

資料庫管理系統在圖書館的利用

陳昭珍

The Application of DBMS in Librarians' Environment

Chao-chen Chen

ABSTRACT

Database Management System (DBMS) is the major tool for computer system design nowadays. For a library, bibliographical database is the core file to support all related operations, if the integrated design of bibliographical database is adopted for a library, then the redundancy, incompleteness, inconsistency of the library records can be avoided.

In this paper, we first introduce the definition, the development, the functions, and the structures of the DBMS, in which the hierarchical model, network model, relational model of DBMS is illustrated with the library files. Then, we explain how the ideas of DBMS is used to design integrated bibliographical database.

一、資料庫管理系統之定義與發展

資料庫 (database) 是一羣有組織、有系統、而且彼此相關的資訊 (information) 或資料 (data) 所組成的集合體。database 一字亦有寫作 database, 或 data-base 者, 到底孰是孰非, 至今仍未有定論。^① 除上述籠統的定義外, 凡稱為資料庫者必定具有下列屬性:^②

1. 資料庫是一些具有固定意義之資料在邏輯上相關連的集合體;
2. 隨意混合在一起之資料, 不能稱為資料庫;

* 著者現為臺灣大學圖書館學研究所博士班研究生。

3. 資料庫之設計、建立與處理均有其特定之目的，而其資料乃就某一羣使用者之需求而蒐集組織的；

4. 資料庫亦可稱為一小世界 (mini-world)，它代表真實世界的某些觀點 (views)，有關該小世界的變化也將反映在此資料庫中。

資料庫管理系統 (database management system, 簡稱為 DBMS) 簡單而言就是在管理上述資料庫的系統。根據葛羅斯 (Audery N. Grosch) 的定義, DBMS 即是以相關檔案 (related files) 構成一資料庫, 提供不同目的、不同使用者利用, 換言之, 當一 DBMS 在線上作業時, 可讓使用者同時利用不同之應用系統檢索資料, 並產生不同目的之產品。^③ 華格納 (Judy Wagner) 則簡單明瞭的將 DBMS 定義為一維護、貯存及檢索在資料庫中之記錄的套裝軟體。^④ 周 (Chow, J. V.) 認為 DBMS 是一種檢索方法 (access method), 透過這種方法可以讓不同使用者之應用程式處理資料庫。^⑤ 而派特森 (Patterson, A. C.) 將 DBMS 描述為一提供貯存及檢索資料庫之人員 (staff)、軟體功能、硬體功能的組合體。^⑥ 各家說法雖略有差別, 不過其本質之描述則一, 簡而言之, DBMS 是指一組可以讓使用者建立及維護資料庫的程式。

DBMS 之發展約始於 1960 年代, 促使 DBMS 發展的原動力是因為當時已發展出以直接存取的磁碟貯存 (Direct-access disc storage) 取代循序性的磁帶貯存方式 (sequential tape storage), 新的貯存方式雖然可以使得資料的查尋更迅速有效, 但是也相對地提高了每一筆記錄的貯存成本, 因此以不重複貯存資料為原則的 DBMS 乃應運而生。^⑦

二、資料庫管理系統之功能與結構

1 資料庫管理系統之功能

在 DBMS 未被大量應用以前, 一般的電子資料處理 (Electronic Data Processing, 簡稱為 EDP) 均是由各應用程式本身建立起該程式本身建立起該程式所需的資料結構。由於各應用程式所需用之資料項目並不一致, 而各程式設計師亦各有其設計偏好, 是以某一應用程式所建立之資料檔往往很難與另一應用程式共用, 如此一來便造成了許多困擾。一般說來, DBMS 與傳統之檔案處理大致有下列五點差異:

a. 就檔案維護之難易性而言: 傳統之檔案結構因與應用程式結為一體, 日後若須修改程式或資料結構時, 往往牽一髮而動全身, 尤其當原程式

設計者離去後，繼任之程式設計師更不易接手，而被迫放棄原系統另起爐灶，造成人力財力上的浪費；但在 DBMS 中，由於其資料結構與應用程式各自獨立的特性（稱為 Program-data independence），使得應用程式之發展及日後之維護工作得以簡化。

b. 就資料之重複性而言：傳統的檔案處理方式由於每一個應用程式均需建立其專屬的資料檔，許多相類似的系統間往往有大量的資料重複儲存，造成空間的浪費，並減緩系統運作之速度，同時若有相同之資料必須更新時，由於必須同時更改好幾個檔，若有不慎，則會造成檔內記錄的不一致；而在 DBMS 中，因為所有的資料都集中管理，相同的資料只需儲存一次，降低儲存成本，增加檢索效率，同時也易於維護資料之正確性與完整性。

c. 就使用者瞭解的難易度而言：DBMS 提供使用者不同層次的資料模式 (data model)，這種資料模式採用邏輯概念，如檔案之關係、性質、目的等，概念性的表現資料，讓使用者易於了解檔案結構，而不必去管它的硬體貯存方式；而傳統的方式除了設計者外，不懂電腦的人根本無法明瞭其設計理念。

d. 就利用的方便性而言：DBMS 允許使用者由不同的角度利用資料，同時由於 DBMS 與人工智慧 (AI) 的結合，使得它可以靈活的組合相關資料，回答非事先設計好的問題（即非預期性的問題），這也是傳統的檔案處理方式所無法做到的。

總而言之，一個完整的 DBMS 系統通常包括下列七點功能：②

1. 控制重複 (controlling redundancy)
2. 分享資料 (sharing of data)
3. 限制未經授權的檢索，以維護資料之安全。(restricting unauthorized access)
4. 提供多種使用者介面，如查尋語言 (query language) 供檢索者、程式語言供應用程式師、固定格式供例行使用者，逐項式介面供初學者，自然語言 (natural language) 供主題檢索使用等等。
5. 能表現出資料間複雜的相關性。
6. 執行整體性限制 (integrity constrain) 以維持資料之正確性。
7. 提供備份與復原系統。

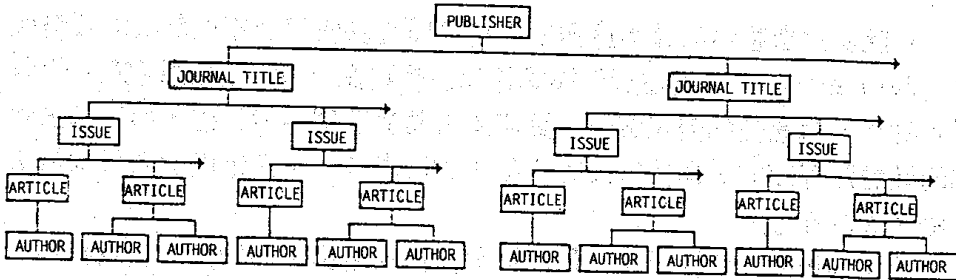
2 資料庫管理系統之資料庫結構

DBMS 之資料庫結構（或稱為資料庫模式 database model）依其發

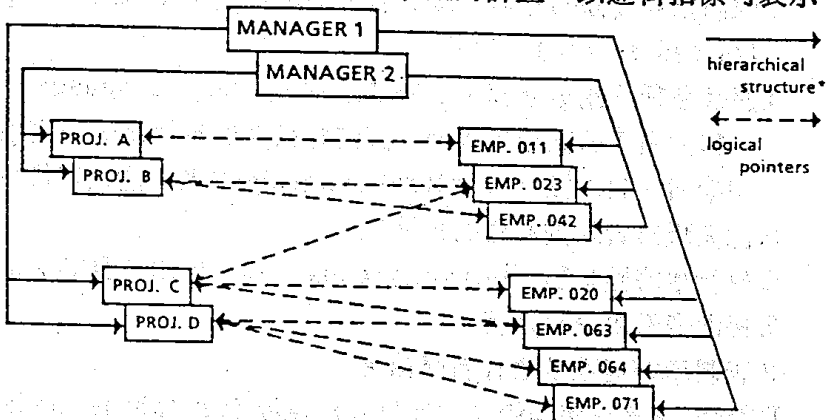
展之先後可分為三種方式：階層式模式 (hierachical model)、網狀式模式 (network model)、及關連式模式 (relational model)，其中以第三種模式，將資料規格化 (normalization)，分解資料之關係 (decomposition) 使之單純化，以便靈活、彈性的連接與維護，功能最強，為目前 DBMS 系統設計之主流，茲分別舉例說明如下：

a. 階層式資料庫：

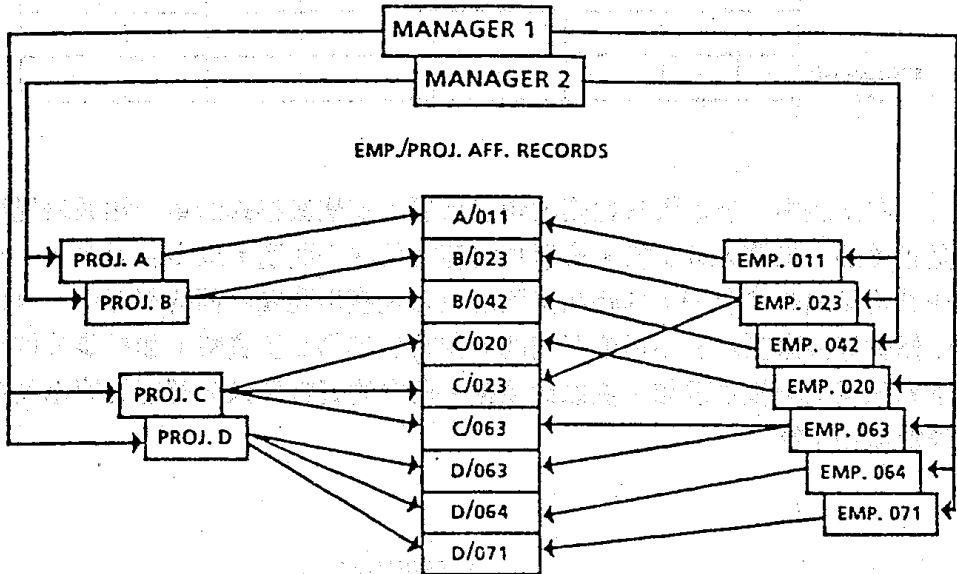
階層式模式又稱為樹狀模式，其觀念主要導源於人類親 (parent) ——子 (child) ——兄弟 (sibling) 之間的關係。以期刊之出版為例，一出版社可能發行好幾種期刊，每一期刊包括好幾個卷期，每一期有好幾篇論文，每一篇論文有一個以上的作者，這種關係可以階層式模式表示如下圖：⑨



階層式模式最大的缺點在於它較難處理多對多 (many to many) 的資料關係，如一篇文章可能有多位作者，而一個作者也可能有多篇文章，這種關係在階層式模式中可能的處理方式之一是重複建檔，即需另外再建立一作者 ↔ 論文檔。第二種處理方式就是給予邏輯指標 (logical pointers) 來指示其邏輯關係，如一公司中，每一計畫之下會有多位工作者，但某些工作者也可能同時從事二種以上的計畫，以邏輯指標可表示下圖：⑩

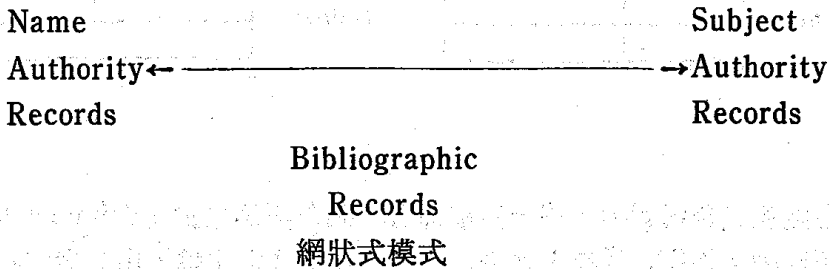


第三種解決方案是增加交叉性記錄 (intersection records)，如下圖①，這種交叉性記錄的功能和邏輯性指標相類似。



b. 網狀式模式

網狀式模式對於記錄之間的連接，和階層式模式相似，只不過網狀式模式可以任意連結記錄。也可以說網狀式模式是一具有交叉性記錄的階層模式，再去掉階層而已。以圖書館之作業為例，其書目檔、名稱權威檔、標題權威檔之關係，可以網狀式模式表示如下：②



c. 關連式模式

關連式資料庫模式以二維平面表 (two-dimensional tables) 來排列資料，橫排 (row) 代表記錄之實體 (entities)，縱列 (columns) 代表記錄之屬性 (attributes)，如一簡單的書目資料檔可列表如下：③

ATTRIBUTES

	ISBN	TITLE	AUTHOR	AFFILIATION	PUBLISHER	PLACE	DATE	CALL #
Entities:	-	-	-	-	-	-	-	-
Bibliographic Items	-	-	-	-	-	-	-	-

在上表中，若以資料檢索的角度來看，這種聚集式的單一關係表現法是正確的，但深入分析後，會發現它存在着許多問題，以出版者而言，其住址可能隨時會改變，同時該出版社也會重複出現在不同的記錄中，只以一個關係表來表現，不但浪費空間，同時也不易更正資料，如果能分解成下列三個檔則較為理想，並以此類推，所有的資料皆以分解至最簡化最單純為原則：④

ATTRIBUTES

FILE 1	ISBN	TITLE	AUTHOR	PUBLISHER	PLACE	DATE	CALL #
Entities							

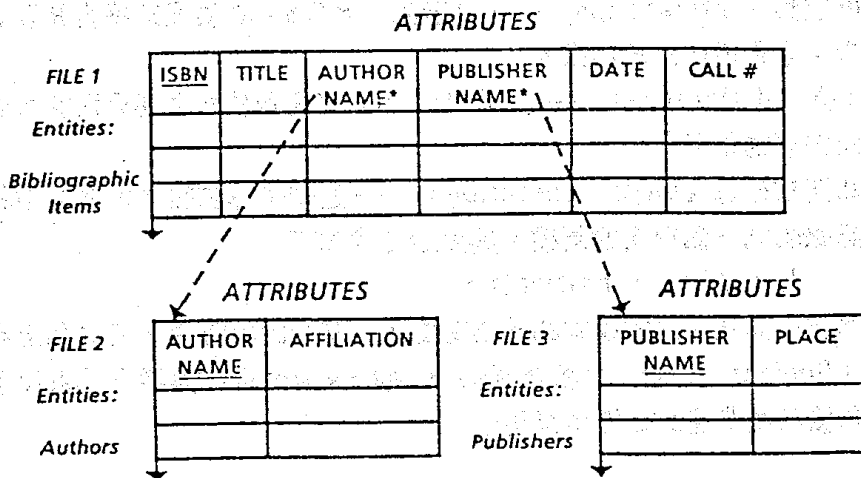
ATTRIBUTES

FILE 2	AUTHOR	AFFILIATION
Entities		

ATTRIBUTES

FILE 3	PUBLISHER	PLACE
Entities		

在關連式資料庫中，每一個檔都有一欄資料為主鍵 (Primary key)，如下圖中，ISBN 為檔 1 之主鍵，作者為檔 2 之主鍵，出版者為檔 3 之主鍵，對檔 2 及檔 3 而言，檔 1 之作者與出版者則為檔 2、檔 3 之外鍵 (Foreign key)。在 DBMS 中，鍵的觀念和圖書館所習用的檢索鍵不同，在一關連式檔中，鍵是檔與檔之間連結的關鍵，該鍵可以是一個檢索鍵，但其他不是鍵的欄位，只要有需要，也都可以當做檢索鍵。⑤



此外，在關連式模式中另一個需要說明的觀點是依存關係 (dependency)，也就是說某一欄位之值乃依另一欄值而定，如作者之學科興趣，服務單位等資料，是依作者是誰而定，和 ISBN 無出關，出版社之住址依出版者而定，這種依存關係，在分解檔案時必須分解出來獨立建檔，如另外建立作者權威檔、出版者權威檔等，這也是利用分解檔案法來減少資料重複建檔的一種。

除了資料規格化與檔案分解法外，要解決操作 (manipulation) 及查尋 (searching) 的問題，必須借助於關連式代數 (relational algebra) 及關連式計算 (relational calculus) 來處理。關連式結構基本的檔案操作稱為關連式代數，它所包括的基本運算方式有下列幾種：^⑩

a. 選擇運算 (select operation) —— 從一檔案中 (原已存在的檔，或新組合的檔)，檢索出符合選擇條件的記錄 (稱為 tuples)。其表示方式如下：

$$\delta \langle \text{selection condition} \rangle (\langle \text{relation name} \rangle)$$

b. 映射運算 (project operation) —— select 主要在找出符合條件的橫欄資料，而映射則在找出符合條件的縱列資料，一般敘述如下：

$$\pi \langle \text{attribute list} \rangle (\langle \text{relation name} \rangle)$$

c. 羣論運算 (set theoretic operation) —— 這種運算是標準的數學羣論運算，其方式包括：

(a) 聯集 (union) —— 表示如：R U S，凡符合 R 或 (or) 符合 S 之條

件者，皆為所要之結果。

(b)交集 (intersection) ——表示如： $R \cap S$ ，必須同時符合 R 及 S 之條件者，才是所要求的結果。

(c)差 (difference) ——表示如 $R - S$ ，即凡屬於 R 之部分而非 S 者，才是所需要的結果。

d. 連結運算 (union operation) ——連結運算在組合兩個檔中相關連的記錄成爲一個單獨的新檔，其表示方式如下：

$$R \infty \langle \text{join condition} \rangle S$$

連結運算是關連式模式的核心功能，在結合的過程中，可以加上布林邏輯 (Boolean logic) 來組合資料，連結運算使得檔與檔之間可以互動，這種機制使關連式結構所以得名。

三、資料庫管理系統在圖書館的利用

資料庫一詞在圖書館中，早已被廣泛的利用，而其爲不同使用者分享的特性，也非常符合電腦處理資料庫的觀點。葛羅斯曾指出，如果圖書館作業對下列問題之答案是肯定的，那麼 DBMS 應該非常適合在圖書館中應用：^①

1. 分散在各檔中的資料是否會導致資料的重複與不一致？
2. 這些重複存在的資料是否使得資料之更新與正確性之維護產生問題？
3. 資料的複雜性是否會延緩新應用程式的建立？
4. 要求一新記錄時是否需大幅度的修改應用程式？
5. 分散在各檔中的相關記錄，對於建立新應用程式有無幫助？
6. 資料查尋主要以隨機或非循序式爲主？

無疑的這些問題的答案，對圖書館而言都是肯定的。不過葛羅斯也批評道，以一般性的DBMS來處理圖書館作業有二個缺點：^②

1. 處理不定長度之記錄有其限制；
2. 製作複雜之輸出格式，如卡片等，仍有問題存在。

圖書館所有處理的資料有很多是不定長度之資料，在馬克阿林等斯 (McAllisters) 之研究中，發現圖書館之資料庫特性不同於一般資料庫者爲：^③

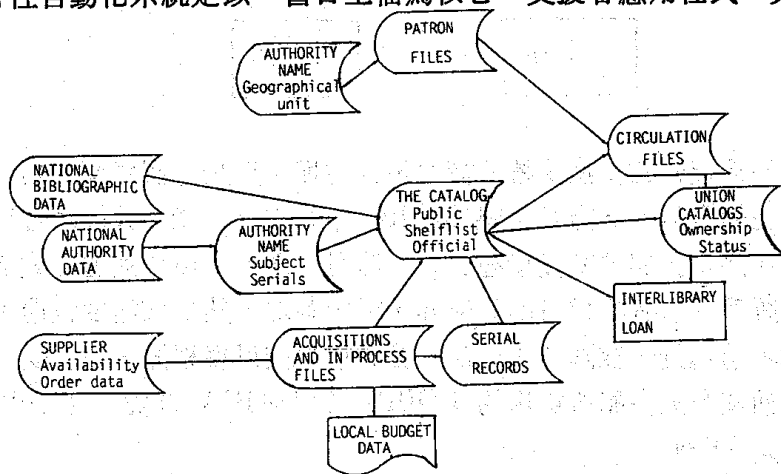
1. 欄位多爲不定長之資料，如書名、作者、標題等；

2. 利用不定長之檢索鍵檢索資料；
3. 有很多種不同類型的檢索鍵；
4. 必須利用權威檔以維護檢索鍵的一致；
5. 必須處理多種文字之資料；
6. 在資料庫中，文獻與文獻之間存在着某種關連。

上述所稱的特性都是針對圖書館書目性資料庫而言，而相關的缺點也都屬於早朝DBMS 功能上的缺失，除了書目性資料庫外，圖書館中尚有非書目性資料檔，如讀者檔、物料檔、書商檔、經費檔、人事檔等，其特性和一般之資料庫並無兩樣，大部分都是屬於定長欄之資料，檢索目的也都可以事先預期，多數圖書館早就以資料庫管理系統處理之，書目性資料庫是圖書館資訊服務的核心，舉凡 SDI、OPAC 等都仰賴於書目性資料庫的邏輯性組合。一個好的書目性資料庫設計時其經濟性的考慮和其他類型之資料庫一樣重要，但一般在設計關連式書目性資料庫時，其特殊問題往往被忽略，以下即針對書目性資料庫之建立與利用再深入說明。

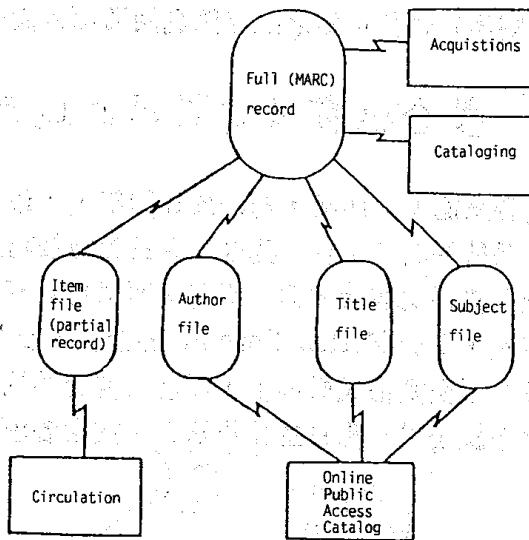
四、整合性書目資料庫的建立

圖書館利用電腦來處理其作業，約始於1960年代，最初設計時皆着眼於流通作業等各別系統的改進，利用孔卡、定長欄等技巧來輸入及貯存資料，每一應用系統之資料結構因其需求不同，而項目多寡不一，無法互相交換。到了1970年代，由於MARC Format 的設計，及DBMS在圖書館的利用使得書目資料之交換格式標準化，而進入整合系統設計時代。基本上，整合性自動化系統是以一書目主檔為核心，支援各應用程式，其結構如下圖：

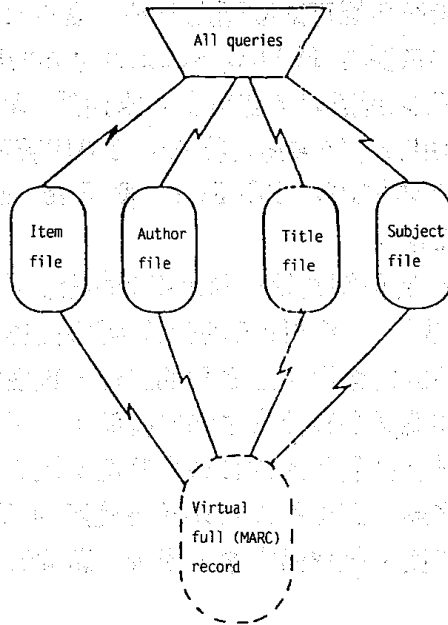


若要實際應用於流通控制及線上公用目錄查尋，除了書目主檔外需另外建立作者檔、書名檔、標題檔、館藏檔等索引檔，各索引檔與主檔之間的關係一般說來有二種方式：

a. 重複建檔：即書目主檔是一個實存檔，其下再分出各別的檢索檔，(參見下圖)如流通檔中有流通作業所必要的資料項，標題檔中貯存簡短的書目資料，或只存記錄識別號，以便與主檔相連，由標題檔中查尋相關主題資料，無疑地較直接由書目主檔入手來的便捷快速，不過相對地也較為浪費貯存空間，一旦資料需要更改時，索引檔之款目也需加以更正並重新排序，這種結構的書目檔乃依每一筆資料庫的編號連續儲存，故稱為線形檔 (linear file)，其輔助的索引檔又可稱為倒置 (inverted file) 這是一種典型的階層式模式，索引檔與主檔之間以鍵看連，叫出既存在主檔內之資料。OCLC 等書目中心，大都採用這種模式。



b. 不重複建檔：採這種結構之書目主檔，只是一個虛擬的檔，實際上並不存任何資料，其方式是將 MARC 格式中之記錄，分散在各別檔中，查尋時再透過控制碼來連接整合。(參見下圖)，由於這種方式必須重組資料，雖然不重複建檔可節省空間，但相對地，其檢索速度較慢，這種資料模式可稱為是一種關連式結構，檔與檔之間以鍵相連，組合成一新檔。國內目前甚多圖書館所採用的 DOBIS 及 URICA 系統，都是屬於這種建檔方式。②①



五、結 語

資料庫管理系統因為必須使用到龐大的記憶體及輔助記憶體，因此在這之前，資料庫系統僅存於大型電腦中，但是，自西元1980年微電腦問世以後，這種情形即逐漸改觀，微電腦以其日益精進的硬體結構，在速度、容量上均足以和往日的大型電腦抗衡，而在早期，只有少數電腦專業人員或資料處理人員才知曉的DBMS，也隨着微電腦普遍利用，遍及銀行、保險、製造、科學界、工程界、政府單位等各業，成爲目前處理資料時，人人不可或缺的工具。同樣地，DBMS在圖書館中也已成爲各種整合性自動化系統設計發展的基本觀念，也有利用DBMS套裝軟體做爲其應用程式發展之基礎者，除了上述之 URICA、DOBIS 等外，WLN (Western Library Network) 即利用 ADABAS 系統來支援其龐大的書目資料庫，Bibliotek Library Services 公司所生產的 Data Retrieval System (DRS) 也是很多圖書館自動化系統所採用的資料庫管理系統，Henco 公司所發展的 INFO 軟體則在英國圖書館界被普遍的採用^②，此外，其他在 PC 上發展的 DBMS 系統更是不勝枚舉，如3BY5、PFS-File、PFS-Report，及 dBASE 等都是功能極強的一般性資料庫管理系統^③，尤其是 dBASE 系統上，利用分解檔案方式建立多層資料表，可以解決多層記錄

之問題，如主題、作者等多屬可重複欄之資料，抽離出來獨立建檔，可解決檢索上的難題，同時只要有 MARC Adaptor 之類的軟體，貯存 MARC 欄位與 dBASE 檔欄位之對照表，即可產生或接受 MARC 格式之資料。另外，尚有一些適用於特殊業務之特殊性資料庫管理系統如 Overdue Writer、Pro-cite、INMAGIC、PC-File+ 或 File Express 等也早已在圖書館界被普遍的採用。

DBMS 的觀念，使得圖書館自動化系統的設計由獨立走向整合，DBMS 在微電腦上的普及，使得經費較小的圖書館也能採用整合性設計系統。選擇 DBMS 系統之原則，最重要的工作，則在於作業研究與系統分析，唯有先透視各作業與各種檔案之間的關係，分析出其最簡化、最有效的連結方式，才能設計出最理想的資料庫管理系統；如果圖書館選擇的是整合性自動化套裝系統，那麼對於各系統所使用的 DBMS 必須特別留心比較，才能選出在速度、效率與成本上皆令人滿意的套裝系統。

參 考 資 料

- ① Caroline M. Eastman, "Database management systems," *Annual Review of Information Science and Technology*, V. 20, (1985) p. 91
- ② Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of Database systems*, (California: The Benjamin/Cummings, 1989), p. 3.
- ③ Audrey N. Grosch, "Fourth generation systems for libraries: the marriage of data base management systems and on-line minicomputer hardware," *Special Libraries*, (July/August, 1977), p. 222.
- ④ Judy Wagner, "Data base management system design for library automation," *Journal of Library Automation*, Vol. 13/1 (March 1980), p. 58.
- ⑤ John V. Chow, "What you need to know about DBMS-part 1," *Journal of Systems Management*, 26, (May 1975), p. 22.
- ⑥ Albert C. Patterson, "Data base hazards," *Datamation*, 18, (July 1972), p. 48.
- ⑦ M. E. D. Koenig "Data relationships: bibliographic information retrieval systems and database management systems," *Information Technology and Libraries*, (Sep. 1985), p. 247.
- ⑧ Ramez Elmasri and Shamkant B. Navathe, *Fundamentals of*

Database Systems, 9. 14-15.

- ⑨ M. E. D. Koenig, "Data relationships: bibliographic information retrieval systems and database management systems," p. 260.
- ⑩ *ibid.*, p. 261.
- ⑪ *ibid.*, p. 262.
- ⑫ Roy Chang, "Relational database management systems: a review," *Technical Service Quarterly*, Vol. 4(4), (Summer 1987), p. 87.
- ⑬ M. E. D. Koenig, *op. cit.*, p. 250.
- ⑭ *ibid.*, p. 251.
- ⑮ *ibid.*, p. 252.
- ⑯ Ramez Elmasri and Shamkant B. Navathe, *op. cit.*, p. 149-157.
- ⑰ Lucy A. Tedd, *An Introduction to Computer-base Library Systems*, 2nd ed., (New York, John Wiley, 1984), p. 37.
- ⑱ A. N. Grosch, *op. cit.*, p. 222.
- ⑲ Safa M. Alak and Sabah A. Ali, "Use of database management systems at the University of Basrah Library," *Program*, Vol. 22, no. 3, (July 1988), p. 275-276.
- ⑳ Deter Brophy... et al., *DOBIS/LIBIS: A Guide for Librarians and System Managers*. (Aldershot, England: Gower, 1990), p. 38-45.
- ㉑ Keith E. Jones and Gavin A. Rea, *URICA: A Guide for Librarians and Systems Manager*. (Aldershot, England: Gower, 1989). p. 31-36.
- ㉒ Lucy A. Tedd, *op. cit.*, p. 37.
- ㉓ Dan Marmion, "Library application of database management systems," *Small Computers in Libraries*, 8(2), (Feb. 1988), p. 14.