間
<input type="checkbox"/>  	1	無名氏	測試一下	2007-08-20 15:34:30
<input type="checkbox"/>  	2	邱大熊	哈囉	0000-00-00 00:00:00

全選 / 全部取消 選擇的資料表:   

按此鈕可以編輯該筆記錄

若按此鈕則可刪除該筆記錄

如果要編輯或刪除多筆記錄，請先勾選記錄前面的多選鈕，然後按這兩個鈕即可

此處顯示目前所有資料

備份與回復資料庫

- 天有不測風雲，勤做資料庫備份是避免重要資料流失的基本原則。
- 除此以外，當您要將資料庫搬移或複製到其他主機，也需要使用備份與回復的功能，本節將說明如何使用 phpMyAdmin 備份與回復資料庫。

備份資料庫

- 請開啟 phpMyAdmin, 在首頁如下操作即可**備份資料庫**：



在首頁按輸出連結可備份資料庫

備份資料庫

- 或者在資料庫操作畫面按**輸出**連結, 則可以**備份資料表** :

1 選擇要備份的資料表所在的資料庫



2 在此按輸出連結
可備份資料表

備份資料庫

- 上面兩者的差別在於，**備份資料庫**時 phpMyAdmin 會備份資料庫的相關定義 (如名稱、欄位、字元集)。
- 所以日後還原時，會先新增一個資料庫，然後再將資料還原進該資料庫。
- 而**備份資料表**時只會備份資料表與其中的記錄，日後進行還原時，必須還原到一個已經存在的資料庫中。
- 備份資料庫與資料表的功能選項差異不大，因此下面以備份資料表為例說明，請如下操作。

備份資料庫

- 1 選擇要備份的資料表（配合 **Ctrl** 鍵可以多重選擇）

右邊的設定保留預設值即可

檢視資料庫的備份概要 (dump schema)

輸出

guestbook

全選 / 全部取消

SQL 選項

於標題加入個人註解 (\n 開新行)

☐ 使用事務塊 (Transaction) 方式

☐ 暫定外來鍵 (Foreign Key) 檢查

SQL 兼容模式

NONE

備份資料庫

2 選擇要輸出的格式，除非您要把資料匯入其他軟體，否則請選擇 SQL

The screenshot shows a web-based interface for backing up a database. It features two radio buttons for output format: 'SQL' (selected) and 'XML'. Below this is a section for download options, including a checked checkbox for '下載儲存' (Download and save), a text field for '檔案名稱樣式 (1):' containing '___DB___' with a checked checkbox for '保留樣式名稱' (Keep name style), and compression options: '不適用' (Not applicable), '"zipped"' (selected), and '"gzipped"'. An '執行' (Execute) button is located at the bottom right. Blue lines with numbers 2, 3, and 4 point to the 'SQL' radio button, the '下載儲存' checkbox, and the '執行' button respectively.

3 勾選此項以便下載備份檔，若不勾選則資料庫內容會直接顯示在網頁上

可選擇使用 Zip 格式壓縮備份檔

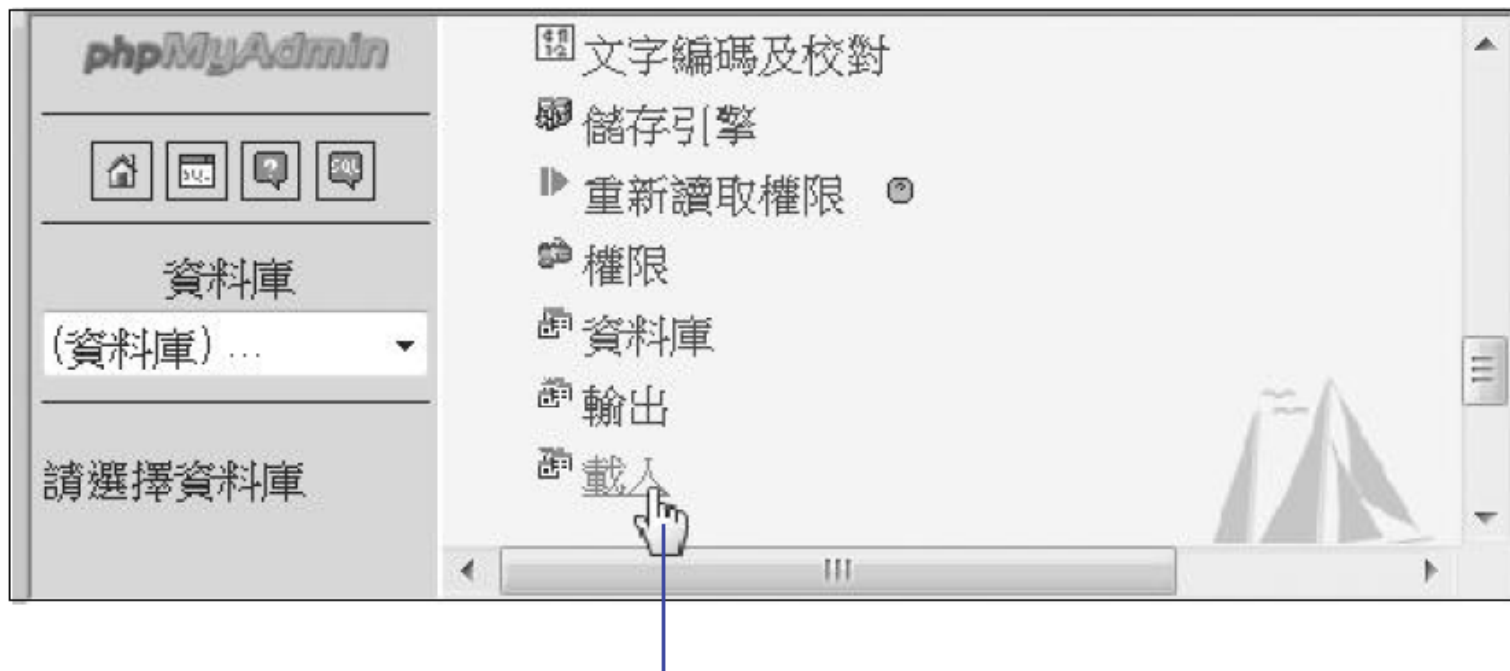
4 按此鈕開始下載

■ 下載的檔案便是您目前資料庫內容的備份檔，使用這個檔案便可以回復或是搬移資料庫。

回復資料庫

- 回復資料庫可以用於兩種場合，一個是資料庫毀損後，在原有主機上將備份的資料還原回去，一個則是將備份資料拿到其他伺服器載入，用以搬移或是複製資料庫到其他主機。
- 兩種場合的用途雖然不同，但是操作的方法皆相同。
- 如果先前以備份資料庫的方式進行備份，則必須以**回復資料庫**的方式還原或載入資料(請先確定伺服器中沒有同名的資料庫)。

回復資料庫



在首頁按載入連結可回復資料庫

- 若之前使用備份資料表的方式，則必須以**回復資料表**的方式還原資料（請先確定資料庫內沒有同名的資料表）。

回復資料庫

- 1 選擇要回復的資料表所在的資料庫



- 2 在此按載入連結可回復資料表

- 回復資料庫與資料表的功能選項差異不大，以下將以回復之後第 9 章要使用的資料庫為例。

回復資料庫

- 請在 phpMyAdmin 的首頁按**載入**連結，然後如下操作：

1 請先放入書附光碟，然後按此鈕選擇光碟 Ch09 目錄下的 Ch09.sql

2 選擇編碼的文字集

載入

載入檔案

文字檔案的位置 (D:\Ch09\Ch09.sql) 瀏覽...

文字編碼檔案: utf8

載入檔案壓縮會自動檢查: 不適用, gzip, zip

回復資料庫

部份載入 —

☒ Allow interrupt of import in case script detects it is close to time limit. This might be good way to import large files, however it can break transactions.

開始時略過多少行記錄 (語法)

載入檔案格式 —

☒ SQL

SQL 選項 —

SQL 兼容模式

☐

執行

此處保留預
設值即可

3 按此鈕即可回復
第 9 章要使用的
Ch09 資料庫

8 - 4 MySQL 資料庫的存取權限

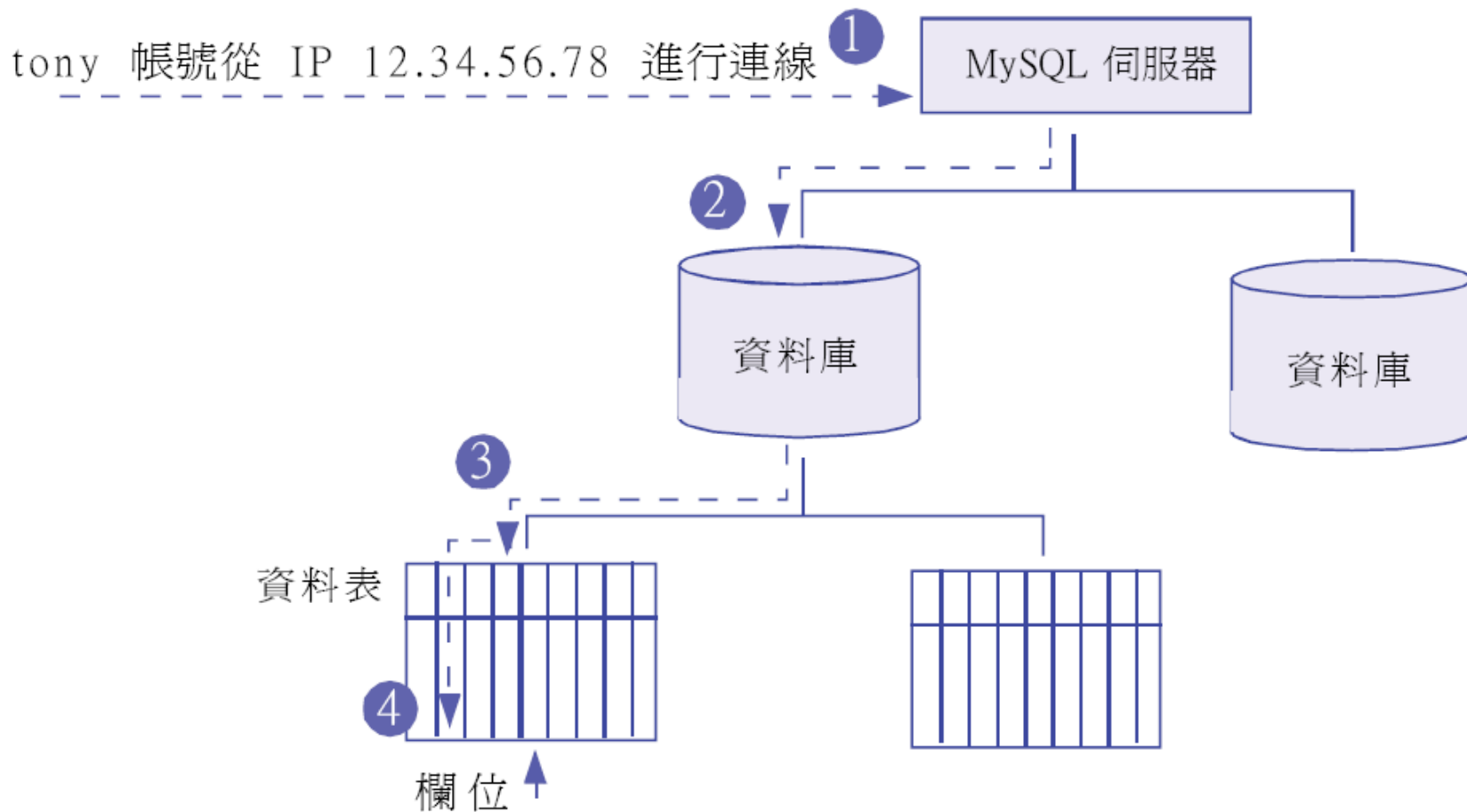
- 本節將介紹 MySQL 的存取權限，讓我們可以管理什麼人可以登入伺服器，能夠讀或寫哪些資料庫，以及可對資料庫做哪些動作。
- 本書為了簡化學習的複雜度，範例程式皆使用 root 帳號登入資料庫。
- 不過正式架設網站時，如果也使用具有最高權限的 root 帳號，只要程式有漏洞，就可能被惡意者入侵進行破壞或竊取資料。

MySQL 資料庫的存取權限

- 所以建議您依照本節的說明，另外建立一個較低權限的帳號（例如限制只能讀取某個資料庫），讓正式網站的程式以低權限的帳號存取資料庫，降低被侵入的危險。

MySQL 的驗證架構

- 當一個使用者要登入 MySQL 伺服器存取資料庫時，必須先經過以下的驗證流程：



MySQL 的驗證架構

- (1) 驗證 tony 帳號的名稱、密碼、與 IP 位址是否正確, 通過後才可登入, 並且取得其整體權限。
- (2) 檢查 tony 是否具有此資料庫的存取權限。若未設定, 則繼承整體權限。
- (3) 檢查 tony 是否具有此資料表的存取權限。若未設定, 則依序繼承資料庫、整體權限。
- (4) 檢查 tony 是否具有此欄位的存取權限。若未設定, 則依序繼承資料表、資料庫、整體權限。

MySQL 的驗證架構

- 從上圖可以看到, 使用者連線 MySQL 伺服器時, 需要正確的帳號、密碼、與 IP 位址才能獲許登入。
- 而登入後是否可以讀寫資料, 則視其是否在資料庫、資料表、欄位各個層級具有正確的存取權限。
- 隨後將說明如何使用 phpMyAdmin 在 MySQL 伺服器中新增使用者, 並且設定該使用者的權限。

新增帳號並設定整體權限

- 請開啟 phpMyAdmin, 在首頁如下操作即可新增使用者：



新增帳號並設定整體權限

此處列出目前伺服器內所有帳號

2 按此連結新增使用者帳號

按此鈕可以編輯此使用者

The screenshot shows the phpMyAdmin interface for managing users. On the left is a sidebar with navigation icons and a database selection dropdown. The main area is titled '使用者一覽' (User Overview) and contains an alphabetical index (A-S). Below the index is a table of users. The first user listed is 'root' on 'localhost' with 'ALL PRIVILEGES'. To the right of the table is an edit icon. Below the table are links for '新增使用者' (Add User) and '移除已選擇使用者' (Remove Selected User). At the bottom right is an '執行' (Execute) button. Three blue lines with text annotations point to specific elements: one to the alphabetical index, one to the '新增使用者' link, and one to the edit icon.

	使用者	主機	密碼	整體權限	授權	
<input type="checkbox"/>	root	localhost	是	ALL PRIVILEGES	是	
↑ 全選 / 全部取消						

新增使用者

移除已選擇使用者
(廢除使用者所有有效之權限並刪除.)

☐ 刪除與使用者相同名稱之資料庫.

執行

勾選帳號後按此鈕可刪除使用者

新增帳號並設定整體權限

4 選擇該使用者可使用的連線來源，選擇本地表示只允許該使用者在本機上登入，若選擇任何主機則不限制其連線來源，如果想要指定 IP，請選擇文字輸入後，在右側欄位輸入 IP

3 輸入使用者的帳號名稱

5 輸入密碼

The screenshot shows a web form titled "新增使用者" (Add User). It contains several input fields and buttons. Blue lines with numbers 3, 4, and 5 point to specific fields: line 3 points to the "使用者名稱" (Username) field containing "user"; line 4 points to the "主機" (Host) dropdown menu which is set to "本地" (Local); line 5 points to the "密碼" (Password) field, which is masked with dots. Below the password field is a "確認密碼" (Confirm Password) field, also masked. At the bottom, there are buttons for "產生密碼" (Generate Password) and "複製" (Copy), followed by an empty text input field.

新增使用者	
登入資訊	
使用者名稱:	文字輸入: user
主機:	本地 localhost
密碼:	文字輸入:
確認密碼:
產生密碼:	<input type="button" value="產生"/> <input type="button" value="複製"/>
	<input type="text"/>

新增帳號並設定整體權限

可在此處設定整體權限（下列各選項的英文字是 SQL 語法，會於第 9 章說明）

整體權限（全選 / 全部取消）

注意: MySQL 權限名稱會以英語顯示

將滑鼠移到項目上面時會出現說明

資料

- ☐ SELECT
- ☐ INSERT
- ☐ UPDATE
- ☐ DELETE

結構

- ☐ CREATE
- ☐ ALTER
- ☐ INDEX
- ☐ DROP
- ☐ CREATE TEMPORARY TABLES
- ☐ CREATE VIEW

系統管理

- ☐ GRANT
- ☐ SUPER
- ☐ PROCESS
- ☐ RELOAD
- ☐ SHUTDOWN
- ☐ SHOW DATABASES

容許建立新資料庫及資料表。


執行

6 按此鈕即可建立使用者

新增帳號並設定整體權限

- 為了安全起見, 建議您不要給予使用者整體權限, 應該依照該使用者所需的資料庫, 給予該資料庫的權限即可。

設定資料庫權限

- 上面新增使用者之後，會顯示使用者的設定畫面 (或者在 8 - 34 頁的圖中按 ，也會出現一樣的畫面)，您可以在這裡變更該使用者的密碼、修改整體權限、或設定資料庫權限。
- 請如下操作即可設定資料庫權限。

設定資料庫權限

1 找到指定資料庫權限項目

2 在此欄位選擇要為該使用者設定權限的資料庫

指定資料庫權限

資料庫 權限 授權 指定資料表權限 執行

不適用

於以下資料庫加入權限:

文字輸入:

文字輸入:

information\schema

ch08

mysql

phpmyadmin

test

執行

更改密碼

不用密碼

設定資料庫權限

若想要讓該使用者具備該資料庫的所有權限，請按此連結即可全選

✎ 使用者 'user'@'localhost' - 資料庫 ch08: 編輯權限

— 指定資料庫權限 (全選 / 全部取消) —

注意: MySQL 權限名稱會以英語顯示

設定資料庫權限

3 在此處選擇權限

資料	結構	系統管理
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> CREATE	<input checked="" type="checkbox"/> GRANT
<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> ALTER	<input checked="" type="checkbox"/> LOCK TABLES
<input checked="" type="checkbox"/> UPDATE	<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> DROP	
	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	
	<input checked="" type="checkbox"/> CREATE VIEW	
	<input checked="" type="checkbox"/> SHOW VIEW	
	<input checked="" type="checkbox"/> CREATE ROUTINE	
	<input checked="" type="checkbox"/> ALTER ROUTINE	
	<input checked="" type="checkbox"/> EXECUTE	

容許刪除記錄.

執行

4 按此鈕即可完成設定