

COBOL

檔案技術與資料庫

葉耀明 編著

松崗電腦圖書資料有限公司 印行

COBOL

檔案技術與資料庫

每本定價 220 元整

書號：210132

編著者：葉 耀 明

發行人：吳 守 信

發行所：道明出版社

台北市仁愛路二段一一〇號三樓

總經銷：松崗電腦圖書資料有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255・3930249

郵政劃撥：109030

印刷者：東崗印刷設計股份有限公司

台北市仁愛路二段一一〇號三樓

電話：3930255・3930249

中華民國七十年七月 第一版

中華民國七十二年二月 第二版

本出版社經行政院新聞局核准登記，

登記證號為局版台業字第一七二九號

目錄

簡介

第一章 概論

1-1 資料	3
1-2 檔 (File)	7
1-3 邏輯組織	12
1-3-1 線性結構	12
1-3-2 非線性結構	17
1-3-3 語言資料結構	19
1-4 實體組織	20
1-5 資料作業	22
1-6 資料庫	25
1-7 資料檔的沿革與展望	27

第二章 資料管理

2-1 記憶裝置	30
2-1-1 磁帶機	33
2-1-2 磁帶盤	36
2-2 集區 (blocking)	36

2-3	緩衝 (buffering)	38
2-4	檔的使用	41

第三章 COBOL 循序檔 41

3-1	緒言	41
3-2	個案研究：音樂日誌	43
3-2-1	邏輯分析	43
3-3	COBOL 循序輸入 - 輸出模式	46
3-4	循序檔作業	51
3-5	平衡線法 (Balance Line Algorithm)	54
3-6	個案研究：會計簿記	56
3-6-1	邏輯分析	57
3-6-2	PROFIT 公司會計簿記系統	60
3-7	個案研究：飛彈目標系統	73
3-7-1	邏輯分析	73
3-8	變長錄	79
3-8-1	COBOL 變長錄	79
3-8-2	飛彈目標系統之變長錄	81

第四章 分類 89

4-1	緒言	89
4-2	內部分類	90
4-2-1	插入法 (Insertion Sort)	91
4-2-2	交換法 (Interchange Sort)	92
4-2-3	快速法 (Quick Sort)	93
4-3	外部分類	95
4-3-1	磁盤分類	95

4-3-2 磁帶分類	97
4-4 COBOL 提供的分類	99
4-4-1 基本分類	99
4-4-2 特定錄分類	103
4-5 合併 (Merge)	107
4-6 結論	111

第五章 COBOL 相對檔 113

5-1 緒言	113
5-2 個案研究	115
5-2-1 邏輯分析	115
5-3 相對檔處理：車輛牌照檔	116
5-4 相對檔循序出入	123
5-5 相對檔雜亂出入	133
5-6 結論	139

第六章 錄位址技術 141

6-1 緒言	141
6-2 位址決定	142
6-3 搜尋 (Search)	144
6-4 更新目錄	147
6-5 位址計算	149
6-5-1 鍵 - 即 - 位址法	149
6-5-2 鍵轉換演算法	149
6-5-3 雜湊法 (Hashing)	150
6-5-4 除法雜湊 (Division Hashing)	150
6-5-5 雜湊碰撞	152
6-5-6 解決雜湊碰撞	153

6-5-7 開於位址法 (Open Addressing)	154
6-5-8 鏈結法 (Chaining)	163
6-6 其他位址技術	165
6-7 結論	166

第七章 IBM 直接檔 167

7-1 緒言	167
7-2 循序建直接檔	168
7-3 雜亂出入直接檔	176
7-4 結論	186

第八章 COBOL 指標檔 187

8-1 緒言	187
8-2 個案研究：信託合作社	189
8-2-1 邏輯分析	189
8-2-2 指標檔作業	190
8-3 指標輸入 - 輸出模式	193
8-3-1 設備部 (Environment Division)	194
8-3-2 資料部 (Data Division)	195
8-3-3 處理程序部 (Procedure Division)	196
8-3-4 指令總整理	206

第九章 COBOL 指標實體組織 209

9-1 緒言	209
9-2 指標循序組織	210
9-3 ISAM之資料實體組織	213
9-4 ISAM溢出錄處理	217

9-5 建 ISAM 檔	222
9-6 VSAM	222
9-6-1 錄組織	224
9-6-2 控制段 (Control Interval)	225
9-6-3 建 VSAM 檔資料集	228
9-7 結論	230

第十章 資料結構 231

10-1 定態資料結構	232
10-2 動態資料結構	234
10-2-1 線性結構	234
10-2-2 非線性結構	237
10-3 鏈結串列	240
10-3-1 線性鏈結串列	240
10-3-2 非線性鏈結串列	246
10-4 檔案結構	247

第十一章 檔案查詢 249

11-1 個案研究：汽車出租系統	249
11-1-1 邏輯分析	249
11-2 複合查詢	253
11-3 TRACE 演算法	256
11-4 錄組織技術	259
11-5 二元列表 (Binary Table)	260
11-6 結論	264

第十二章 安全性與完整性

265

12-1 資料完整性 (Integrity)	265
12-1-1 外部確認 (External Validation)	266
12-1-2 內部確認 (Internal Validation)	267
12-2 資料安全性 (Security)	270
12-2-1 內部安全 (Internal Security)	270
12-2-2 外部安全 (External Security)	273
12-3 結論	274

第十三章 檔案設計

275

13-1 檔案組織	275
13-2 實際考慮	282
13-3 檔案設計標準	285
13-4 檔案設計步驟	286
13-5 個案研究：失竊汽車檔	288
13-5-1 邏輯分析	288

第十四章 系統設計與完成

299

14-1 系統設計通論	299
14-1-1 設計品質	303
14-1-2 文件記錄	303
14-2 程式師 (Programmer)	304
14-3 層次規劃 (Structured Programming)	306
14-4 測試與程式確認	311
14-5 程式及設計之其他細節	312
14-6 結論	314

第十五章 資料庫系統

315

15-1 準技術考慮	315
15-1-1 資料定義語言——DDL	317
15-1-2 資料處理語言——DML	317
15-1-3 資料庫管理者——DBA	318
15-1-4 資料庫管理系統的結構	318
15-2 資料庫系統的要求與目的	319
15-3 資料模式 (Data Model)	322
15-4 CODASYL 資料庫模式	326
15-5 CODASYL 資料描述語言 (DDL)	327
15-6 CODASYL COBOL 資料庫設備	329
15-6-1 資料庫錄及其相關術語	330
15-6-2 錄選定	330
15-6-3 DML 指令	331
15-7 結論	333

附錄

附錄 A	335
附錄 B	337

第一章 概論

在開始討論本書的內容之前必須對本書涵蓋的最基本元素——資料 (data)，做一個明確的定義與說明。何謂資料？其與運算的關係為何？雖然此為相當基本的問題，但要回答此問題時既不簡單，也無固定的答案。對於本書所討論的方向而言，我們可定義：

資料 (data)：為存在程式外面，被程式處理的一種情報 (Information)。由於主記憶體內方便於執行運算，而大部份時間都存於外界，副記憶體 (secondary memory) 之內，例如磁盤，磁帶之中。

起先，早期的程式設計，存於記憶體的資料只專屬於其特定的程式，不能被別的程式所共用。而目前的演變已慢慢傾向於資料與處理此資料的程式相互獨立，此演變可歸納為幾種因素：

- (一) 程式系統日趨複雜，相同的資料往往反覆出現於不同的檔 (file) 與不同程式之內。
- (二) 由於資料的大量需求，花費在收集、存檔與使用所需求資料的費用，已成為整個系統的主要開支。
- (三) 維護資料時無法避免的資料結構上之改變，希望能儘量不影響到其相關的程式系統，以減少軟體維護的費用。

2 COBOL 檔案技術與資料庫

以上觀念的進步，對資料存放技術之發展有顯著助益。由於資料維護成本不斷提高，吾人慢慢注意到資料的保護，使其免於結構上之經常更換。如此就必須發展資料存放一般化。基於此了解，已迫使吾人對資料有效、成本低的運算做重新考慮，進而面對到資料管理的問題。

由於技術不斷改進，以前認為有效的解決方法，目前已被淘汰。電腦程式，就如汽車一樣，會隨使用時間的流逝而退化，到最後，維護舊車的費用比買一輛新車還貴。因此在解決電腦所面臨的問題時，往往將舊程式丟棄，重新設計，反而可得較好的效果。不幸地，在資料維護上往往面臨無法負擔舊檔案系統的維護費，也無法承受更新系統的費用，造成進退兩難無所適從的狀況。為解決此困境，因而引進了資料庫的觀念。定義名詞如下：

資料庫 (Data Base)：資料檔的集合，檔與檔之間有關係存在，可同時為多個應用程式所使用。

資料庫管理系統 (data Base Management System, DBMS)：包括資料庫本身以及為維持資料檔之間相互關係所須的軟體程式。

這些定義稍嫌簡單，但相當清晰，足夠目前需要。

基於此傾向，本書的重心即在於檔案處理的原則與技術、如何以 COBOL 來解決檔案問題以及檔與資料庫二者相互的關係。

在設計檔案時，一個問題常被提出：

如何處理資料與程式之間的關係？

吾人可以用三點來回答：

- (一) 對於不同之程式，資料的使用，越簡單越好。
- (二) 資料跟程式相互獨立。
- (三) 所用的資料易於查尋。

並非我們所述的技術都能滿足此三種要求，但也儘量符合這些原則了。

1.1 資料

電腦時代的來臨，提供了處理大量資料的能力。收集大量資料是一回事，但是否能夠有意義、適當的處理資料又是另外一回事。在商業社會中，因為資料處理系統工作的不當而倒閉破產的大有人在。因此在收集大量資料之後，還必須付出可觀的心血於資料組織結構方面，才能夠有效的處理這些資料。電腦運算的目的，並非在於資料的收集，而是為了幫助吾人做決策。如果一個決策並非偶然靈感的觸發與不合邏輯之決定，則其必為基於各來源的資料情報而做的最好選擇。在有些狀況下，這些策略可經由電腦適當的演算法（Algorithm）分析出最好的決定。可舉一例說明：當碰到客戶的帳單過期 60 日還未繳費，即可由電腦自動決定，送出信件來追繳費用，或向客戶下警告，威脅要尋求法律途徑解決。此即為電腦幫人類所作最簡單決策。

成本與收益的取捨

在設計資料處理系統時，常事先估計可能得到的收益與必須花費的成本將二者做一比較取捨，以決定最適當的系統。一般而言，理論上最優良的系統，往往並非合乎使用、最適當的系統。因理論上最優良的系統為滿足其理想，常不可避免地提高了系統成本，雖有收益的提高也不一定能補償成本的差額。故必須借助成本與收益的取捨分析，方能決定較適合使用的系統。

資訊財產

在審核一公司的資產時，會計師只考慮每樣物品其可換算的金錢價值；然而以經理的眼光來看，則着重於其手下職員編制為其資產的標準。當然，對一個證券市場分析家而言，此兩個因素都為其評定公司資產的標準。但有很多公司其所擁有最珍貴資產並非表現在收支圖表與組織編制之上，而是其在本行業務上所擁有的知識與資料。

4 COBOL 檔案技術與資料庫

經營生意，須具備有資本、人力與資料情報三要素，三者缺一不可，而其中任何一項可以產生其他兩項。例如，資本與情報都可吸收人力，因人類工作就為了賺取高薪，且每人都有求知慾。的確，一個優良的技術人員往往為了追求其專門技術的最新知識，而寧願接受低薪的職位。在過去，資料情報被視為各部門的副產物。如，財政資料屬於會計部門，市場情報隸屬於推銷部門；工程資料則屬於設計生產部門……。那時人力彙集系統的技術完全限制了資料收集處理工作的發展。因為以管理的觀點來看，金錢被視為生產的主要原料，資料運算作業於會計部門內處理，資料處理報告乃基於此作業來提出，給經理由此而做管理上的決策。此種方式忽略了資料需求的重要特性。目前基於此種缺失的發現，很多公司已將公司組織更改，使資料運算報告的工作與其他部門相獨立，以避開因各部門內部作業的影響，造成資料情報的偏差。

在一個成功的企業組織中，資料情報與個體相互獨立是成功經營很重要的因素。此現象可用生物學上有機組織的特性以表現出來：細胞是構成組織的個體，而細胞與細胞之間的生長死亡都相互獨立，因此雖然細胞的生命期短、死亡率高，但還是不影響有機組織的生長生存。由此特性可知，以為自己在組織中是不可缺少的一員，而威脅辭職不幹，實為不智的策略，因為組織可以另外雇用取代者，加以修護因此雇員的離去所造成的傷害與損失，而繼續生存下去。但話說回來，此種修護常花費不少，因人員的離去即構成組織資料的遺失與組織能力的減弱。故一成功的企業組織必妥善的管理資料，與個體相互獨立，以防止因人員的離去而造成傷害。

資訊系統 (Information System)

評定一個資料的價值困難之處，在於決定何資料值得收集，與必須貯存於何處。如不能適切地決定資料的價值，會因而導致程式系統的失敗。由於吾人不可能收集到所有資料，也無此必要，因此如何選用吾人所需資料來做處理，實為重要課題。由於資料數量的激增，使得處理上資料轉換變成難以理解，因此為求易於處理，必須將資料使用做有意義的詮釋。

• 資訊 (Information)：為資料 (data) 在處理上，針對人類做有意義的詮釋。

• 資訊系統 (Information System)：則為可提供收集貯存原始資料，進而加以處理，而產生可直接為人類使用的輸出之系統。

資訊系統可分為四大部分

- (一) 輸入
- (二) 系統處理器
- (三) 外界記憶體
- (四) 輸出

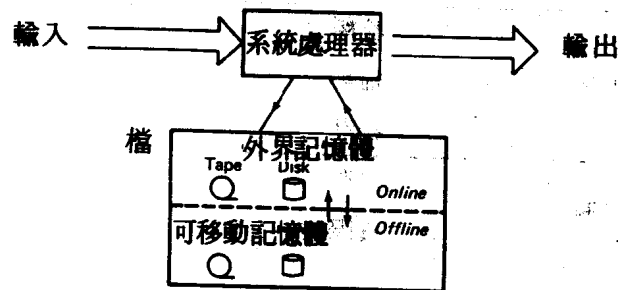


圖 1-1 資訊系統

以下列舉了各式各樣資訊系統的應用

- 會計
- 飛機航線售票系統
- 帳單
- 顧客資料錄
- 所得稅錄
- 保險錄
- 庫存控制
- 地籍資料錄
- 病人醫療錄
- 福利錄

由以上各類應用實例並不難想像其工作狀況。

表 1-1 表示處理資訊需求的三個概括階段即(←)作業上，(□)管理上 (⇒)行政上，三者的活動情形。此三階段當然可以於一個系統中同時出現。

表 1-1 資訊系統的活動階段

階 段	活動情形
行政決策	智力的
	不可預言的
	瀏覽
管理控制	詮釋
	歸納
	統計
作業控制	自動
	實際的

系統性能 (System Performance)

為決定最適當的系統，則必須先定義測定系統性能的標準，例如在飛機航線售票系統中，做存檔、查尋、更改預訂座位情形時有鑑定合於標準之反應時間。一般都以秒為單位來計算。便宜但反應慢的系統是無法被接受的。

性能標準必須於系統定義完成前被訂定，此標準即構成系統定義的一部份。例如，資料保證無錯，可立即反應，最新機種，系統費用可滿足用戶的預算等等，都可做為性能之標準。當性能標準加以數量化之後，吾人即可做系統性能測試。如吾人可定錯誤發生率必須低於千分之一，而加以測試；或可定反應時間的標準為：系統所有作業要求，其 80 % 之要求於 30 秒之內答覆反應，95 % 之要求於 60 秒內答覆反應，且反應時間不得超過 2 分鐘。實際說來，系統性能的測定主要以用戶對系統工作的滿意程度為標準。在資料處理上，性能的測定，則以下列為準：

(←) 檔平均搜尋時間

- (二) 做雜亂出入 (random access) 錄 (record) 之平均反應時間
- (三) 每天異動處理 (transaction) 的最小工作量。

資料處理的目的有兩方面

- 1) 組織成大量資料，使機器有效地使用與操作。
- 2) 組織詮釋資料，使人類能有效地使用。

在早期的資訊系統中，此二目的常糾纏一起難以區分，目前已可做很大程度的區分，此造成極大的好處。為解決第一目的，機器貯存資料則採用資料庫的方式；為解決第二目的，則以目前多種應用程式式提供了多種演算法來處理。

1-2 檔(File)

在電腦未發明幾世紀前，古人就知道對資料有組織的貯存，以備使用時的方便。例如古代所發明的卷軸古書，即為現代電腦所用循序檔 (sequential file) 的前身。由卷軸古書再進化改進的書本，為一頁一頁裝訂起來，有章節、頁數，可用書前或書後的目錄來查所需資料的頁數、章節，而找到所需資料，這種書本即可視為電腦所用指標檔 (indexed file) 的例子。到近代，更有各種科學方法來整理貯存資料，其中以電腦的檔案系統為最進步的方法。

利用電腦檔案系統來做資訊的來源有數個理由：

- 減低作業費用
- 增進資料使用效率
- 有效控制報告的要求
- 增加處理工作量
- 增加變通性

欲確定檔案系統的價值，因用戶資料來源與使用的理由之不同而改變，但我們還是可以提出檔案系統的基本價值：即其對企業的組織結構有很大的助益。此種助益是屬於資料處理技術上的，並沒辦法用金錢來加以衡量。

子群 (Subgroupings)

本書所論的資料 (data) 是指事實 (facts) 的組合。此事實乃由觀察與測量所得的數據。爲了貯存與作業上的方便，必須明確地定義資料所屬的子群。在檔案作業運算時，子群與其相關的術語，廣泛地被使用，術語與其定義歸納於表 1-2

表 1-2 檔案術語

名稱	定義
資料 (data)	以可出入形式存在的事實之集合
資料庫 (data base)	相關檔的組合
檔 (file)	相關錄的組合
錄 (record)	相關資料之組合，爲檔的基本單位
項 (item)	在檔中可指定出入 (access) 的最小單位
聚 (Aggregate)	在錄中項的組合
群 (group)	在錄中連續項的組合
數元 (bit)	資料貯存在電腦的最小單位 (爲 2 進位的一個位數)
數元組 (byte)	數元的組合，可用位址 (address) 指示的最小單位 (通常含有 8 位數元)

在資料處理的領域內，到目前爲止，術語還沒有完全統一。因爲在此領域中，有很多範圍還在研究開發之中，爲了討論上的方便，專家學者們常創造自己專用的術語來做分析討論。經過日積月累就發生了相同的概念與涵義，却有不同的術語可以表示；同一個術語，在不同的書本上、系統上，常表示了不同的概念與意義。這種混亂的現象最易造成初學者的混淆不知適從。本書所討論的檔案術語，在此處一經定義，全書就全部延用此種定義，以避免諸君的混淆。

在表 1-2 中所定義的資料子群中，吾人做處理工作時，主要牽涉到的基本單位