

Chapter

# 15

鍾震耀 博士

中華企業資源規劃學會資深專案經理

國立中央大學企業管理系資管組博士

國立台北科大/中原大學/萬能科大兼任助理教授

eric9281@gmail.com

許秉瑜 博士

國立中央大學國際事務處國際長

國立中央大學企業管理系教授

## 資料庫管理與 ERP 系統

- ✓ 學習從資料庫管理角度理解 ERP 系統導入
- ✓ 瞭解資料庫角色的重要性
- ✓ 瞭解資料庫設計的技巧與溝通

2015/02/02 Ver2

# 資料庫管理與 ERP 系統

## 前言

在當今重視國際化競爭與建立完整企業 e 化環境的氛圍下，企業資源規劃 (Enterprise Resource Planning, ERP) 系統已經成為每家企業在追求競爭優勢與營運效率過程中不可或缺的基石，企業使用 ERP 軟體時所需要的科技環境也隨著網路(Network)與資料庫(Database, DB)技術進步而不斷演進，從過去僅能安裝在 AS/400 或 RS/6000 等大型主機(Mainframe)架構環境上的 ERP 軟體(如圖 1 所示)，進展到目前主流市場上最多 ERP 軟體安裝的主從架構(Client/Server)環境(如圖 2 所示)，甚至有些 ERP 軟體廠商因為技術快速進步所發展出的新一代 ERP 軟體的架構透過軟、硬體遠端的服務都支援雲端運算(Cloud Computing)架構環境，國內中華電信 HiCloud 與神通科技 MiCloud 兩大公有雲上都已經架設 ERP 系統，給於規模不大的公司租賃使用，國外也有一些軟體廠商將 ERP 系統架構延展支援雲端環境，例如微軟公司也將 ERP 系統搬上自家雲端平台 Azure 提供給企業租用。

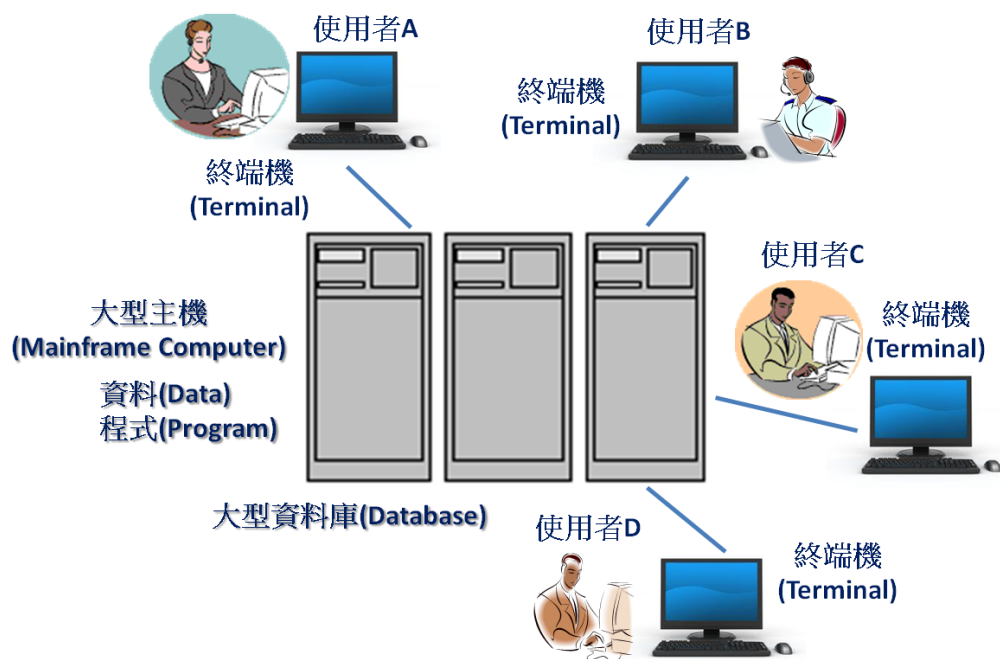


圖 1：大型主機(Mainframe)架構環境

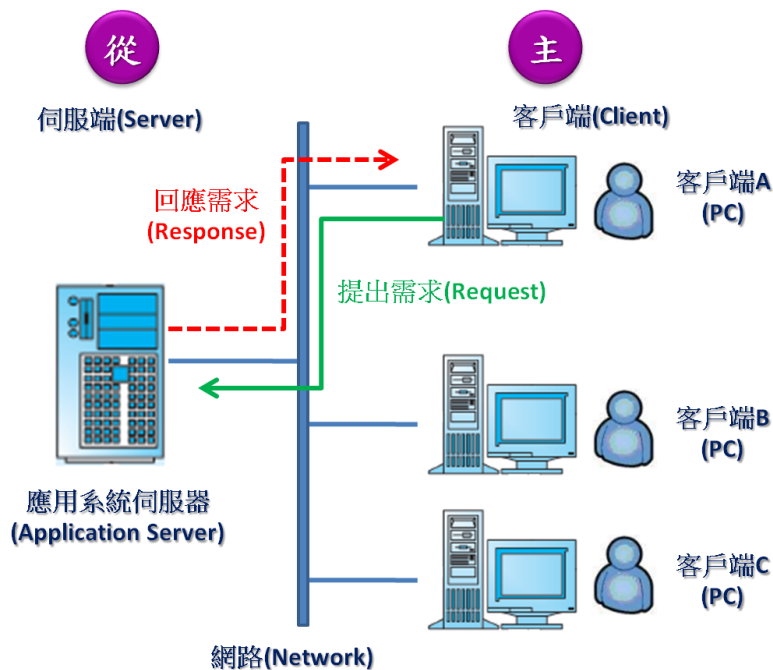


圖 2：主從架構(Client/Server)環境

此外為了讓 ERP 系統導入後能在企業執行面順暢運作，導入時利用企業流程模擬工具來描述與設計該公司所有應用模組 (Application Modules) 的企業流程 (Business Processes)，使得這些企業流程盡量符合 ERP 軟體所提供的標準流程與系統功能，即趨向使用所謂的 ERP 軟體最佳實務範例 (Best Practices) 與參考模型 (Reference Models)，實務上就是希望能減少 ERP 導入專案所撰寫的外掛程式 (Add-on) 數量，盡量讓客製化 (Customization) 的機會降到最低，ERP 系統的軟體廠商之如此建議是希望將來使用該 ERP 軟體的企業，都能根據標準設定的 ERP 軟體流程以及 ERP 軟體的設計邏輯，讓各個應用模組之間能夠自動且無誤地傳遞資訊與提供決策所需表單與報表資料，換言之，若要讓 ERP 系統發揮其功效，企業的日常營運流程必須要與 ERP 標準軟體流程互相吻合，如此便可以把 ERP 系統的潛力徹底發揮出來，否則就必須增加許多額外費用成本撰寫外掛程式來補足企業流程與 ERP 軟體流程兩邊不相吻合的部分，目前實務上已有一些企業流程模擬工具被使用，如 ARIS (Architecture of Integrated Information Systems) 即為一個理論與軟體工具兩者兼備的企業流程模擬工具。

由技術面上來看，將企業內部運作的主要流程所需要的資料整合規劃出一套完整的資料庫，並於此資料庫上讓事先已經完成設計、開發好的 ERP 軟體根據企業日常營運活動(生產規劃、銷售與配銷、人力資源、倉儲、採購、財務會計等)蒐集並可以儲存新的交易資料(Transaction Data)，亦可依需求刪除或修改資料庫現有的交易資料，以及提供查詢與列印報表功能，在大型的 ERP 系統中，此整合的資料庫通常會包含有大約 2 萬張的資料表(Tables)，如圖 3 所示，這一套完整的資料庫(包含這些數萬張的資料表)在導入 ERP 系統時，會同時被建置完成，除非遇到需要客製化的部分，不然資料庫的綱要(Schema)內容就都已經確定下來，即包含資料庫中各資料表的各個欄位資料型態、欄位大小、欄位是否允許虛值(Null Value)、那些欄位為主鍵(Primary Key, PK)、那些欄位為外來鍵(Foreign Key, FK)、以及那些欄位是否要設定有索引(Index)功能等綱要內容都已經確定下來，接下來再透過 ERP 團隊的顧問規劃完整轉檔計劃，交由公司的 MIS 部門負責將舊系統(Legacy System)的資料轉入 ERP 系統的資料庫中，如此看來資料庫很顯然是 ERP 系統中最重要基礎技術之一，因為不管是由哪一 ERP 軟體廠商所提供的 ERP 系統都會用到資料庫，簡而言之資料庫是一個幫助企業儲存公司每天日常活動所使用到的資料的場所，其所扮演的角色重要性由此可知不容忽視，如圖 4 所示。

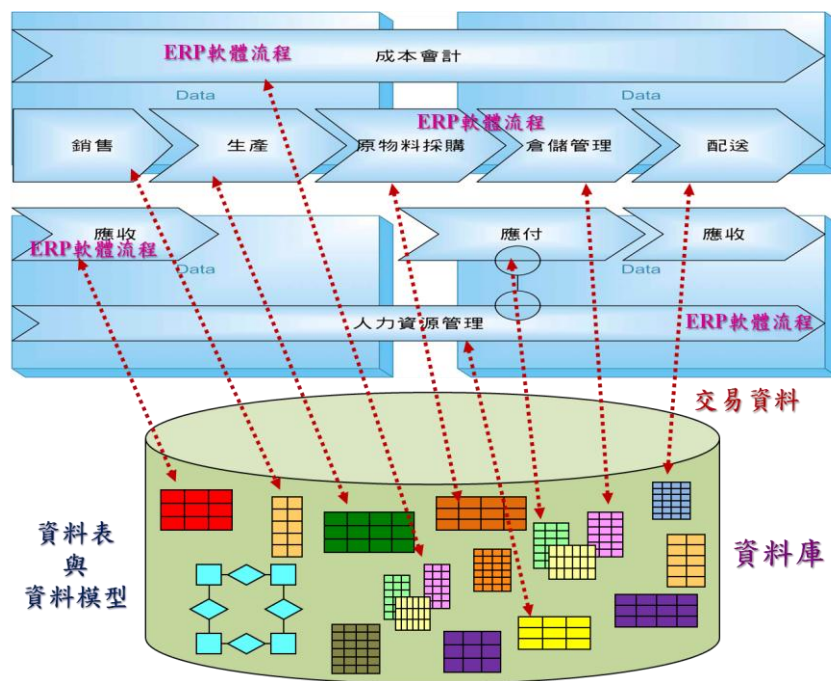


圖 3：ERP 流程與資料庫系統

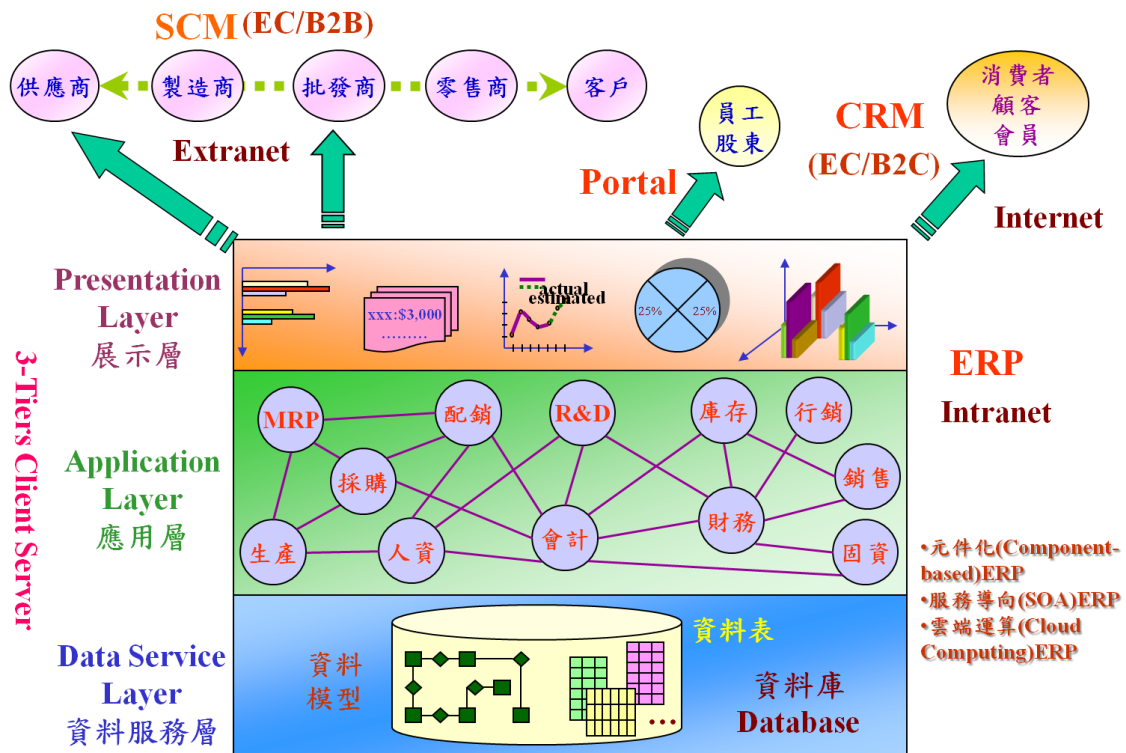


圖 4：資料庫在 e 化環境中的角色

而歸屬於 MIS 部門的 IT 工程師們對於資料庫的知識與應用都有一定程度認知，甚至有些人員(例如資料管理師 DBA)必須天天透過一些軟體工具進到資料庫環境工作，但是在 ERP 導入專案中有很多非 IT 人員(公司內其他業務單位)加入，甚至成為主導整個 ERP 導入專案的主要角色，但是發現在 ERP 導入專案會議上卻常常見到雙方存在許多無法輕易瞭解的資料庫專用話術而造成溝通不良，常見非 IT 人員不易瞭解的有主鍵(Primary Key, PK)欄位與外來鍵(Foreign Key, FK)欄位這兩件事情，因為 IT 人員經常會不自主地使用這兩個名詞來與非 IT 人員進行溝通，告訴他們建立 ERP 報表時要用。

另一方面，我們也發現 IT 工程師也經常被價格(Price)這一個欄位搞混，到底非 IT 人員所需要的是進貨價格、銷售價格、VIP 價格、稅後價格中的哪一個術語，客戶如果沒有詳細說明清楚該欄位的意義，IT 人員是會設計出錯誤的 ERP 報表，因為這些價格是屬於不同企業流程活動進行時候所使用，不可以混為一談，唯有回頭好好跟非 IT 人員訪談溝通，再根據 ERP 系統中的資料庫資料模型內容進行比對找出產生對應資料的企業流程

活動，此為實務上較常用的方法，目前實務上所使用的資料庫資料模型以及資料模型圖仍以實體關係體模型(Entity Relationship Model, ERM)所繪出的實體關係體圖(Entity Relationship Diagram, ERD)較多人使用，例如 IT 人員熟悉 ERP 系統資料庫模型中所有資料表之間的關聯性、主鍵、外來鍵、索引以及資料欄位所在資料表位置，當與非 IT 人員溝通時，就可使用 ERD 中的邏輯順序來尋找進貨價格、銷貨價格等欄位經常出現的資料表位置，此時就會明白非 IT 人員所說的需求內容，才能夠瞭解客戶的需求內容，以上所述的這些溝通不良狀況經常在 ERP 專案導入時出現，尤其是雙方在討論客製化範圍確認所使用到的外掛程式規格內容會議上發生誤會，進而耽誤 ERP 專案導入時程，其影響深遠不可言喻，雙方不得不在這問題發生前，努力提早瞭解並得到共識，故認識資料庫與 ERP 系統的議題其重要性刻不容緩。

除了前言之外，本章節共分四個小節，第一節從兩個實務情境內容解讀資料庫與 ERP 系統經常需要溝通的問題，以及 IT 人員與非 IT 人員在 ERP 專案導入時須聽得懂的話術，第二節說明何謂資料庫系統(Database Systems, DBS)、資料庫管理系統(Database Management Systems, DBMS)以及資料庫(Database, DB)，第三節簡介設計資料庫所使用到的技術，實體關係體模型與實體關係體圖，最後在第四節中舉一個量身訂做、完全客製的 ERP 導入個案討論。

## 第一節 從兩個實務情境內容解讀資料庫與 ERP 系統

### 情境一：場景與內容

在一個藥妝產品製造業工廠的會議室內，早上正在進行 ERP 專案導入的規格商討與確認，客戶端員工正在抱怨透過此 ERP 軟體原物料放行功能畫面無法進行判讀是領取哪一個版本的包材資料，因為該情境公司的包材會因為不同時間點將包材外觀重新設計過，即便只修改包材外觀顏色，也是屬於不同版本的包材，因此必須對包材版本做控管 (Version Control) 的工作，而目前將導入的 ERP 系統畫面上缺少一個描述版本資訊的欄位，使用者透過公司的 ERP 專案導入的專案經理 (Project Manager, PM) 建議希望 ERP 軟體公司的顧問 (Consultant) 能夠協助修正，將包材的版本欄位加到原物料放行領料的功能畫面中，讓客戶端員工可以看到包材的版本欄位資訊，顧問思考後答應此需求變更，並且說明以進貨批號來填入識別，問客戶端員工是否同意？客戶端員工同意說沒錯！只要在畫面上看的見這一個欄位資訊讓我們工作方便這樣就可以了。但是顧問進一步提出說明這部分必須撰寫外掛程式讓客戶端員工進到此功能畫面時可以看到包材的版本資訊，這樣就可以了嗎？因為版本欄位並非原先 ERP 軟體標準功能中的設定欄位，但是技術上可以克服額外加到功能畫面中……，通常後面這一段顧問的說明內容，客戶端員工都興趣缺乏沒有在專心聽，也聽不太懂技術內容，但同時也怕不懂會被對方取笑，所以客戶端員工通常會微笑點頭裝懂，溝通會議結束，而剛出道沒有太多專案經驗的顧問也會以為客戶端員工好像聽懂且同意了，事實上並不是這樣情形，這是一個誤會的開端。

經過三週，此修正功能驗收後開始使用，結果第四週在會議上包材數量不足夠的問題被提出，客戶端員工驗收修正後的 ERP 功能，發現每次領完包材料件之後，倉儲 (Warehouse) 現場盤點的包材數量永遠與 ERP 軟體上的庫存量數據有出入，導致後續採購誤判數量不缺，以為包材料件足夠不需要採購，但事實上包裝部門的包材料件嚴重數量不足，無法在預定時程內完成包裝 (Packing) 工作，造成包裝工作嚴重塞車，需要大量加班時數才能完成出貨 (Shipping) 前的包裝工作，為此，高層臨時透過專案督導委員會緊急召開會議討論，重新檢視版本欄位加入 ERP 軟體中的問題，經過三天討論，後來查出畫面中版本欄位就只留下最後一版本資訊顯示，向上查不到舊版本資訊，經 ERP 軟體公司

另一位資深顧問出面協助查出，當時外掛程式規格只開出顯示包材版本資訊，ERP 軟體上永遠只留下最後一版本包材資訊顯示，舊版本資訊都被蓋掉了，程式設計師 (Programmer) 僅依照規格完成程式功能，因此 ERP 軟體上包材庫存量與現場盤查數量會不同，客戶端專案經理非常生氣，就差沒當場翻桌撕破臉，因為這是一個很離譜的錯誤，ERP 軟體公司高層出面道歉，外掛程式與時間成本由 ERP 軟體公司自行吸收且同意更換一位顧問才平息這件事情的紛爭。

造成此錯誤的原因究竟為何呢？因為 ERP 軟體公司的當時顧問以撰寫程式的變數方式開出外掛程式的需求規格，並沒有使用資料庫觀點開需求規格，姑且不評論當時顧問是否深知資料庫觀點為何？以及如何運用資料庫規劃技術？但是如果客戶端人員事先被訓練過資料庫概念，且知道資料庫的精神與基本重點話術，則此問題雙方就會提早進行溝通，而不致於到了缺少包材料件讓包裝部門工作延遲與加班才蹦出這一個嚴重的問題。

### 延伸討論：

根據情境一的内容，顧問答應協助在 ERP 系統功能畫面中增加版本資訊欄位的需求，如果不以程式設計觀點，而以資料庫的設計觀點，應該如何做哪些事情才是正確且長遠的對策？(提示：增加欄位呢？或者增加資料表？細節處裡內容會是什麼？)



## 情境二：場景與內容

在一個 3C 通路商的會議室內正在進行 ERP 專案導入的規格商討與確認，客戶端員工正在抱怨使用此 ERP 軟體時，郵寄給客戶的對帳單上面無法顯示收款條件相關資訊，這樣業務人員到客戶公司現場時就無法順利說明收款細節與收款項目，因為客戶基本上從對帳單裏頭並無法判定是否已經達到收款條件內容，例如為何今日業務會過來收款，是否收款日已經到期或超過了呢？這些客戶的疑問讓業務會很頭痛，因此客戶端員工希望此 ERP 軟體在對帳單功能面畫中能夠加入並且顯示收款條件欄位資訊，讓客戶知道是否收款時間已到，但是此 ERP 軟體標準流程功能中並沒有這一個收款條件欄位，希望顧問能夠幫忙解決這一個需求，此需求是屬於應該要給於的範圍，但是卻是一個問題，顧問幾經思考後答應會將此問題帶回去公司想出解決方法，也同意此需求變更是合理的要求，並且說明只要多一個可以顯示收款條件內容的欄位，內容使用收款條件代碼或者是直接帶入收款條件呢？問客戶端員工是不是這樣的要求，客戶端員工同意說沒錯！就利用收款條件代碼帶出收款條件內容值就可以，這樣子看見 ERP 系統的畫面會比較容易理解收款條件為何，這樣處理就可以了。

這位實務情境的顧問相當有經驗，知道這個需求變更是必須完成交給客戶的功能，但是苦思目前 ERP 軟體標準功能中並無此欄位，如前言所說明，這一套完整的資料庫(包含這些數萬張的資料表)在導入 ERP 系統時，會同時被建置完成，除非遇到需要客製化的部分，不然資料庫的綱要(Schema)內容就都已經確定下來，因此在標準 ERP 軟體功能下，資料庫中收款單功能對應到的綱要並沒有收款條件欄位，這是無法改變的事實。此時顧問思維過去所接受的 ERP 教育訓練與個案經驗，當企業流程與系統功能不搭配時，需要馬上想到解決方法來沉著應戰，方法上有五種可利用，分別為：

1. 從 ERP 系統標準功能中再找尋一次。
2. 是不是有快速解決的方法?如果是 ERP 系統中未設計的欄位，可否用其他欄位代替?
3. 在企業流程與 ERP 系統功能間找到妥協，找出折衷的方法
4. 考慮撰寫外掛程式來對治不搭配的問題
5. 直接修改 ERP 系統程式(應極力避免，因可能導致系統日後無法升級)

經過幾次與客戶端員工溝通，並偕同客戶端員工一起確認企業流程與 ERP 系統功能，結果，這一位顧問找到了一個可行的答案，透過第二個方法，如果是 ERP 系統中未設計的欄位，可用其他尚未被使用的欄位來代替，因為這一套 ERP 軟體在對帳單功能畫面上地址資料有預留兩個欄位，為地址 1 與地址 2，因為地址 2 幾乎都空白不用，暫時以地址 2 欄位作為收款條件欄位使用，在此個案中雙方在資料庫與 ERP 系統一起檢視，來回問答並解惑，IT 人員嘗試著用客戶端員工聽得懂的話術進行溝通對話，而客戶端員工在資料庫基本話術中事先接受訓練過，知道主鍵欄位與外來鍵欄位對於功能畫面與建立報表的概念，才能夠快速順利得到溝通的效果。

### 延伸討論：

根據情境二的內容，顧問利用”其他尚未被使用的欄位來代替”解決問題，不管是否顧問運氣好，但是至少能將過去所學的 ERP 訓練課程與經驗派上用場，樹立顧問自己沈穩專業的形象，專案結束後 ERP 軟體公司或許會考慮將這一個需求放入標準 ERP 軟體功能畫面中，請問如果沒有而是個案處理，請問將來 ERP 軟體改版升級時，想想看 ERP 軟體公司以及這一家 3C 通路商可能會發生哪些事情？需要如何面對呢？

## 個案小結：

根據上述兩個實務情境說明，ERP 導入個案內容重點都與資料庫基本需要理解的術語有關係，那就是主鍵(PK)欄位與外來鍵(FK)欄位的認識，有了這兩大重點的基礎就可以讓雙方在 ERP 導入專案會議上的溝通迎刃而解，第一個個案雖然剛開始為一個負面的情境內容，但是從中學習到 ERP 軟體顧問的認知如果沒有進一步由客戶端員工協助確認，通常互有誤會存在甚至扭曲客戶端的需求內容，而這些所謂客戶端員工協助確認的能力，可以先由資料庫基礎概念中學習，從學習 IT 人員經常口中所說的主鍵(PK)與外來鍵(FK)欄位的用意，以及用在撰寫外掛程式(或臨時報表)時 IT 人員經常會詢問候客戶端員工的問題中熟悉主鍵(PK)欄位與外來鍵(FK)欄位使用方法，譬如顧問常會詢問用這個某某欄位來幫您撈資料、撰寫報表程式可以嗎?上述的”這個某某欄位”通常就是指資料庫中某資料表的主鍵欄位，為何經常會使用主鍵欄位幫忙撈資料或撰寫報表程式呢?因為主鍵欄位的資料值被規定在資料庫中一定不會有重複值出現，也就是唯一值存在資料庫中，因此不會撈錯資料給使用者查看，例如在客戶資料通常都建立在客戶主檔資料表中，正常來說我們會幫每一位不同的客戶安排一個唯一的編號，這些客戶編號組成一個欄位就是常見的客戶編號欄位，因此每一位客戶在此欄位中一定對應到一個客戶編號值，例如客戶王大明，給於編號 C0001，將來看見客戶編號值 C0001 就馬上反應到這 C0001 正是指我們公司的客戶王大明先生，絕對不會查到的是別人，這就是主鍵(PK)的用處。

另外，在資料庫中與主鍵(PK)欄位有同重要地位的欄位就是外來鍵(FK)，當我們也常聽到 IT 人員經常說為了讓使用者可以在功能畫面或者報表上看見詳細資料，可以透過某一個欄位幫你帶出大量的詳細資料，上述的”某一個欄位”通常就是指資料庫中某資料表的外來鍵(FK)欄位，為何可以使用外來鍵(FK)欄位就可以幫忙帶出詳細資料呢?因為某資料表的外來鍵欄位被規定在資料庫中也可以成為另一個資料表的主鍵欄位，如此就可以很容易串接合併(Join)分屬於兩個資料表的所有欄位，想看那些詳細資料欄位就告訴 IT 人員，他就會使用剛剛的語氣問您”可以透過某一個欄位幫你帶出大量的詳細資料，這樣可以嗎?”這時候請非 IT 人員一定要記得”某一個欄位”就是這一個資料表的外來鍵欄位，例如銷貨單異動資料表上有客戶編號欄位，如果沒意外這裡都是以代碼方式顯現某一個客戶，C0001，但是代碼值 C0001 人類不輕易明白此客戶是誰、住在哪裡、付款條件為

何等資訊，透過客戶編號欄位就可以到客戶主檔資料表中查到更詳細的客戶資料，王大明、住新竹、貨到兩個月付現金，這就是外來鍵(FK)的用處。

第二個 ERP 導入個案為一個正面的個案內容，從中學習到 ERP 軟體顧問的努力尋找解套方法，並在過程中與客戶端人員分享 ERP 軟體知識，以及資料庫中欄位設定的狀況，最後用外來鍵方式帶出客戶要的詳細畫面功能，不但降低了雙方誤會的可能性，更沒有扭曲客戶端的需求內容，且這些客戶端員工有能力確認顧問提出的規格內容 IT 術語，是一個成功溝通的方式。

## 第二節 資料庫系統

### 2.1 資料庫角色的重要性

對於長期以來檔案系統(File Systems)在企業組織運作所產生的問題，強調「集中(Centralization)」與「整合(Integrity)」的管理資料理念的資料庫技術的出現無疑是給予企業主一線曙光，在過去檔案系統年代，資料(Data)與應用程式(Application Programs)之間相依程度太高，會因為資料結構的變動或資料儲存方式的改變而需要去修改相關的程式內容，如圖 5 所示，資料庫技術可以讓資料與應用程式之間相依程度減到最低，即不會因為資料結構的改變或資料儲存方式的變動而需要去大量修改相關的程式內容，如此觀念就是所謂的資料獨立(Data Independency)特性。

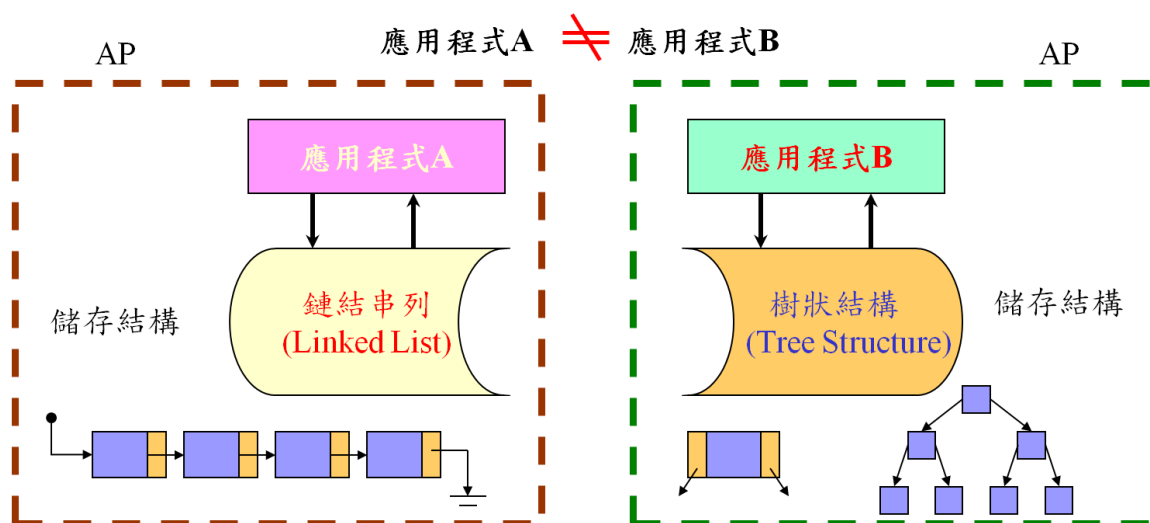


圖 5：資料相依性高的缺點

為了讓資料與應用程式之間相依性降低，在 1975 年由制定標準的機構 ANSI/X3/SPARC 提出資料庫管理系統(Database Management Systems, DBMS)三層架構，分別為外部層(External Level)、概念層(Conceptual Level)、內部層(Internal Level)，如圖 6 所示，並請軟體廠商遵循各層制定的標準來設計、開發資料庫管理系統產品，資料庫管理系統是一個由軟體廠商設計開發出來的軟體，可以提供定義(Definition)、建立(Creation)、

查詢(Querying)、修正(Update)等功能來控管(Administration)資料庫(Database, DB)內容以及維護儲存在資料庫中公司各項營運交易資料與資料表(Table)。以目前市場上最為多公司採用且最穩定的關聯式資料庫管理系統(Relational Database Management Systems, RDBMS)為例，透過 RDBMS 使用建立(Create)、移除(Drop)、修改(Alter)指令來定義企業將來使用的資料庫與資料表，此外還提供查詢(Select)、新增(Insert)、刪除(Delete)、修改(Update)等指令來維護資料庫中資料表(Table)內的資料，至於對於是否有權限存取資料庫中的資料，以授權(Grant)、取消授權(Revoke)兩個指令功能來控管每一個登入關聯式資料庫管理系統的使用者帳號之存取權限，上述那些可以完成某些功能的指令來維護資料庫中資料的正確性與完整性統稱為 SQL 指令，SQL 是指結構化查詢語言(Structural Query Language)，由 IBM 公司於 1970 年代所研發，是關聯式資料庫管理系統存取資料特有的標準語言。

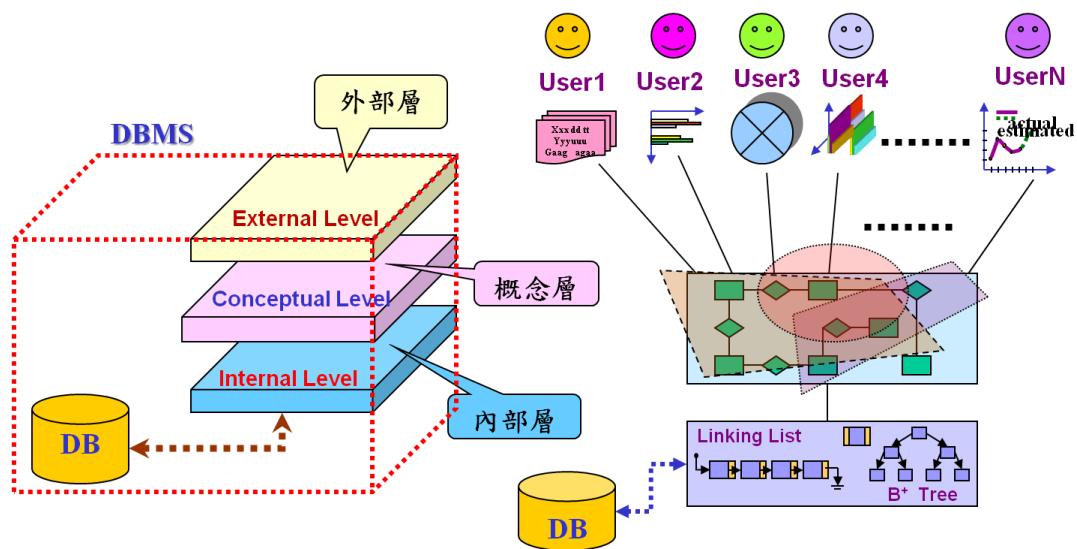


圖 6：資料庫系統三層架構

借助三層架構可以滿足資料獨立特性，在各層所描述各層資料的定義內容即稱為綱要(Schema)，資料庫管理系統中的三層架構與綱要簡述如下：

1. 外部層(External Level)是指一般使用者觀點的資料，通常都只有整個企業資料庫資料

的一部分，如圖 6 所示。而外部綱要(External Schema)主要是用來描述外部層所要顯示的資料，如圖 7 所示，因為每一個外部綱要只會描述資料庫的部分資料，並且隱藏其他部份的資料，或者是有權限才能讀取的資料，因此每一個外部層使用者觀點都需要一個外部綱要，一個資料庫管理系統是可以擁有與管理多個外部綱要，圖 7 中有兩個外部綱要，User1 與 User2 的外部綱要各別定義的 3 個欄位。

2. 內部層(Internal Level)是指實際儲存觀點所呈現的資料，是實際資料庫儲存在電腦儲存裝置的資料，這裡所指通常是硬碟中。而內部綱要(Internal Schema)是描述內部層實際儲存觀點的資料，包含定義儲存資料的資料結構(Data Structure)方式以及哪些資料須要建立索引(Index)或主鍵欄位是否要設定索引叢聚(Clustering)，內部綱要內容最接近機器，但不涉及實際儲存媒體的硬體結構描述，圖 7 中有一個使用 c 語言表達的內部層綱要。
3. 概念層(Conceptual Level)是指資料庫設計師(Database Designer)或 DBA 觀點的資料，是完整的資料庫，概念上所呈現的是這家企業營運所需要的以及所產生的交易資料結合而成的資料集合體，因此概念綱要(Conceptual Schema)只會有一個，是以組織中之全體使用者為主呈現出資料庫全部資訊的內容，概念綱要亦為外部綱要與內部綱要內資料之間的橋樑，圖 7 中顯示整個資料庫系統就只有一個概念綱要，但是可以提供給數個不同使用者觀點的外部綱要修改後使用或不修改直接沿用，圖 7 中 User1 與 User2 的外部綱要各別修改概念綱要欄位並定義的自己有興趣或有權限想看的外部綱要，User1 的外部綱要就是指 E\_No、Name、Age 那 3 個欄位，User2 的外部綱要就是指 No、Name、Address 那 3 個欄位。

另一方面，如圖 4 所示，當企業經營活動延伸至外部時，如上下游整合的供應鏈(Supply Chain Management, SCM)活動，甚或針對末端進行顧客關係管理(Customer Relationship Management, CRM)活動時，上下游供應鏈中各個成員資料的分享與透通(例如由上游廠商協助下游廠商管理監控存貨狀況)，以及分析顧客消費的交易資料之進階運用(例如瞭解客戶消費行為購物關聯性的市場購物籃分析)，皆需要資料庫相關技術的支援，

因此一個較完整的資料庫規劃與設計，以及較正確與一致的資料提供，是未來任何一家企業競爭的利器，當然如果能進一步搭配以交談方式擷取不同觀點的線上分析處理(On-Line Analytic Processing, OLAP)技術、注重多維度與某一主題大量彙總性資料查詢的資料倉儲(Data Warehouse, DW)系統以及利用演算法(Algorithm)找出對使用者決策活動上俱有管理意義之關係規則的資料探勘(Data Mining, DM)方法，如此的整合性技術配套方案為企業所帶來的效益，是從注重效率(Efficiency)的「把事情做對(Do The Things Right)」層次之觀念，提升到注重效能(Effectiveness)的「做對的事情(Do The Right Things)」層次的競爭理念，對於競爭優勢有著持續的未來，接下來針對資料庫系統的組成做基礎性的介紹與說明。

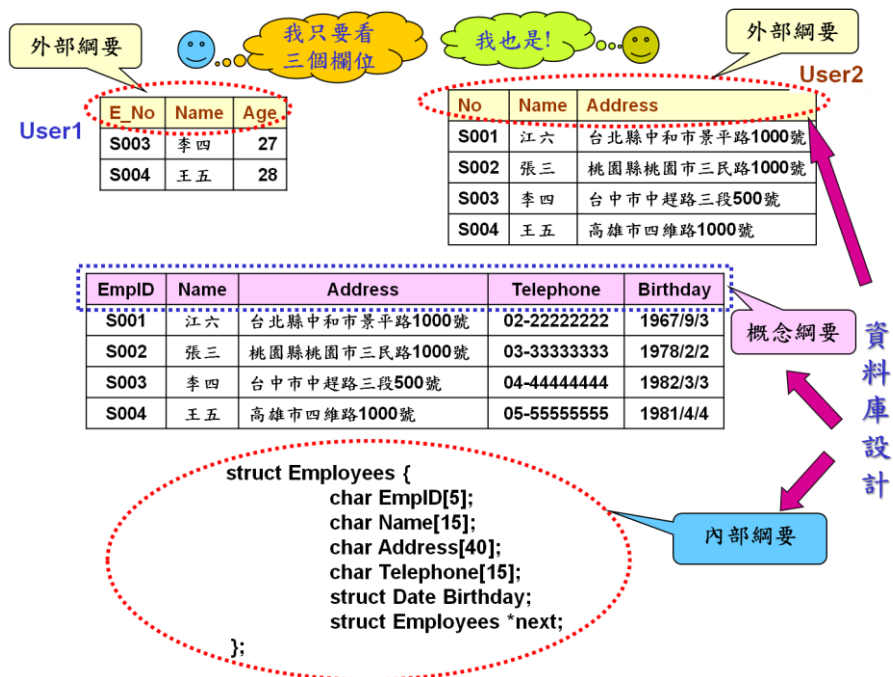


圖 7：資料庫系統三層綱要概示

## 2.2 資料庫系統概述

對於學習資料庫的初學者，都會先將傳統檔案系統(File Systems)環境的缺點問題(例如資料獨立性太低)、資料庫技術的歷史演進進行瞭解，除此之外，在討論基本資料庫應



用領域前也都會先將資料庫(Database, DB)、資料庫管理系統(Database Management Systems, DBMS)、資料庫系統(Database Systems, DBS)等三個相當容易混淆的名詞一一清楚所代表的意義的不同，以避免後續學習其他資訊系統應用領域時，會讓資料庫等相關角色名詞錯亂而無所適從甚或發生進一步學習的困難，為此先說明資料庫技術的歷史演進，接著再談論 DB、DBMS、DBS 三個名詞的範圍與意義。

首先以圖 8 所示，從 1960 年代開始使用傳統檔案(Files)為來服務企業內部的業務應用，這些檔案(Files)作為電腦化資訊系統(Computer-based Information Systems, CBIS)的資料儲存體(因為大眾都已經明白需要電腦輔助，後來電腦化三個字可被省略，稱為資訊系統 IS)，例如經濟訂購量(EOQ)資訊系統、安全存量(Safety Stock)資訊系統、物料清單(Bill of Material, BOM)管理資訊系統以及 1960 年代中期的物料需求規劃(Material Requirements Planning, MRP)資訊系統。

1970 年代先後出現了階層式資料庫(Hierarchical Database)以及網路式(Network Database)資料庫，在階層式資料庫中資料的資料儲存方式是一個樹狀結構(Tree Structure)呈現，如圖 8 所示，每一個節點上都儲存一筆記錄(Record)，節點間資料順序是從樹根(Root)開始由上而下，由左而右次序，適合用來描述一對多的資料關係，例如一個供應商(Supplier)可以供應給公司不同種類的產品(Products)，每一種產品中又包含許多不同的原物料(Materials)，這些資料節點之間的關係是透過鏈結(Link)來完成，譬如使用鏈結串列(Linked List)方式實作，早期 IBM 公司的 IMS/VS(Information Management Systems/Virtual Storage)就是階層式資料庫管理系統(Hierarchical Database Management Systems)，查詢路徑必須事先在資料儲存時候有建立鏈結，否則沒有鏈結的資料就會查詢不到，此外也不適合描述真實世界中多對多的資料關係，因此有學者相繼提出以網路方式來儲存資料，其儲存資料的資料結構是階層式資料結構的擴充，資料節點之間鏈結方式不只可以由上而下的一對多關係，且允許有由下而上的一對多關係，IBM 公司的 IDMS(Integrated Database Management Systems)就是網路式資料庫管理系統(Network Database Management Systems)，這兩類的資料儲存方式是必須事先規劃設計好，將來查詢的路徑才有依據，以階層方式網路方式串接，圖 8 中箭頭所在也就是可查詢路徑，如果兩個資料節點間沒有設定鏈結路徑，則將來無法查詢資料，以查詢路徑設定來看，網路式比階層式較複雜，

網路式是階層式的一種延伸方式，讀者可以從上面描述中察覺到這兩類資料庫管理系統無法達到資料獨立特性，因為程式撰寫與資料儲存方式之間依賴性很高，且兩類資料庫管理系統都是先有產品之後，再回頭訂定其資料模型內涵。

1980 年代資料改以二維表格(Table)方式儲存，且查詢方式是不需要事先規劃設計，此二維表格因為是有意義規定的表格，特稱為關聯表(Relation)，此類資料庫管理系統稱為關聯式資料庫管理系統(Relational Database Management Systems, RDBMS)，學者 E. F. Codd 先以數學的集合論(Set Theory)當基礎提出關聯式資料模型，將資料以關聯式(Relation)的表格方式來表示，在 1982 年之後才有產品發展出，是目前市面上最多公司採用的資料庫管理系統。

真實世界中資料的多元豐富化已延伸到大量的圖片、聲音、影像等多媒體(Multi-media)資料以及非結構化的文件(Texts)資料，二維表格方式因有限制已經無法表達，產學界紛紛尋求解決方法，到了 1990 年代以物件(Objects)方式來表達資料庫中各項資料，包含傳統固定長度資料、多媒體資料、文件資料，再加當時盛行的物件導向系統分析與設計(Object-Oriented System Analysis and Design, OOSA&D)、物件導向程式設計(Object-Oriented Programming, OOP)語言的推波助瀾，因此有物件導向(Object-Oriented Database Management Systems, OODBMS)以及物件關聯(Object-Relational Database Management Systems, ORDBMS)資料庫管理系統相繼被提出應用。

到了 2000 年代，企業主看到公司累積大量歷史資料，因此想進一步了解公司資料庫中所保留的歷史資料是否有過去經營經驗的可取之處，因此出現主題導向式(Subject-Oriented)資料庫，此為資料倉儲(Data Warehouse)的概念，另外企業主也試著嘗試請 IT 人員發展一些聰明的演算法則(Algorithms)來詮釋公司累積大量歷史資料背後隱含的企業經營知識(Knowledge)，此為資料探勘(Data Mining)的觀念，而目前已經有不少軟體廠商將這兩個觀念融合發展全新的想法，此即商業智慧(Business Intelligence, BI)的開端。

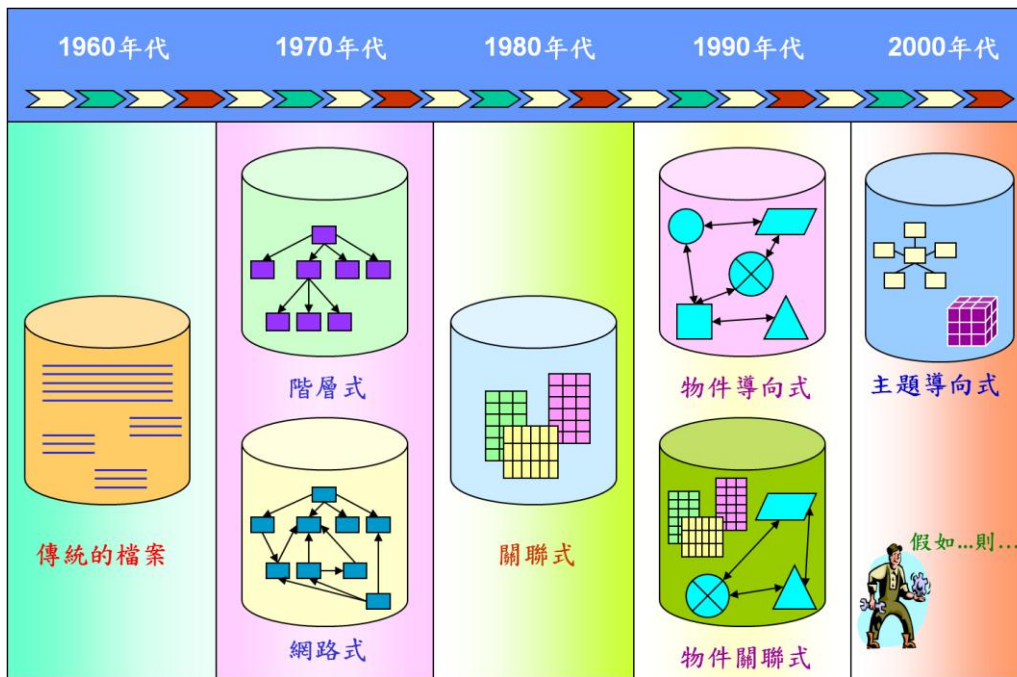


圖 8：資料庫技術的歷史演進

一般而言資料庫系統(DBS)會由四個部份所組成，如圖 9 所示，分別為由(1)使用者(User)、(2)資料(Data)、(3)硬體(Hardware)以及(4)軟體(Software)等所組成，以下針對四項組成的成員分別說明。

(1)使用者(User)

使用者又可依其負責的工作內容細分為下列三類使用者：

- (A)應用程式設計師/應用程式(Application Programmer / Application Programs)
- (B)直接(或終端)使用者(End Users)
- (C)資料庫管理師(Database Administrator, DBA)

(2)以資料庫系統的運作方式可將資料粗分為兩大類：

- (A)分別為由使用者與系統產生的運算資料(Operational Data)
- (B)以及作為備份(Backup)或回復(Recovery)時參考依據的異動記錄(Transaction Log)

此上所述(A)與(B)兩類資料所要求空間大小會因為資料庫系統應用領域的不同而在資料庫軟體參數設定上有所不同，例如薪資系統的批次處理工作重點會要求運算資料空間大一點，換言之，此二部份的資料即是組成資料庫的主體部份。

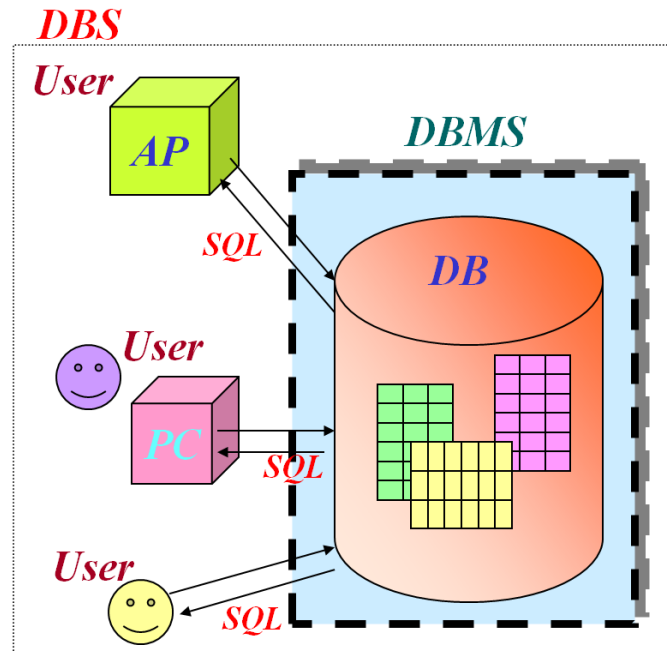


圖 9：資料庫系統組成元件架構圖

(3)硬體設備部份一般包含整個執行資料庫系統功能的電腦主機、輔助儲存設備(例如磁碟、硬碟與光碟機等)、防範未來災變的備份設施(例如磁帶機、可讀寫光碟機等)，甚至需準備停電時可啟動的不間斷電源系統(Uninterruptible Power Systems, UPS)等等週邊設備。

(4)軟體部份包含資料庫管理系統(DBMS)本身、一些擷取使用資料庫中資料所開發的應用程式(Application Programs)、介於資料庫管理系統與應用程式之間的中介軟體(Middleware)，例如 ODBC 資料庫驅動程式可以完成異質性資料整合的工作，這些軟體中以 DBMS 為最重要。

綜合以上四個部份的說明可以彙整出，資料庫(DB)是儲存資料的場所，資料庫管理系統(DBMS)是一套已經開發完成軟體程式，基本上提供使用者儲存資料的功能，亦可視

為使用者與資料庫之間溝通的介面橋樑，換言之，資料庫系統(DBS)可看成是由軟體(例如資料庫管理系統即為軟體)、資料(例如資料庫即為儲存資料的場所)、使用者、硬體(例如構成安裝資料庫系統平台的伺服器或其他設備)所組成的一個電腦化集合體。

### 第三節 資料庫設計

#### 3.1 資料庫生命週期

為求資料庫系統執行的穩定度，必須有賴於優良的資料庫設計，為此，本段落先從資料庫生命週期(Database Life Cycle, DBLC)議題討論起，任何產品從無到有皆有其生命週期的概念，資料庫系統亦可視為一項產品，其生命週期如圖 10 所示，就像生命體一樣會經歷生、老、病、死的過程，但對於資料庫系統而言會經由維護、更新、強化等工作將生命週期不斷地循環而使系統更趨於穩定下去，基本上由圖 10 中可以看出資料庫生命週期可粗分為六個階段，而資料庫設計工作被定位在第二個階段，此階段如果要再仔細探究其細部工作項目又可分為 7 項子工作必須完成，其中細分為與資料庫軟體無關的基礎設計部份，此部份由 (2a) 資料分析與需求蒐集、(2b) E-R Modeling、(2c) 正規化 (Normalization) 的過程以及 (2d) 資料庫雛型檢驗等四項工作完成；另一部份是與

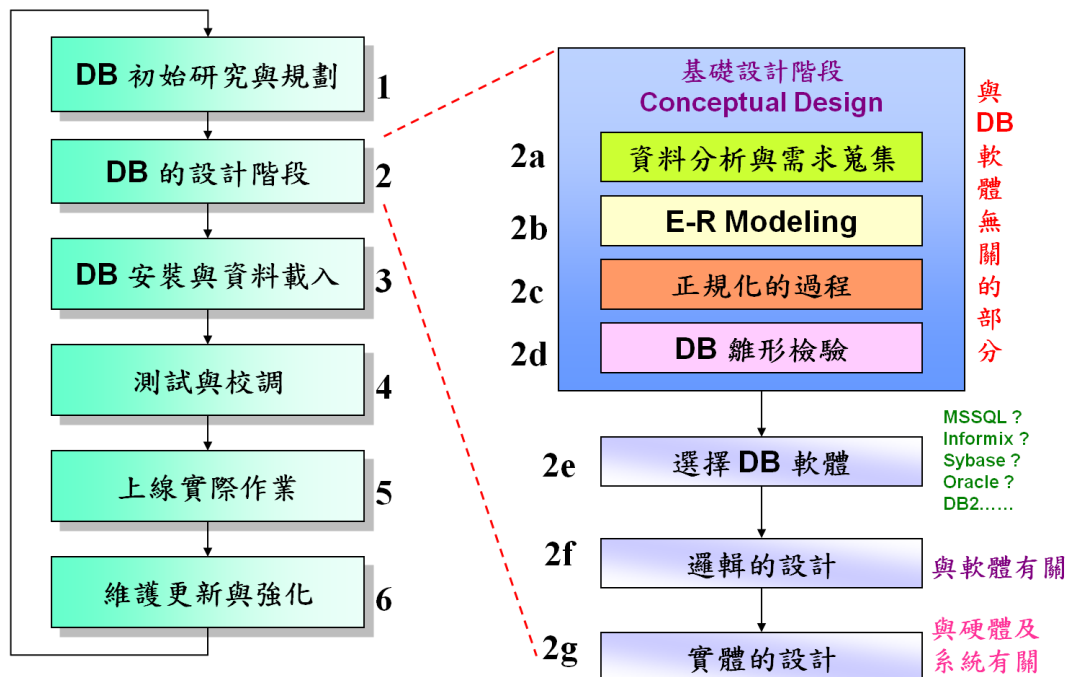


圖 10：資料庫的生命週期

資料庫軟硬體有關係的部份，此部份首先必須進入實際資料庫軟體產品選擇的決策活動，如 (2e) 選擇資料庫軟體，接著將 (2b) 規劃過的 E-R Model 轉換成所選擇的資料庫軟

體儲存結構，即進入 (2f) 邏輯設計 (即一般所謂的程式設計工作)，最後需根據所搭配的硬體與作業系統平台做 (2g) 實體設計工作。

誠如上述所言，真正與資料庫軟體有關的工作只不過是整個 DBLC 第二階段的 (2f) 小段，但是此段卻讓大部份人員不斷地天天忙碌追逐，甚或疲於奔命於開會決議要採用 MySQL 產品呢？MS Access 產品呢？DB2 產品呢？Sybase 產品呢？Oracle 產品呢？MS SQL Server 產品呢？還是某一個物件導向資料庫管理系統 (Object-Oriented Database Management Systems, OODBMS) 產品呢？皆在此處花了資料庫設計專案工程相當多時間，卻忘了與整個專案工程最重要也同時為最有用的部份，即 (2a)、(2b)、(2c) 以及 (2d) 工作的正確性，此四小階段雖然在實務界已經使用甚久，但就是很少有人提及或談論，甚或嚴重到刻意不願意碰觸，若以關聯式資料庫管理系統 (Relational Database Management Systems, RDBMS) 為例，究其專業領域知識而言，皆以陳品山教授 (Peter P.S. Chen) 於 1976 年所提出的實體關係體資料模型 (ERM) 與 E. F. Codd 於 1972 年所提出的正規化 (Normalization) 技術為主，如圖 11 所示，如不加以釐清其設計內容與步驟前後順序，對企業未來 ERP 與其他 e 化系統的發展影響甚巨，上述兩項技術在整個資料庫規劃與設計過程的中被認為是資料庫系統優劣的檢視條件，其重要性相當，為此本章會先針對實體關係體資料模型設計技術在 3.2 節中做完整說明。

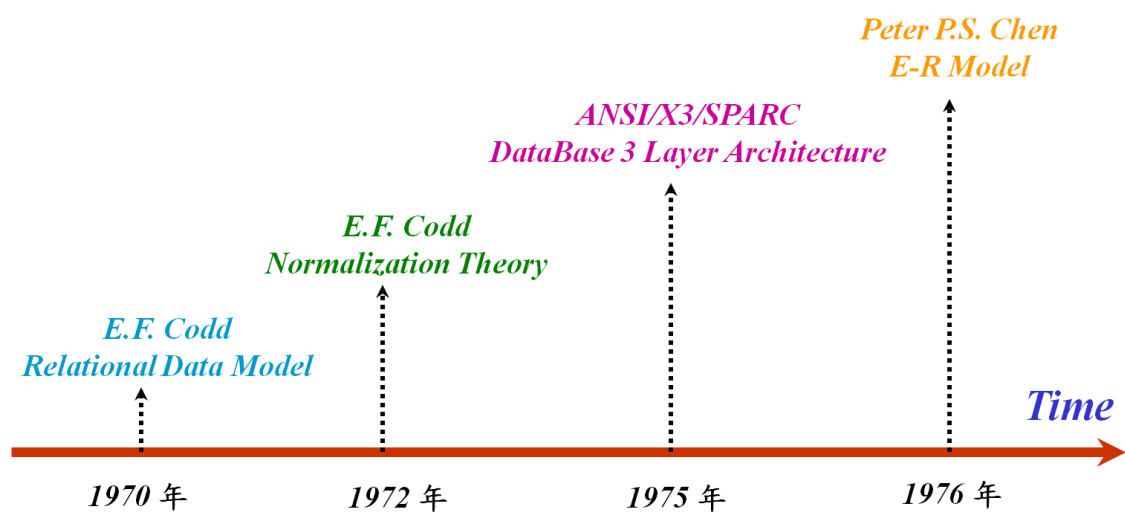


圖 11：資料庫設計的理论工具

### 3.2 實體關係體資料模型

到目前為止大部份資料庫系統、管理資訊系統、資訊管理導論的專書都會提及實體關係體資料模型(ERM)，甚或近年來國內以資料庫為基礎的程式設計電腦用書也會多多少少將 ER Model 介紹出來，作為程式設計師資料存取藍圖的依據，為此，沒有理由在 e 化為基礎的 ERP 發展過程中可省略這個重要技術，其重點說明如下：

- (1) 全名 *Entity Relationship Model*。
- (2) 縮寫名稱為 *ER Model*。
- (3) 中文稱為 實體關係體模型。
- (4) 由陳品山教授 (*Peter P.S. Chen*) 於 1976 年所提出。
- (5) 是一種語意資料模型 (*Semantic Data Model*)。
- (6) 目的-輔助規劃資料庫系統中概念層設計 (*Conceptual Layer Design*)。
- (7) 功能-協助 SA 人員做資料的分析與規劃的工作。
- (8) 技巧-以圖示 (*Graphical*) 方式表達，易於理解。
- (9) 產出-實體關係體圖 (*Entity Relationship Diagram*)。
- (10) 簡稱 *ER Diagram* 或 *ERD*，可作為 DBS 公司整體性資訊架構建構藍圖，同時也可當作與使用者溝通的工具。

基本上此模型由實體 (Entity)、關係體 (Relationship)、連結數 (Connectivity) 以及屬性 (Attribute) 等四項元素所組成，分述如下：

#### (1) 實體 (Entity)

意義：一個能被清晰分辨的事或物，吾人可區別之者。

觀念：一個人、一個零件、一家公司，甚至一個觀念 (或抽象事與物)，多媒體資料庫系統設計此門課程 3 學分，3 學分摸不著，課程名稱亦摸不著，但有實際活動，此為抽象觀念；至於區別 (Identify) 一詞牽涉到唯一性，隱含著 Primary Key 的概念。



附註：一般在表達上會引用實體集合（Entity Set）的觀念，Entity Set 是由一群性質相似的 Entity 所集合而成的，即同類的 Entity 可以構成一個 Entity Set，例如「員工」為一個 Entity Set，而張三、李四皆為員工，則稱張三、李四為該 Entity Set 內的一個 Entity。

符號：Entity Set 會以一個矩形圖（符號）代表之，並將此 Entity Set 名稱填入矩形中，通常採用名詞的方式來命名該名稱。

例如：如圖 12 所示，有學生、課程、老師、部門、物料等實體集合。

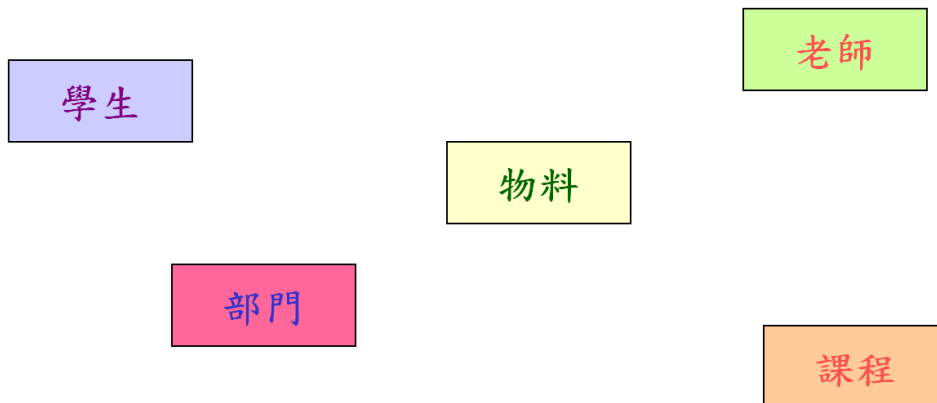


圖 12：實體集合（Entity Set）

## （2）關係體（Relationship）

意義：任一個 Entity Set 與另一個 Entity Set 相互作用的結果會產生 Relationship，且這種 Relationship 可能不只一種。

觀念：沒有相互作用就不會產生任何 Relationship，學生 Entity Set 與課程 Entity Set 相互作用的結果會產生選課 Relationship，假如學生來學校卻不上課，則與課程無關，此時學生與課程為兩個獨立 Entity Set，無任何 Relationship 存在。

符號：Relationship 會以一個菱形圖（符號）代表之，並將此 Relationship 名稱填入菱形中，通常採用動詞的方式來命名該名稱。

例如：如圖 13 所示，有編班、選課、住院、聘僱、授課等關係體。

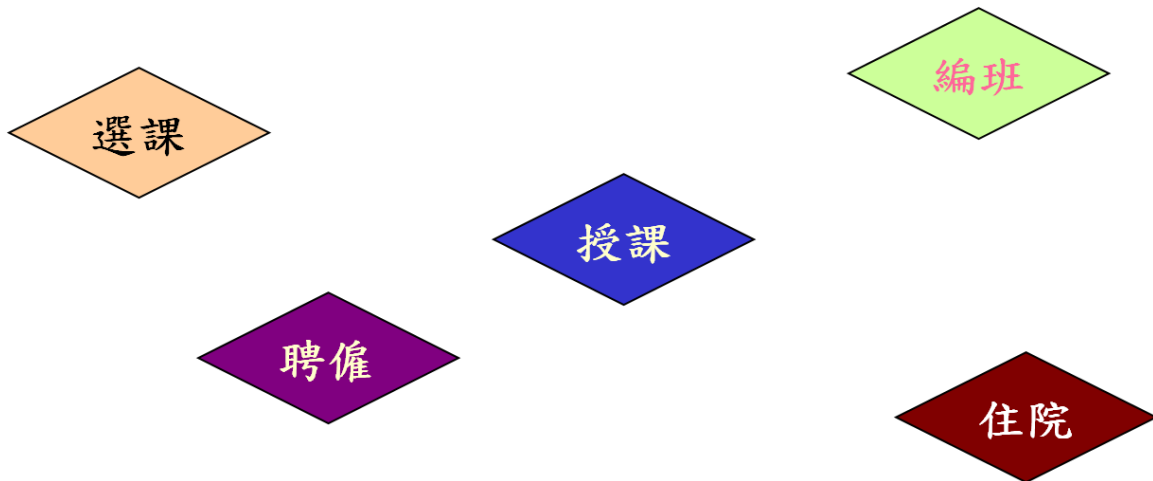


圖 13：關係體 (Relationship)

(3) 連結數 (Connectivity)

意義：說明 ERD 圖中資料之間的對應關係。

觀念：只有三種對應關係，一對一關係 (1 To 1)、一對多或多對一關係 (1 To Many or Many To 1) 以及多對多關係 (Many To Many)。

符號：將 Connectivity 標記於箭頭附近。

例如：如圖 14 所示，有學生與課程的多對多選課關係、部門與員工的 1 對多聘僱關係以及病人與病床的 1 對 1 住院關係。

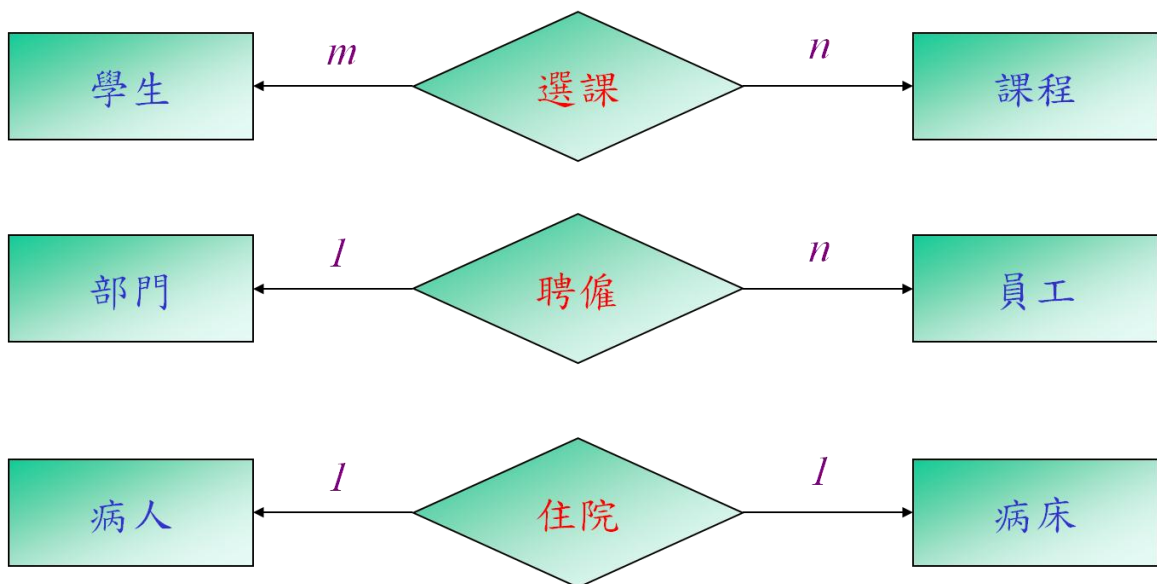


圖 14：加入連結數 (Connectivity) 的簡易 ERD 圖

附註：連結數 (Connectivity) 該如何決定，可以用聯集的概念來判定，

如圖 15 所示，步驟分述如下：

Step 1 先以 1 的觀念由左邊到右邊 (From Left To Right) 分析一遍

Step 2 再以 1 的觀念由右邊到左邊 (From Right To Left) 分析一遍

Step 3 最後採用集合理論中聯集的觀念取最大值 (Max)

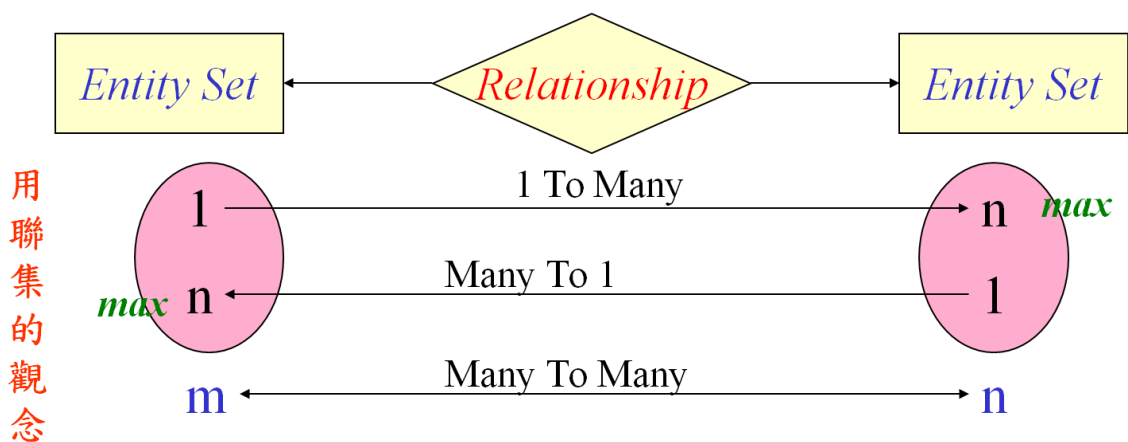


圖 15：連結數決定概念圖

#### (4) 屬性 (Attribute)

意義：用來描述並說明 Entity Set 或 Relationship 的性質或特徵。

觀念：需要在眾多的屬性中挑選一個甚或一組的屬性，當作此 Entity Set 或 Relationship 是獨立可區別的事與物，稱為主鍵屬性，觀念上與資料庫中關聯表地的主鍵 (Primary Key) 相同，隱含著唯一性的概念。

符號：以一橢圓形或圓形符號表示之，並將此 Attribute 名稱填入橢圓形中，同時在代表著唯一性屬性的地方標示底線。

例如：圖 16 所示，以教師與課程的多對多授課關係為例，教師 Entity Set 以姓名 Attribute 為 Primary Key，課程 Entity Set 以課程名稱 Attribute 為 Primary Key，授課 Relationship 以姓名 + 課程名稱複合 Attributes 為 Primary Key。

但是通常會遇到同名同姓以及課程名稱相同的問題，因此引進教師編號 (teacher\_no) 以及課程編號 (course\_no) 兩個屬性當主鍵屬性，如圖 17 所示。

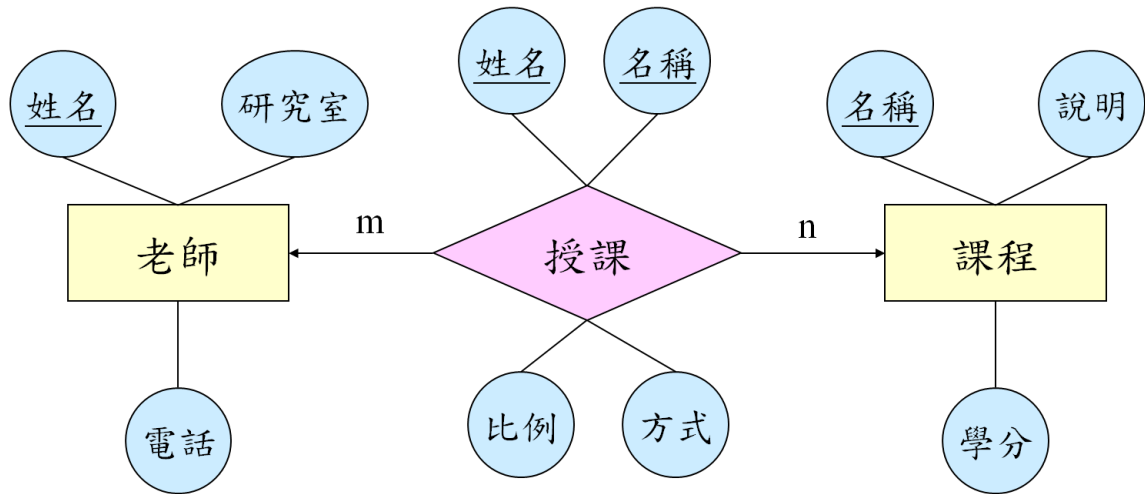


圖 16：完整的 ERD 圖(I)

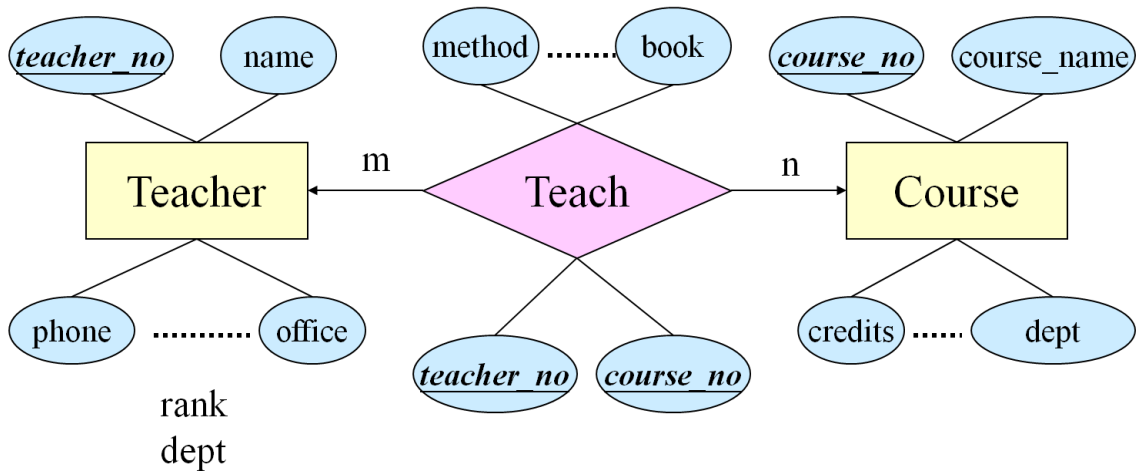


圖 17：完整的 ERD 圖(II)

根據上述的說明，嚴格說起來整個 Database 中就只存有兩樣東西，不是 E（指實體集合 Entity Set）就是 R（指關係體 Relationship），對於已經具有完備企業整體性觀點的 ERD 圖的組織，Peter Chen 曾指出，通常模型圖中的 E 與 R 皆已經達到第三正規化型式 (Third Normal Form；簡稱 3NF)，此時可以針對模型圖中每一個 Entity Set 都必須建立相對應的

主檔 (Master File)，而針對每一個 Relationship 都必須建立相對應的異動檔 (Transaction File)，如圖 18 所示，依照三層資料庫管理系統的觀念來看，希望達成資料獨立 (Data Independence) 特性的理念，在 ER Model 出現後讓整個三層資料庫管理系統架構更具完整性。

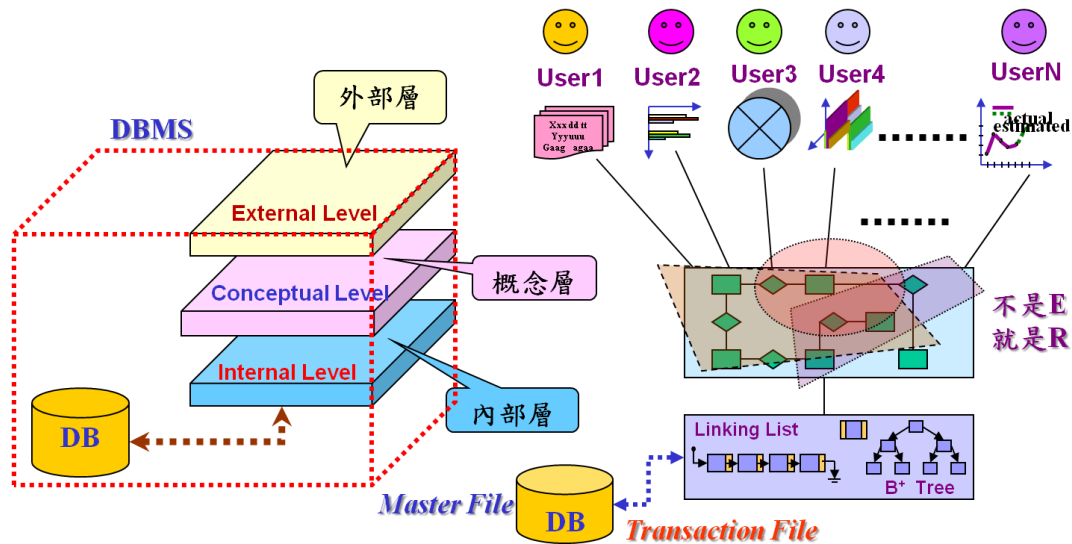


圖 18：三層資料庫管理系統架構與 ER Model 關係

實務上要如何找尋 Entity Set 一直是資訊人員的困擾，一般而言可從人 (Person)、地 (Place)、物 (Thing)、觀念 (Concept) 以及事件 (Event) 五個方面來尋求，下面以某一大學業務系統中尋求 Entity Set 為例，分述如下：

- (A) 人 (Person) 的部份，例如學生、校友、教職員、申請者以及工友。
- (B) 地 (Place) 的部份，例如大樓、教室、學生宿舍。
- (C) 物 (Thing) 的部份，例如設備與物料。
- (D) 觀念 (Concept) 的部份，例如課程、班級以及職等。
- (E) 事件 (Event) 的部份，例如帳目與課外活動。

若以 ERD 製作流程觀之，可以將上述理論觀念修正後整理成七個步驟，如圖 19 所示，從問題需求面的瞭解，到 ERD 藍圖的定義出，至最後因工作不確定性與新增加業務等需求的變動所必須做的調整步驟，可以提出做為初學者在資料庫系統規劃與設計的方

針。

## • ERD 的繪製步驟

STEP1: 瞭解 *Problem*

STEP2: 尋找 *Entity Set*

STEP3: 分析 *Relationship*

STEP4: 定出 *Connectivity*

STEP5: 條列 *E* 與 *R* 的 *Attribute*

STEP6: 決定 *E* 與 *R* 的 *Primary Key*

STEP7: 調整.....

一步  
一  
腳  
印

圖 19：ERD 的繪製步驟

### 3.3 企業組織處理資料的未來

根據本章開頭所陳述的論點來看，實務界存在著新技術與新運用的殷切需求正無限度地不斷漫延開來，尤其是當企業為了在全球競爭環境下想要持續地擁有良好經營優勢，勢必會搭企業 e 化的這一班列車，例如消除或減低整條供應鏈上因為資訊無法分享而造成的需求不確定性問題所提出的供應鏈管理(SCM)e 化資訊系統，或是追求與顧客或消費大眾維持良好互動關係的顧客關係管理(CRM)e 化資訊系統，甚或以個別企業資源整合為基礎的企業資源規劃(ERP)e 化資訊系統，即在實務界或學術界常常都可以接觸到的 SCM、CRM、ERP 等名詞，目前這些名詞都被各大企業統稱為 eBusiness Applications，而不管是否在技術面採取何種資訊科技架構（例如傳統大型主機集中式架構、N-Tiers 的主從式

架構甚或是分散式架構)，但最重要的是一定要有一個穩定的管理資料與即時提供完整資料的機制存在，至於是否採用物件導向的技術標準，至今仍然眾說紛紜，較少成功案例可以說明，無論如何在此資料庫系統就扮演著一個非常吃重的角色，根據一項 Internet 未來世界潮流中人類消費特性調查報告中指出，不會使用電腦與網際網路的使用者勢必會寸步難行且會事事窒礙難行，在網際網路上買商品、買車票遠行、買電影票休閒、必須在 Internet 上找工作以及學生註冊上學等等目前人為工作事項都會一項一項地在網際網路上完整的實現，屆時不會使用 Internet 的人就很難適應一般大眾化的生活，但值得注意的一點，若是資訊人員無法事先將完整且穩定的資料庫系統建置好，這些理念都將成為想法而無法實現，所以一個良好資料庫規劃與設計的工作完備被視為企業生存的重要指標。

## 第四節 個案討論

每當學習過資料庫管理課程或資料庫系統課程的初學者都常常會問一些問題，「這些理論學了之後要如何使用？什麼時間點要用哪一個(套)理論，為什麼？到業界工作之後真的會用到這些理論嗎？未來去應徵有關 ERP、CRM、SCM 等相關工作時，這些理論還是可以用嗎？聽說不是都已經改成採用物件導向(Object-Oriented, OO)技術了嗎？……」，說真的，筆者也真希望有人能夠給予完整的答案來釐清這些問題，在此舉一導入 ERP 系統的個案 A 公司為例來說明資料庫設計理論的使用時機。

A 公司為一家在中部的製造業公司，A 公司原先已經存有支援營運的資訊系統，但是在早期 A 公司內部各單位的資訊系統均為各單位獨立發展軟體或採購現成的套裝軟體(Package)，為了順利取得國外客戶公司代工的訂單，因此 A 公司被國外的客戶公司要求現有許多獨立的資訊系統軟體必須重新整合，或者 A 公司也可以升級為具有即時(Real Time)、整合(Integration)資訊能力的 ERP 系統，然而因為 A 公司產業的特殊性，因此並沒被國外客戶要求需要國外的哪一套 ERP 軟體，當 A 公司之前已經擁有許多資訊系統如下，人事資訊系統、薪資資訊系統、採購資訊系統、倉儲資訊系統以及財產資訊系統等，每一資訊系統皆有支援該資訊系統的資料庫機制，換言之，A 公司目前擁有許多資料庫系統被儲存各個業務面活動的執行資料，因此各個資料庫中的資料之間彼此並無完整的關聯性存在，甚至有出現許多衝突造成公司人力與物力在運用上的重疊和浪費，例如員工編號欄位在人事資訊系統中為文字型態、長度為 6 碼(096006)，但是在薪資資訊系統上卻是文字型態、長度為 8 碼(10096006)，在會議討論 2 個月後，經由多次的評估最後決定量身訂做重新打造屬於適合公司特性的 ERP 系統，於技術面採用具有 3 層主從式(Client-Server)架構的程式語言自行開發 ERP 系統，以微軟公司的 ASP.NET 程式語言開始設計並規劃導入此量身訂做的 ERP 系統，而存放資料的資料庫系統決定採用甲骨文(Oracle)公司的資料庫管理系統產品，至此已敲定並完成未來技術面的雛型架構，另一方面，由於該公司已經導入 ISO 認證制度，包含資訊部門的業務執行也都已經納入 ISO 的稽核重點，因此 A 公司在本次 ERP 系統的導入工作也必須遵循資訊系統軟體開發的 ISO 制度的規定，亦即該要有的步驟一個都不能省略，以及步驟順序也不可以任意調動，該填具的文件都必須於規定工作時間內完成(即 ISO 制度有規定不能因為趕時程先進行工作，



等事情完成後才回頭補填文件)，如此才不會被稽核 ISO 制度的組織發現而記上一筆業務執行缺點，例如，此 ISO 制度中，在資料庫方面規定必須依序完成各系統的資料字典 DD (Data Dictionary) 以及實體關係體圖 ERD 兩項文件，至於正規化技術(Normalization)只需要在關聯式資料表規格書(Table Specifications)中有提及要達成第三正規化(Three Normal Form, 3NF)即可，接下來透過描述 A 公司的當時重點工作項目，讓初學者對於資訊系統建置的過程有概念性的印象。

首先 A 公司將各單位的主管集合起來，由 A 公司老闆親自在會議中說明並宣達公司即刻啟動 ERP 導入專案，之後由一位資深系統分析師(System Analyst, SA)接管會議確認全公司的業務流程，經由詢問一一導引出各單位所掌管的業務以及業務之間的前後順序關係，同時請一位專案經理(Project Manager, PM)協助使用會議室中數位設備投射列示出各單位所掌管的業務有那些，以及這些業務執行的前後順序，當確認全公司的業務流程圖後，接著再由各單位業務資深承辦人協助 SA 繪製該每一項業務的詳細活動內容，該公司在此部分以 SAP 公司發展的流程模擬工具 ARIS(Architecture for Integrated Information Systems)的事件驅動流程鏈圖(Event-driven Process Chain, EPC)圖來表達，繪製過程中同時考慮目前使用的企業流程是否合理，討論企業流程化繁為簡的可行性，流程的討論的過程中會遇到跨單位的活動執行，當各單位主管意見不同時，透過專案經理辦公室(PM Office)提報督導委員會(Steering Committee)仲裁後確認企業流程狀況，最後再將所有確認過的 EPCs 圖整合串接起來，決定出需要資訊化的業務項目，至此企業流程規劃工作已經完成，未來 A 公司使用的 ERP 系統各個模組的詳細活動內容清楚地定義出來，並以此作為 ERP 導入的企業藍圖(Business Blueprint)，接著將此藍圖轉交給下一階段接手的系統設計師(Systems Designer, SD)與程式設計師(Programmer, PG)資訊技術(IT)人員，進入了系統設計的階段。

接獲企業藍圖之後，SD 人員開始進行 ERP 系統的細部設計與資料庫設計，ERP 系統的資訊流(Data Flow)是設計重點，資訊流所涵蓋的資料項目必須具有唯一性，因為會影響後續資料庫設計的內容，譬如有同名異義的問題存在，在許多業務中經常都會有價格這一個欄位，但是銷售業務與採購業務的價格欄位是具有不同的管理意涵，必須區隔開來分別給予不同欄位名稱，接著請相關單位討論資料規格部分事項，包含資料欄位名

稱(Field Name)、資料型態(Data Type)、長度(Length)、是否有預設值(Default)、是否允許虛值(Null Value)、是否設定為主鍵值欄位(Primary Key, PK)、此欄位是否要建立索引(Index)等，並產生資料字典(Data Dictionary, DD)，過程中允許修正 DD 內容，但最後必須凍結 DD 不能再修改，沒問題後就依照 EPCs 圖與 DD 文件產生 ERP 系統的 ERD 圖，以此 ERD 圖做為設計資料庫的基礎，接著就由業務單位使用者告知資訊技術人員將來工作的 ERP 系統軟體希望的使用需求畫面(Screen)並由資訊技術人員產生資訊畫面，為了保證資訊畫面無誤，A 公司再一次召集相關業務單位討論資訊畫面是否正確達到業務需求並檢視 ERD 圖是否需要修改，如果無誤就執行 Oracle 資料庫管理系統建置工作，根據 ERD 圖進行資料庫中資料表(Tables)的建立，並展開後續程式設計(Coding)以及程式完成後的軟體測試(Testing)的工作，程式並非一次撰寫就會如期完成，必須經過不斷地修正程式才會有穩定的應用軟體系統產生，為了考慮周全還必須進行全部模組功能整合測試的工作，以及使用者使用驗收測試，同時間會開設 ERP 使用的教育訓練課程，一切無誤後會執行 ERP 系統環境安裝以及執行資料轉檔工作，檢查無誤後可以挑一個時間通知相關業務單位舉行上線說明會，隨後舉行系統上線(Go-Live)，並且讓該 ERP 系統進入系統維護階段。

該個案 A 公司雖然是導入量身訂做的 ERP 系統，但是從導入過程中不難發現資料庫機制的產生也必須要有章法、程序與步驟，資料庫內的資料不會無中生有而產生，由個案知是由先行的 EPC 流程分析得來，再透過 DD 的確認讓資訊人瞭解到哪些資料項目未來會轉變為資料庫中的資料欄位，而在 ERD 圖的產生後會要求先經過 3NF 的資料分類歸納後才將資料庫中的關聯式資料表全部確認出來，這些過程雖然在資料庫課程中斷斷續續被介紹出來，但是實務上該如何使用？相信讀者都已經在此個案描述的內容中清楚驗證，必須與其他工具搭配以及做稍微修正才可以落實在任何 e 化系統的建置，至於是否有另外一個可以解決的方案，有待後續研究。

## 習題

### 選擇題

- 1.( ) 在當今重視國際化競爭與建立完整企業 e 化環境的氛圍下，什麼系統已經成為每家企業在追求競爭優勢與營運效率過程中不可或缺的基石 (A) ERP (B) SCM (C) CRM (D) MIS 答案 A
- 2.( ) 在大型的 ERP 系統中，此整合的資料庫通常會包含有大約 2 萬張的資料表(Tables)，這一套完整的資料庫(包含這些數萬張的資料表)在導入 ERP 系統時，是如何建置完成 (A) 不需要同時 (B) 同時 (C) 過幾年後 (D) 邊使用便建置 答案 B
- 3.( ) ERP 導入過程中，當資料庫綱要內容都已經確定下來後，接下來由 ERP 團隊的顧問規劃完整轉檔計畫，交由公司的哪一部門負責將舊系統(Legacy System)的資料轉入 ERP 系統的資料庫中 (A) 倉儲 (B) 業務 (C) MIS (D) 會計 答案 C
- 4.( ) 當企業流程與系統功能不搭配時，需要馬上想到解決方法來沉著應戰，方法上有五種，哪一種方法應極力避免，因可能導致系統日後無法升級 (A) 從 ERP 系統標準功能中再找尋一次 (B) 如果是 ERP 系統中未設計的欄位，可否用其他欄位代替 (C) 在企業流程與 ERP 系統功能間找到妥協，找出折衷的方法 (D) 直些修改 ERP 系統程式 答案 D
- 5.( ) 資料庫管理系統是指 (A) DBS (B) DBMS (C) DB (D) DD 答案 B
- 6.( ) 下列哪一個選項不是資料庫系統(DBS)的組成 (A) 軟體 (B) 硬體 (C) 資料 (D) TCP/IP 答案 D
- 7.( ) 下列哪一個選項不是資料庫系統(DBS)的使用者組成 (A) DBA (B) DC (C) End User (D) 程式設計師 答案 B
- 8.( ) 資料庫中與 ERP 人員溝通時最常用的觀念為 (A) 主鍵欄位 (B) 候選鍵欄位 (C) 超鍵欄位 (D) 替代鍵欄位 答案 A
- 9.( ) 資料庫中與 ERP 系統人員溝通時除了主鍵欄位外，另一個重要的欄位是 (A) 候選鍵欄位 (B) 外來鍵欄位 (C) 超鍵欄位 (D) 替代鍵欄位 答案 B

- 10.( ) 通常實體關係體資料模式工具是使用在資料庫的哪一層次設計 (A) 外部層 (B) 內部層 (C) 概念層 (D) 實體層 答案 C
- 11.( ) 在資料庫個案中 A 公司是因為什麼原因需要導入 ERP 系統 (A) 與外國公司接單 (B) 雲端運算環境盛行 (C) 網際網路傳輸問題 (D) 穿戴式行動設備盛行 答案 A
- 12.( ) 在資料庫個案中 A 公司進入系統設計階段經常發生爭議必須釐清，譬如銷售業務與採購業務的價格欄位是具有不同的管理意涵，此即 (A) 業務人員抗拒問題 (B) 同名異義的問題 (C) 網際網路傳輸問題 (D) 異名同義問題 答案 B
- 13.( ) 在資料庫個案中 A 公司 ERP 專案導入結束時有一個專有名詞 Go-Live 是指什麼活動 (A) ERP 系統上線活動 (B) 程式測試工作 (C) 需求分析工作 (D) 系統設計工作 答案 A
- 14.( ) 在資料庫個案中 A 公司業務單位使用者將來工作的 ERP 系統軟體希望的需求畫面(Screen)是由誰來完成設計 (A) 軟體公司老闆 (B) 使用 ERP 的業務單位使用者自己 (C) 軟體測試人員 (D) 資訊技術(IT)人員 答案 D
- 15.( ) 在資料庫個案中，A 公司的 ERP 系統為量身訂做方式導入，系統設計階段中有一份很重要的資料字典(Data Dictionary)文件簡稱為 (A) DA (B) DB (C) DC (D) DD 答案 D
- 16.( ) 在資料庫個案中，A 公司的 ERP 系統為量身訂做方式導入，當 ERP 導入過程時專案組織中有一個很重要的單位督導委員會(Steering Committee)，此單位的角色是 (A) 出錢單位 (B) 虛設單位 (C) 有問題或衝突時的仲裁單位 (D) 稽核單位 答案 C