

第一部分

数据库和 FoxPro 入门

这一部分扼要地介绍了数据库的基础知识和 FoxPro 的基本情况，由第 1 章和第 2 章组成，主要对象是对数据库或 FoxPro 不熟悉的读者。

第 1 章首先说明了关系型数据库的基本概念、语法规则、变量的使用要求，并对关系型数据库中常用的名词术语作了概要的解释。在此基础上，简述了关系型数据库在管理工作中的应用情况，并对其发展趋势提出了笔者的一管之见。

第 2 章介绍了 FoxPro 的优点、技术指标和安装过程，比较了 DOS 和 Windows 平台上的开发特点，并对手工编程与自动编程的情况作了介绍。

通过第一部分的内容，读者可以对关系型数据库及 FoxPro 有一个比较全面的了解，为运用 FoxPro 开发应用程序打下一个比较好的基础。

第1章 数据库概述

- 数据库基本概念
- xBASE 数据库主要术语及语法
- 计算机管理信息系统发展综述

人类已进入了信息时代。信息时代的最显著标志就是信息量的急剧膨胀。如何用计算机迅速、准确地存取大量的数据,促使了计算机科学领域的一个重要分枝——数据库系统的形成和发展。数据库系统简称 DBS (DataBase System)。

本章采用提纲式的叙述方式向读者介绍数据库系统的基础知识。已经熟悉数据库系统的读者可跳过本章。

1.1 数据和数据库

数据是信息的具体反映(信息流),它是一种重要的资源。如一个国家的人口、工农业产值等等。数据的类型有各种各样,数据的大小、长度可以各不相同。

数据库(DB)是以一定的方式存储在某种储存介质上(如磁盘、磁带、光盘等)的相关数据的集合。其特点是数据高度共享,冗余度小。

本世纪50年代末,出现了数据库管理系统的雏型——文件管理系统。60年代末在美国诞生了第一个商品化的数据库系统。从80年代初开始,美国 Ashton-Tate 公司先后推出了 dBASE II, dBASE III, dBASE III PLUS, 编译 dBASE 等商品化数据库系统。1987年2月美国 Fox Software 公司推出了与 dBASE 完全兼容的 FoxBASE+1.00 版,此后该公司又先后推出了 FoxBASE+2.00 版, FoxBASE+2.10 版。

dBASE 系列和 FoxBASE+ 系列都是关系型数据库系统,统称 xBASE 数据库。xBASE 数据库在国内有多种汉化版本。1994年3月, Borland 公司推出了 dBASE IV 2.0 中文版,同年11月,微软公司推出了 FoxPro 2.5b for DOS 中文版和 FoxPro 2.5b for Windows 中文版。随着功能更强、使用更方便的数据库系统的推出, xBASE 数据库系统将会让位于来者。

根据数据的组织结构不同,数据库通常分为三种类型,即层次模型、网状模型和关系模型。其中关系模型的存储结构与人们平常使用的二维表格相同,容易为人们理解,并成为目前在微机上流行的数据库系统。前面提到的 xBASE 数据库系统及本书将要重点讨论的 FoxPro 2.5b 中文版都是关系型数据库系统。

在后面的各章节中,如无特别说明,所说的数据库均指在微机上流行的关系型数据库。

下面是一个关系型数据库的实例。

表 1-1 职工奖金

姓 名	全 勤	奖 金	发放日期	像 片	备 注
张三	否	200.26	94/01/01	gen	事假一天
李四	是	300.43	94/01/31	Gen	
王五	是	300.28	94/02/01	gen	
赵六	否	150.21	94/03/01	Gen	病假三天

1.2 数据库文件

表 1-1 是我们打印得到的一张二维表。表里的数据实际上存放在一个数据库文件中。这些数据在数据库内组织如下：

```
USE 职工奖金.DBF
```

```
LIST
```

Record #	姓名	全勤	奖金	发放日期	像片	备注
1	张三	.F.	200.26	01/31/94	gen	Memo
2	李四	.T.	300.43	01/31/94	Gen	Memo
3	王五	.T.	300.28	02/01/94	gen	Memo
4	赵六	.F.	150.21	02/01/94	Gen	Memo

一个完整的数据库有时是一个单独的文件，其扩展名为 .DBF；有时是由两个文件组成，另一个文件的扩展名为 .DBT（在 FoxPro 中为 .FPT）；此外，在 FoxPro 中，一个完整的数据库有时甚至是由三个文件组成，即数据库文件、记忆型和通用型数据文件以及结构化复合索引文件，其扩展名分别为 .DBF、.FPT、.CDX。

.DBF 文件存放定长类型的数据，如“字符型”（C 型）、“逻辑型”（L 型）、“数值型”（N 型）、“日期型”（D 型）；同时还存有不定长数据类型的指针，如“记忆型”（M 型）和“通用型”（G 型）。

.DBT（或 .FPT）文件是为了减少数据的冗余度而设计的。它用于存放不等长的数据。如大块的说明性文字或诸如图形、声音、录象之类非文本数据（只适用于 FoxPro for Windows）。

.CDX 文件是 FoxPro 中存放索引的文件，称为“结构化复合索引文件”。它与通常的 .IDX（.NDX）索引文件不同，是在创建或修改结构时由用户定义的（用户也可以在程序中用命令创建 .CDX 文件）。

在打开数据库文件（.DBF）的同时，.DBT（.FPT）文件和 .CDX 文件也随之打开。

上例的数据库就是由三个文件组成。

```
dir 职工奖金.*
```

```
Volume in drive C has no label
```

```
Volume Serial Number is 3451-1404
Directory of C: \

职工奖金  DBF          7832   07-19-94  10: 02a
职工奖金  FPT          178920  07-19-94  10: 57a
职工奖金  CDX          24567   07-19-94  10: 58a
          3 file(s)    111329 bytes
          18061312 bytes free
```

1.3 数据库结构

数据库的数据结构，是记录数据库所包含的各字段的名称、长度、类型、小数点位数等信息的“信息表”。上例的数据库结构如下：

```
display structure
Structure for database C: 职工奖金.DBF
Number of data records:      4
Date of last update : 07/18/94

Field      Field Name      Type      Width  Dec
  1         姓名          Character    6
  2         全勤          Logical      1
  3         奖金          Numeric     8      2
  4         发放日期      Date        8
  5         像片          General    10
  6         备注          Memo       10
* * Total * *                44
```

数据库结构存放在数据库文件中。数据库文件（.DBF、.DBT 或 .FPT、.CDX）都由两部分组成。第一部分是对数据库结构的说明，通常称为“文件头”。这部分的大小除 .DBT（.FPT）文件外，取决于数据库结构的复杂程度。一旦数据库结构确定后，“文件头”的长度就确定了。第二部分是实际存储各个具体数据的部分，这部分的大小由数据库所存放的数据的多少来确定。随着数据的添加或删除（物理删除），这部分的大小是变化的。

如果把“文件头”比作“派出所”的话，那么“文件头”后面的数据就好象“派出所”辖区内的“居民”。该“居民区”内的有关信息都可以在“派出所”里找到。因此，如果要用其他语言存取数据库数据的话，必须清楚数据库文件的“文件头”的结构。

1.4 xBASE 数据库主要术语

1. 记录

记录是指数据库中存放各行数据的单元组。在表 1-1 所对应的数据库“职工奖金.DBF”中，每一行就是一个记录。

2. 字段

字段是指数据库的各记录中，分门别类地存放列信息的单元组。在表 1-1 所对应的数据

库“职工奖金.DBF”中，每一列就是一个字段。

3. 字段名

字段名即字段的名称。在表 1-1 所对应的数据库“职工奖金.DBF”中，“姓名”、“全勤”、“奖金”、“发放日期”、“像片”、“备注”都是字段名。

字段名应以 26 个英文字母或汉字开头，由字母、汉字或下划线“_”组成，最长 10 个字节。字段名不允许重复定义，即字段不能重名。

4. 字段类型

字段类型即字段属性。字段类型决定了字段所能存放的数据。

一般数据库的字段类型有五种类型：字符型、数字型、日期型、逻辑型和记忆（备注）型。FoxPro 增加了两种数据类型：通用型（General）和浮点型（Float）。

（1）字符型字段

字符型字段的数据值是由可显示的汉字和字符（包括空格）构成的，用字母 C 表示。系统允许一个字符型字段的最大宽度为 254 个字符。上例中的“姓名”即是一个字符型字段。

（2）数字型字段

数字型字段的数据值是由正负号、小数点和 0 至 9 的数字构成的，用字母 N 表示。上例中的“奖金”是数字型字段。

数字型字段的最大宽度为 20。该类型字段的宽度是最大整数位、最大小数位之和，小数点和正负号分别算作一个数位。小数的宽度在 0—19 之间，并且小于等于字段宽度减 2。

（3）日期型字段

日期型字段是用于存放日期的数据，用字母 D 表示。格式为：年.月.日或月/日/年等。其中“年份”可以是两位，也可以是四位；“月份”、“日期”均为两位数字。

上例中的“发放日期”是日期型字段。

日期型字段的固定宽度为 8 个字节。

（4）逻辑型字段

逻辑型字段的数据值是由逻辑真（.T.）和假（.F.）构成的，用字母 L 表示。接受 T、t、Y、y 为逻辑真（.T.）值；接受 F、f、N、n 为逻辑假（.F.）值。

上例中的“是否全勤”是逻辑型字段。

逻辑型字段的固定宽度为 1 个字节。

（5）记忆型字段（也称为备注型字段）

记忆型字段用于存放大块的数据，数据存放在数据库文件的辅助文件（.DBT）中。它在数据库文件中用“memo”一词指明，用字母 M 表示。

上例中的“备注”是记忆型字段。

记忆型字段最大可达 4096 个字节。FoxPro 2.5 及其以后版本则不限制记忆型字段的长度，记忆型字段内可以存放所有的 ASCII 码，包括不可打印的控制符。

（6）浮点型字段

浮点型字段用于存放数值运算精度要求较高的数据，是专为科学数据而设置的。用字母 F 表示。其最大宽度的限制及宽度的计算方法与数字型字段相同。

（7）通用型字段

通用型字段用于存放非文本的其他类型数据，如图形、声音、录象等，用“gen”一词

指明，用字母 G 表示。其长度不受限制。

上例中的“像片”是一个通用型字段，它用于存储图象数据即职工的像片：

实际上，加配“多媒体卡”后，通用型字段还可以存储诸如声音、录象等数据。

在“职工奖金.DBF”数据库中，各字段的“字段名”、“字段类型”、“字段长度”和“小数位数”如下表所示。

表 1-2 “职工奖金.DBF”数据库结构

字段名	字段类型	字段长度	小数位数
姓名	Character (字符型)	6	
全勤	Logical (逻辑型)	1	
奖金	Numeric (数值型)	8	2
发放日期	Date (日期型)	8	
像片	General (通用型)	10	
备注	Memo (记忆型)	10	

5. 字段类型、长度及小数位的关系

表 1-3 字段类型、长度及小数位的关系

字段类型	长 度	小数位
字符 (C)	1—254	无
数字 (N)	1—19	有
日期 (D)	8	无
逻辑 (L)	1	无
记忆 (M)	≤4096 (FoxPro 无限)	无
浮点 (F)	1—19	有
通用 (G)	无限	无

6. 字段总数

构成一个记录的字段的总个数。FoxPro 2.5b 的字段总数可达 256 个。

7. 记录长度

一个记录所包含的全部字段的总字节数。它等于全部字段宽度的总和加 1。

FoxPro 2.5b 的最大记录长度为 65535 字节。

8. 记录号

记录在数据库中存放的序号（物理序号）。

9. 当前记录号

打开数据库后，数据库指针所指的记录序号。

10. 记录总数

数据库中的记录总个数。

FoxPro 2.5b 允许的最大记录数为 10 亿。

11. 查询

在打开的数据库中查找满足一定条件的记录。

12. 索引

按关键字值建立记录间的逻辑顺序。索引值存于索引文件中。索引文件的扩展名为“.IDX”。结构化复合索引文件的扩展名为“.CDX”。

13. 排序

对当前打开的数据库文件按指定的条件重新排列顺序，排序的数据存入新的数据库文件中。排序后产生新的数据库文件。

14. 统计

对已打开的数据库文件中的某些信息进行统计分析、汇总等，以便打印出各种统计报表。

1.5 xBASE 数据库命令语法

xBASE 数据库语言的命令既可以以交互的方式由用户在命令行或命令窗口逐条输入并执行；也可以把命令编辑在若干个文件（.PRG 等）中重复使用，这个过程也就是我们通常所说的“编程”。

事实上，后一种用法才真正体现 xBASE 数据库语言的优越性，也是开发各种管理系统的基础。另外，支持“真编译”的 xBASE 数据库语言还可将调试运行好的 .PRG（或其他执行文件）编译成脱离数据库系统能在计算机操作系统下直接运行的可执行文件。

xBASE 数据库语言是一种高级语言，主要由命令、表达式和函数组成。

1.5.1 xBASE 数据库命令语句

1. 命令语法规则

(1) 句式

xBASE 数据库的命令语句类似英语的祈使句。每个命令均以“谓语动词”开头，命令的操作对象是“宾语”。多数命令语句还加上一些功能子句作为“补语”以扩大同一命令的使用范围。如：

RE PLACE	NE XT 100	奖金	WIT H 300.50	FOR 全勤
↑	↑	↑	↑	↑
命令动词	范围	操作对象	功能子句	逻辑条件

(2) 顺序

在命令语句中，命令动词是必须的且一定要放在句子开头，操作对象和功能子句的顺序可先可后。如：

REPLACE 奖金 WITH 300.50 FOR 全勤 NEXT 100
 等价于：
 REPLACE NEXT 100 奖金 WITH 300.50 FOR 全勤

(3) 省略

省略包括两方面，一是命令和关键字的缩略；一是操作对象和功能子句的省略。

a) 命令和关键字的缩略

为了简化键盘输入，在不致产生混淆的前提下，命令、关键字和函数名可以缩写为前4个字母。如 REPLACE 简化为 REPL，FIELDS 简化为 FIEL，SYSMATRIC () 缩写为 SYSM ()。

但是，某些命令和关键字缩写为前4个字母会产生歧义，不能缩写为前4个字母。如 GETFONT() 与 GETFILE() 前4个字母相同，就不能缩写为前4个字母。

b) 操作对象和功能子句的省略

有时操作对象和功能子句是可以省略的。如：

REPLACE	奖金	WITH 300.50	(省略范围和逻辑条件)
DELETE			(只有命令动词，省略其他子句)

2. 常用的功能子句

(1) 范围子句

范围子句用于确定命令的作用范围，共有4种形式。

ALL 表示命令的作用范围是数据库文件的所有记录

RECORD N 表示数据库文件中第N个记录 (N 可以是常数，也可以是数值型表达式)

NEXT N 表示从当前记录起 (包括当前记录) 的N个记录。

REST 表示从当前记录起 (包括当前记录) 到库尾的所有记录。

对于需要给出范围子句的命令，如果不给出范围子句，系统将自动取缺省值。不同的命令其范围缺省值是不同的。

例如，LOCATE FOR <expr1> 命令的范围缺省值是：ALL；

而 REPLACE 命令的范围缺省值是：NEXT 1。

(2) 字段名列表子句 FIELDS

字段名列表子句用于规定命令可以处理的字段。如果选用了字段名子句则只有列于字段名列表的字段才被处理。

字段名子句也可省略，系统将自动取缺省值。字段名列表子句的缺省值是所有字段。

例如，COPY TO <filename> 命令的字段名列表子句的缺省值是所有字段。

(3) 逻辑条件子句 FOR 和 WHILE

逻辑条件子句的作用类似于范围子句，亦即把命令的作用范围限制在满足逻辑条件的记录上。其中 FOR 子句用于筛选出符合子句中 <条件> 的所有记录；而 WHILE 子句一旦遇到不满足子句中 <条件> 的记录时就终止命令的继续执行。

FOR 子句和 WHILE 子句同时使用时，WHILE 子句的优先级比 FOR 子句的优先级高。

(4) 字体、字号、字型子句 FONT 和 STYLE

在 FoxPro 2.5b 中文版 for Windows 中，经常用到 FONT 子句和 STYLE 子句。该子句用于确定字体、字号和字型。如果省略，则系统取当前活动窗口的字体、字号和字型作为缺省值。

1.5.2 xBASE 数据库变量

xBASE 数据库系统中将数据区分为常量和变量。变量应以 26 个英文字母、汉字或下划线“-”开头，由数字、字母、汉字或下划线“-”组成，最长 10 个字符。

xBASE 数据库使用两种变量：字段变量和内存变量。

字段变量是已打开的数据库的字段名。一个数据库中的每一个字段都是一个字段变量。

内存变量与字段变量不同，它存于内存之中，独立于数据库文件。内存变量是一种临时性的工作单元，需要时可随时定义；不用时又可释放。在给内存变量命名时，最好根据内存变量的用途把同一类的内存变量赋与某种特征，这样当不用这些内存变量时可以用一条命令把它们释放掉。如：

```
RELEASE ALL LIKE TEMP *
```

内存变量与字段变量一样也有类型之分，共有五种。其中四种（字符型、数字型、逻辑型、日期型）与字段变量相同，另一种称为屏幕（S）型，用于存储当前屏幕的信息。

当内存变量与字段变量重名时，字段变量优先。在这种情况下，如果要引用同名的内存变量，可在内存变量名前加上“M→”（在 FoxPro 中加“M→”或加“M.”都可以），以指明引用的是内存变量。

在编写程序时，有时为了使变量的含意清楚，提高程序的可阅读性，可以使用超过 10 个字符的字符串来给变量命名。当内存变量名长度超过 10 个字符时，系统只取前 10 个字符来识别。此时要注意前 10 个字符的唯一性，以避免重名。

1.5.3 xBASE 数据库函数

函数用来完成某种运算，返回系统状态信息或作出某种判断。函数被调用后，被调用函数向调用者返回函数值。函数的返回值分为四种，即字符型、数字型、日期型、逻辑型。

根据函数的功能，可将函数分为 8 类。

- | | |
|------------|-----------------------------|
| 1. 算术运算函数 | 如，ABS(X) 返回 X 的绝对值； |
| 2. 字符串处理函数 | 如，TRIM(X) 返回去掉尾部空格后的字符串； |
| 3. 日期函数 | 如，DATE() 返回系统当前日期； |
| 4. 类型转换函数 | 如，CTOD(X) 返回字符串 X 对应的日期； |
| 5. 测试函数 | 如，FILE(X) 返回文件名为 X 的文件是否存在； |
| 6. 标识函数 | 如，ALIAS(X) 返回 X 工作区的别名； |
| 7. 输入函数 | 如，INKEY() 返回用户最近所按的键值； |
| 8. 用户自定义函数 | 由用户自己编写的带返回值的子程序。 |

1.6 计算机管理信息系统（MIS）发展综述

1.6.1 MIS 系统的作用

随着计算机在办公领域的日益普及，人们对计算机的应用已不仅仅局限于文字处理，而更多地用计算机来进行管理工作。用计算机进行管理工作就要用到“数据库”。

尽管“数据库”这个计算机专业术语对于广大办公室管理人员并不陌生，然而，如果

要让管理人员运用“数据库”语言来编写管理程序,就可能是一件不容易的事了。

数据库系统解决了快速、高效存取和处理大量数据的问题,但在实际应用中仍存在着使用者与开发者的矛盾。从事管理工作的管理人员可能不熟悉数据库语言,而精通数据库语言的软件专业人员又不一定直接参与管理工作。此外,直接运用数据库语言对数据库进行操作、管理,其效率是很低的。

为了解决上述矛盾,一般情况下,管理人员并不直接运用数据库系统进行管理工作,而是由编程人员根据本单位管理工作的实际要求进行设计,开发适合本单位需要的数据库应用系统,并提供给管理人员使用。这样的系统我们称为计算机管理信息系统,简称 MIS 系统 (Management Information System)。

1.6.2 MIS 系统的现状

MIS 系统解决了使用者与开发者脱节的矛盾且具有功能齐全、操作简便、易于使用等优点,但同时也存在着许多不足。

一方面,对于要求高的用户来说,MIS 系统只能处理事务性的管理工作,往往功能比较单一,不能完成更深层次的管理、决策工作,不具备人工智能的特色。

例如,一个工厂的财务 MIS 系统只能管理工厂财务方面的事务;产品 MIS 系统只能处理有关产品方面的问题。对于工厂资金的使用,产品结构的调整等有关工厂发展方向的重大决策问题,MIS 系统是不能解决的。

为了使 MIS 系统能处理更深层、更繁多、更复杂的事务,MIS 系统开始向集成化、智能化的方向发展。带有集成化、智能化的专家系统已在某些领域得到了应用,其效果是令人鼓舞的。作为国家“863”高技术自动化领域的计算机集成制造系统 (CIMS) 就是 MIS 系统向集成化、智能化发展的实例。

另一方面,从更加广泛的应用来看,如办公自动化领域,用户对 MIS 系统的要求又是另一个极端。对于这些用户来说,传统 MIS 系统存在着如下缺点。

首先,MIS 系统开发费用高。除非本单位有开发人员,自己开发可降低开发费用以外,一般情况下,请外来编程人员开发需要支付较高的开发费用。

其次,MIS 系统开发周期长。开发工作需要经过系统调研、系统设计、模块划分、程序设计、上机编程、系统调试、用户培训、交付使用等一系列过程。

第三,一般 MIS 系统通用性差。MIS 系统属于“量体裁衣”的定做产品,为甲单位开发的 MIS 系统,一般乙单位不能直接使用。

第四,系统维护困难。通常 MIS 系统的开发者并不是使用者,使用中出现的问題,一般只能由开发者来解决。随着实际情况的变化,MIS 系统需作一些修改或调整,这些工作也只能由开发者来完成。

这些缺点阻碍了 MIS 系统在普通办公自动化领域的广泛使用。

为了解决这方面的矛盾,近年来 MIS 系统的发展又出现了两种趋势。

一是给编程人员使用的“数据库自动编程系统”。

一是供最终用户使用的“傻瓜型数据库系统”。

“数据库自动编程系统”给编程人员带来了很大方便。其主要优点是:

(1) 减少了工作量

以往用手工编程,需要逐字逐句输入程序并调试,造成重复劳动,工作效率低,自动编程可大大提高编程效率并减少调试时间;

(2) 减少了程序的错误率

手工编程容易出错,而自动编程则可以避免诸如命令拼写错误等人为因素造成的错误;

(3) 提高了程序的可维护性

手工编写的程序其风格与编程者的思维方式、爱好有很大关系,往往因人而异,而自动编程生成的程序比较规范,可读性好,易于维护;

(4) 降低了对开发人员的要求

手工编程要求开发者对数据库语言有较深入的理解并具有较高的技术水平和开发经验,而使用自动编程系统则可降低对开发人员的要求。利用自动编程系统,即使是新手也可生成出较高质量的程序。实际上,生成程序的质量与自动编程系统本身直接相关。

“数据库自动编程系统”也有不足。其主要缺点是:

(1) 程序的冗余度较大

自动编程系统生成的程序往往有许多不需要或重复的语句,造成不必要的冗余,而手工编程则可减少不必要的语句,避免重复;

(2) 自动编程系统一般只能解决程序级或模块级的问题,对于系统的需求、各模块的相互联系、相互调用还必须由开发人员解决;

(3) 自动编程系统的使用对象仍然是编程人员而不是最终用户。在这方面,“傻瓜型数据库”就显示出自己的优势来了。

“傻瓜型数据库”把数据库的命令“菜单”化,用户通过“点菜”的方式就可完成对数据库的操作。

“傻瓜型数据库”类似于“傻瓜”像机。拍照者可以不知道什么是“光圈”、什么是“快门”,照样可以用“傻瓜”像机拍出满意的照片;同样,用户可以不知道什么是“字段”、什么是“记录”,也可以用“傻瓜型数据库”管理自己的数据。

“傻瓜型数据库”的优点在于:

(1) 功能相对简单、操作方便

(2) 不需编程、适合管理人员直接使用

(3) 不需要维护

(4) 适用面较广

其缺点是:

(1) 功能不能太复杂

(2) 不能满足用户的特殊需要

(3) 用户重复操作多

FoxPro 2.5b 中文版本身带有若干个程序生成器,具有自动编程的功能;同时 FoxPro 的许多命令也可以通过“点菜单”的方式执行。因此我们可以说 FoxPro 2.5b 中文版不仅是一种很好的数据库编程语言,同时也兼有自动编程的功能并可以作为“傻瓜型数据库”来使用。有关这方面的内容,我们将在以后各章中详细讲到。

1.6.3 MIS 系统的未来

MIS 系统及其各分枝的相互联系和优缺点可用图 1-1 表示:

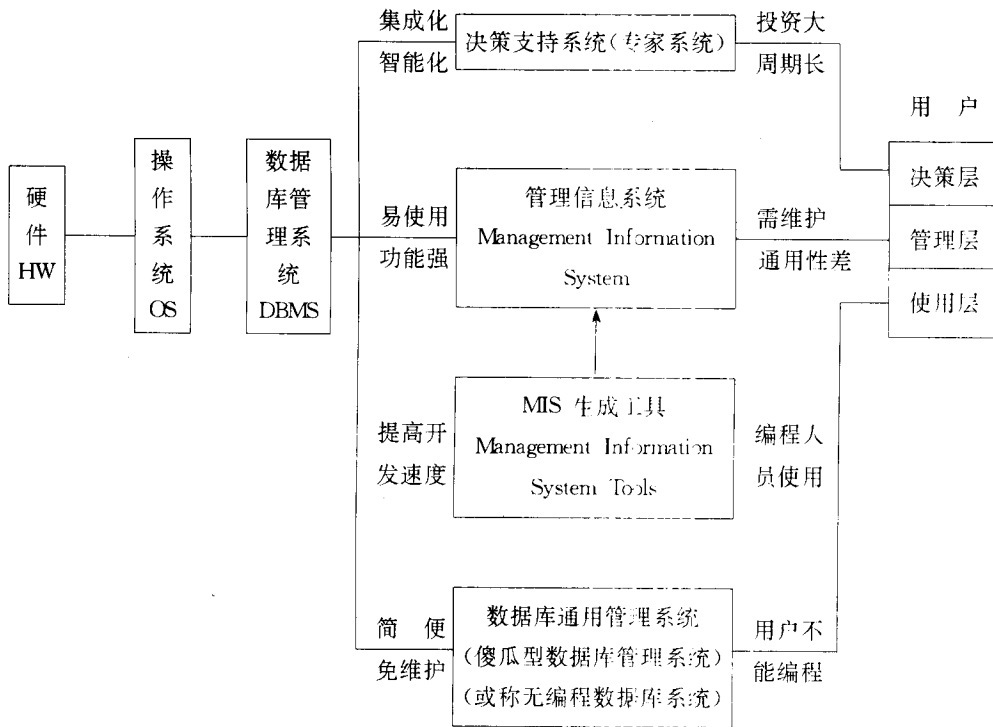


图1-1 MIS系统及其分枝

MIS 系统及其分枝系统各有优缺点，各有自己的适用范围，各有自己的使用群体。在今后的发展过程中将会互相补充、长期共存。同时各个分枝之间将会互相渗透、互相溶合，产生新的更加方便实用的 MIS 系统。

例如：自动编程系统加入人工智能的因素，淡化“编程”意识，使用户在使用过程中不知不觉地自动完成编程工作。

又如：“傻瓜型数据库”向用户开放接口，使有编程能力的用户可根据自己的需要，编写自己的模块嵌入到“傻瓜型数据库”中来完成用户的特殊操作。

可以预见，MIS 系统的发展将会向高层次研究领域和大众化普及两个方向展开。

在高层次研究领域方面，MIS 系统将向“智能化”、“多媒体”、“多目标”、“集成化”领域迈进；从单机系统向联机系统、网络系统发展。

在大众化普及方面，随着计算机硬件价格的下降和全民计算机知识的普遍提高，计算机进入普通办公室，进入寻常百姓家的进程将会加快。使用计算机管理日常事务、管理家庭事务的用户将会大大增加。适用于这类用户的“傻瓜型”、“微型”数据库产品也将会有较大的发展。

1.7 本章小结

本章简略地讲述了数据库的基本概念、数据库系统的基础知识、数据库语言的构成和语法规则；从使用者的角度回顾了 MIS 系统的发展过程，简单讨论了 MIS 系统的现状并对其未来作了展望。

通过这一章的讲述，希望读者能建立起一个比较完整的数据库系统的概念；对数据库语言及其使用有初步的了解；对 MIS 系统的发展有所认识。这些内容对以后各章的学习是有帮助的。

第2章 FoxPro 2.5b 中文版概述及安装

- FoxPro 2.5b 中文版的优点、技术指标及文件类型
- DOS 和 Windows 平台开发特点比较
- 编程新概念——程序生成器
- FoxPro 2.5b 中文版运行环境及安装

2.1 从 FoxBASE+ 到 FoxPro 2.5

在美国 Fox Software 公司推出 FoxBASE+ 以前, 国内微机上流行的数据库系统几乎都是 dBASE 系列。1987 年 2 月 Fox Software 公司推出了 FoxBASE+1.00 版, 同年 7 月该公司又推出了 FoxBASE+2.00 版。FoxBASE+2.00 版不仅与 Ashton-Tate 公司的 dBASE III PLUS 完全兼容, 而且在运行速度、适用机型等方面更胜一筹。例如, FoxBASE+2.00 版比 dBASE III PLUS 快 6—7 倍, 比编译 dBASE 快 2 倍左右。

时隔一年, 1988 年 7 月 Fox Software 公司推出了功能更强、运行速度更快的 FoxBASE+2.10 版, 其运行速度比 2.00 版快 20%, 平均比 dBASE III 快 8 倍。稍后又推出了能使用 xBASE 数据库数据的 FoxGraph 绘图软件, 使 FoxBASE+ 的应用系统增色不少。

FoxBASE+ 系列虽然有不少优点, 但其用户界面不尽人意, 用户开发的应用程序仍脱离不了 FoxBASE 系统。

1989 年, FoxPro 1.0 问世, 用户界面有了较大改进, 采用了类似 Macintosh 的用户界面。但仍然没有提供将用户应用程序编译为 .EXE 可执行文件的编译工具。

1991 年, 带有编译工具包 Distribution Kit 的 FoxPro 2.0 面市, 其功能较 FoxPro 1.0 有了较大提高。除编译工具外, FoxPro 2.0 还提供了若干开发工具, 并提供了与 C 语言的接口: library Construction Kit。

1992 年 6 月 Fox Software 公司与微软公司合并后, 于 1993 年 1 月正式推出了 FoxPro 2.5 for DOS 版和 FoxPro 2.5 for Windows 版。

1994 年 11 月微软公司正式推出了符合中国市场需求的 FoxPro 2.5b 中文版。它包括 FoxPro 2.5b for DOS 中文版和 FoxPro 2.5b for Windows 中文版两个版本。

FoxPro 2.5b 中文版是以英文版 FoxPro 2.5b 为基础, 通过修改源程序代码, 从内核进行了汉化, 彻底解决了汉字双字节的问题, 避免了半个汉字的现象。字段名、窗口名和变量名都可以使用汉字。但是 FoxPro 2.5b 中文版并不是一个完全汉化的软件, 而是一个部分汉化的工具软件。其特点是:

- (1) 开发人员使用的界面保留英文, 如帮助文件、菜单提示信息、对话框等;
- (2) 可以在中文或英文操作系统下运行 (在英文操作系统下运行时, 不能显示汉字);
- (3) 用 FoxPro 2.5b 中文版开发的应用软件可以做到完全汉化, 最终用户可以工作在全汉字环境下;

(4) FoxPro 2.5b 中文版和用 FoxPro 2.5b 中文版开发的应用软件具有完整的汉字处理能力。

FoxPro 2.5b 中文版的推出,使国内微机数据库系统的应用又迈上了一个新台阶。特别是 FoxPro 2.5b for Windows 中文版能在 Windows 平台上运行,使得 FoxPro 不仅能处理图形、声音、录象等数据,而且能充分利用 Windows 的许多功能,如“动态数据交换 (DDE)”等。随着人们对它的进一步了解、使用和开发, FoxPro 2.5b for DOS 中文版和 FoxPro 2.5b for Windows 中文版的用户将会进一步增加,其应用领域也会进一步扩展。

2.2 FoxPro 2.5b 中文版的优点

除了具有上述处理双字节汉字的特点外, FoxPro 2.5b 中文版还有许多优点。归纳起来主要表现在以下六个方面。

1. 完整的系列产品

微软公司在推出 FoxPro 2.5b for DOS 中文版和 FoxPro 2.5b for Windows 中文版之前已经推出了 DOS 6.2 中文版 (PDOS 6.2) 和 Windows 3.1 中文版 (PWIN 3.1)。因此 FoxPro 2.5b for DOS 中文版和 FoxPro 2.5b for Windows 中文版可以分别运行于 PDOS 6.2 和 PWIN 3.1 而不需要其他汉字操作系统支持,实现了汉字操作系统与数据库管理系统的统一。这样就保证了系统的完整性,避免了在一种汉字系统下开发的管理信息系统换到另一种汉字系统下就可能不能正常运行的问题。

FoxPro 2.5b 中文版的完整性还表现在 FoxPro 2.5b 是一个面向多个操作系统平台的开发工具。在一个平台上(如 DOS)开发的应用软件,可以不经修改或只需作极少改动就可在另一平台(如 Windows)上运行。

2. 完备的开发工具

FoxPro 2.5b 中文版不仅具有完整的命令体系和丰富的函数(命令和函数超过 600 条),而且还配备有一套功能齐全的开发工具。这些工具包括:

- (1) 屏幕生成器
- (2) 菜单生成器
- (3) 相关查询生成器
- (4) 报表/标签生成器
- (5) 项目管理器
- (6) 文档生成器
- (7) 编译工具 Distribution Kit (DK)
- (8) 连接工具 Connectivity Kit (CK)
- (9) 与 C 或 C++ 接口 Library Construction Kit (LCK)
- (10) 转换工具 Migration Kit (MK)

一般的管理信息系统通常包括如下五个部分:控制部分、数据输入、数据检索、统计运算和结果输出。FoxPro 2.5b 中文版提供的每个生成器都对应于一般管理信息系统的某个特定部分。除了项目管理器 and 文档生成器是专为开发人员设置的以外,其他生成器的主要应用对象都可以在管理信息系统中找到。例如,菜单生成器对应于控制部分,屏幕生成

器对应于数据输入（录入界面），相关查询生成器对应于数据检索而报表/标签生成器则对应于输出部分。至于统计运算，则可在表达式编辑窗口进行编辑并检查其合法性，也可由开发人员根据用户的需求，编写具体的运算式。

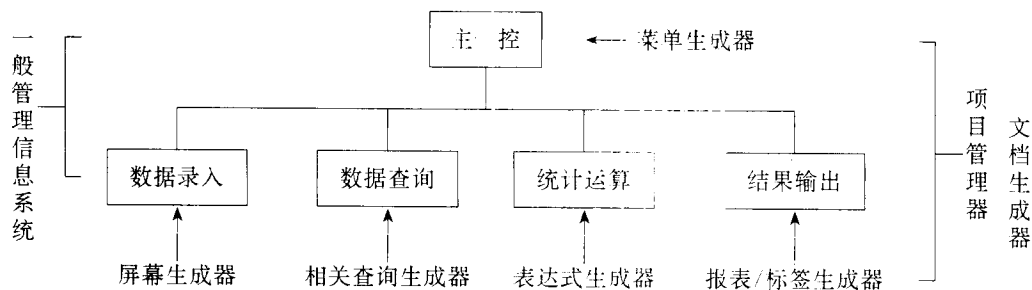


图2-1 FoxPro 2.5b 生成器与一般管理信息系统模块关系图

从图2-1可看出，对于一般的管理信息系统应用软件，开发人员可以不用手工编写程序，而直接运用FoxPro 2.5b提供的生成器，就可生成管理信息系统的各个模块。同时，项目管理器 and 文档生成器也极大地方便了开发人员，并为系统的维护提供了强有力的支持。对于复杂的管理信息系统或有特殊要求的系统，开发人员可以利用FoxPro 2.5b提供的600余条命令和函数，编写有特殊要求的模块。如果还不能满足要求，则可用FoxPro 2.5b提供的外部接口，调用其他语言编写外部模块。

3. 完善的联机帮助系统

FoxPro 2.5b 中文版提供了一整套方便实用的联机帮助系统。在FoxPro 2.5b 中文版 for Windows 中，在用户使用过程中，不管用户处在哪个窗口，只要同时按下Ctrl+鼠标左键，有关这个窗口的帮助信息就会显示在帮助窗口中。

值得一提的是，FoxPro 2.5b 中文版的帮助文件中包容了FoxPro 2.5b 中文版的所有命令和函数。每个命令、函数的使用方法都有详细解释，并附有已经录入好的例子。

在编程过程中，如果用户对某个命令或函数不太熟悉，随时可以打开帮助窗口，从中搜索自己想要了解的命令或函数。如果还想了解与该命令或函数相关的其他命令或函数，可以从SeeAlso中选取与之相关的内容。如果想进一步学习某个命令或函数的具体用法，可以从Example中查看实际用法。还可以把Example中提供的程序拷贝到一个.PRG文件中并执行该.PRG文件，观察其运行效果。这是一个很好的学习FoxPro语言的方法。

实际上，FoxPro 2.5b for Windows 中文版的帮助文件FOXHELP.HLP是一个将近两兆字节的文件，FOXHELP.DBF(.FPT)则超过两兆字节。分别是Windows风格和.DBF风格的帮助文件。其中包括了FoxPro 2.5b版的《语言参考手册》(Language Reference)的所有内容。如果手头没有FoxPro 2.5b版的《语言参考手册》的话，完全可以利用帮助文件，从中获得编程所需命令或函数的用法。

4. 完全菜单化的命令体系

FoxPro 2.5b 中文版的系统菜单功能比较全面。不太熟悉FoxPro 2.5b语言的用户可直接使用菜单完成各种操作；对于经验丰富的开发人员来说，使用菜单操作也可减少开发人员在键盘上的输入。

5. 最大限度的向下兼容能力