

计算机软件
应用系列

FoxPro 程序设计

FoxPro:

直接运行 dBASE III & IV 应用程序

丰富的编程工具

直观友好的用户界面

速度最快, 功能最强

循序渐进、以点带面、图示丰富、示例实用



● 李竹华 萧东 等编著

● 科学出版社

计算机软件应用系列

FoxPro 程序设计

李竹华 萧 东 等 编著

科学出版社

1994

(京)新登字 092 号

内 容 简 介

本书通过目前最新最优秀的数据库 FoxPro 2.5,系统地介绍了 FoxPro 的功能特点、安装配置、基本操作和使用技巧。通过学习本书,即可迅速掌握和运用 FoxPro 进行基本编程。

FoxPro 的使用过程就是 FoxPro 的程序设计过程,全书着重突出了这一新观念。学习完本书后,再辅之以一些简单的程序设计常识,就能成为一个合格的 FoxPro 高级用户。

本书内容系统全面,讲解深入浅出,书中配有精心设计的实例和大量图示。它既可作为自学 FoxPro 的教材,也可以作为大专院校和各类培训班的教材,还可供广大 FoxPro 用户参考使用。

JB425/55

计算机软件应用系列

FoxPro 程序设计

李竹华 萧 东 等 编著

责任编辑 刘晓融

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

北京理工大学印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

★

1994 年 9 月第 一 版 开本:787×1092 1/16

1994 年 9 月第一次印刷 印张:14

印数:1—4 000 字数:326 000

ISBN 7-03-004199-2/TP·376

定价:15.90 元

目 录

第一章 数据库背景知识	1
1.1 什么是数据库	1
1.2 谁使用数据库	1
1.3 为什么要使用数据库	1
1.4 三种数据库	2
1.5 Xbase 历史	3
1.6 数据库基本概念	5
1.6.1 数据工作区	5
1.6.2 数据库别名	5
1.6.3 记录指针	6
1.6.4 数据类型	6
1.6.5 变量	7
1.6.6 常量	8
1.6.7 函数	8
1.6.8 运算符	8
1.6.9 表达式	9
1.6.10 保留字	10
1.7 数据库基本操作	10
1.8 文件类型和扩展名	11
1.9 小结	12
复习思考题	12
第二章 FoxPro 概貌	14
2.1 FoxPro 的版本与特征	14
2.2 安装 FoxPro	16
2.3 启动 FoxPro	18
2.4 FoxPro 的用户界面	18
2.4.1 菜单	18
2.4.2 对话框	20
2.4.3 系统窗口	21
2.4.4 命令窗口	22
2.4.5 消息	23
2.4.6 热键和 Ctrl 简捷键	24
2.5 FoxPro 的工具	24
2.5.1 关系型按例查询	25
2.5.2 图形化的浏览窗口	25

2.5.3	应用程序生成器 FoxAPP	25
2.5.4	RUN 菜单	25
2.5.5	屏幕生成器	26
2.5.6	报表生成器	26
2.5.7	菜单生成器	26
2.5.8	项目管理器	26
2.5.9	联机式上下文敏感帮助	26
2.6	FoxPro 帮助系统	26
2.7	退出 FoxPro	28
2.8	小结	29
	复习思考题	29
第三章	建立数据库	30
3.1	问题的提出	30
3.2	定义数据库结构	31
3.3	保存数据库结构	33
3.4	修改数据库结构	34
3.5	输入记录	35
3.6	文本编辑器	36
3.6.1	文件管理	37
3.6.2	光标移动	37
3.6.3	输入文本	37
3.6.4	选定文本	37
3.6.5	删除文本	38
3.6.6	插入文本	39
3.6.7	移动和拷贝文本块	39
3.6.8	查找和替代	39
3.6.9	复原和重做	40
3.6.10	Preferences 对话框	40
3.6.11	输入备注型字段内容	41
3.7	其他数据库及汉字	41
3.8	小结	42
	复习思考题	42
第四章	浏览数据库	44
4.1	打开数据库	44
4.2	漫游浏览窗口	46
4.3	浏览备注型字段内容	48
4.4	修改浏览窗口	49
4.4.1	移动字段显示位置	49
4.4.2	改变字段显示宽度	50

4.4.3	Change 方式下的浏览窗口	50
4.4.4	只显示指定字段	51
4.4.5	调整窗口分区	52
4.5	关闭浏览窗口	54
4.6	关闭数据库	54
4.7	小结	54
	复习思考题	54
第五章	修改数据库	56
5.1	增加记录	56
5.2	定位记录	57
5.3	更新记录	60
5.4	全局更新	60
5.5	删除记录	61
5.6	使用 Change 方式	63
5.7	小结	63
	复习思考题	63
第六章	单数据库查询	65
6.1	启动 RQBE 窗口	65
6.2	建立查询	66
6.2.1	指定数据库	66
6.2.2	指定显示字段	66
6.2.3	指定查询结果送往何处	68
6.2.4	指定查询条件	69
6.2.5	查看 SQL 语句	70
6.3	执行查询	71
6.4	修改查询	71
6.5	保存查询	72
6.6	执行保存下来的查询	72
6.7	小结	73
	复习思考题	73
第七章	建立简单报表	74
7.1	启动报表布局窗口	74
7.2	建立快速报表	75
7.2.1	定义快速报表格式	75
7.2.2	预览快速报表	76
7.3	建立自定义报表	77
7.3.1	基本知识	77
7.3.2	定义字段列的标题	79
7.3.3	调整字段宽度和位置	80

7.3.4	定义报表标题	81
7.3.5	指定篇眉和脚注	81
7.3.6	调整文本字体	81
7.3.7	增加文本和图片	82
7.3.8	报表最终样式	83
7.4	保存报表	83
7.5	页布局设置	84
7.6	打印报表	84
7.7	根据查询结果制作报表	84
7.8	小结	85
	复习思考题	85
第八章	建立邮政标签和规范化信件	86
8.1	启动标签布局窗口	86
8.2	定义标签格式	87
8.3	定义标签内容	88
8.3.1	增加字段对象	88
8.3.2	增加文本对象	89
8.3.3	调整文本格式	89
8.4	预览标签	90
8.5	保存标签	90
8.6	打印标签	90
8.7	建立规范化信件	91
8.8	小结	92
	复习思考题	92
第九章	排序和索引	94
9.1	排序	94
9.1.1	按一个字段对数据库排序	94
9.1.2	按多个字段对数据库排序	95
9.1.3	按含有数字的字符型字段排序	97
9.1.4	排序的局限性	98
9.2	索引	99
9.2.1	单项索引文件和复合索引文件	99
9.2.2	建立结构化的复合索引	99
9.2.3	浏览索引过的数据库	101
9.2.4	建立独立的复合索引文件	103
9.2.5	用 Seek 查找记录	103
9.2.6	修改索引	104
9.2.7	重建索引	104
9.3	小结	104

复习思考题	105
第十章 连接数据库	106
10.1 为什么要连接数据库	106
10.2 怎样连接数据库	106
10.3 浏览连接后的数据库	108
10.4 以一对多的关系连接数据库	109
10.5 保存数据库间的连接	110
10.6 取消连接	111
10.7 小结	111
复习思考题	111
第十一章 多数据库查询	113
11.1 启动 RQBE 窗口	113
11.2 建立查询	113
11.2.1 指定数据库	113
11.2.2 指定显示字段	115
11.2.3 指定输出目标和查询条件	116
11.2.4 查看 SQL 语句	116
11.3 执行查询	116
11.4 修改查询	117
11.5 保存查询	119
11.6 退出 RQBE 窗口	119
11.7 小结	119
复习思考题	119
第十二章 建立复杂报表	121
12.1 使用页布局设置对话框	121
12.2 报表标题	121
12.3 制表日期和页号	122
12.4 编辑日期格式	123
12.5 对记录进行分组	125
12.6 对报表字段进行格式定义	126
12.7 建立摘要区	127
12.8 筛选记录	128
12.9 小结	128
复习思考题	129
第十三章 建立用户屏幕	130
13.1 问题的提出	130
13.2 启动屏幕设计窗口	130
13.3 设计屏幕	132
13.3.1 指定屏幕整体属性	132

13.3.2	移动字段对象	133
13.3.3	画线和框	134
13.3.4	改变字段标题	134
13.3.5	修改字段标题	134
13.3.6	改变字段显示顺序	135
13.4	保存屏幕	135
13.5	生成屏幕代码	135
13.6	运行屏幕	135
13.7	建立按钮	136
13.7.1	建立 Exit 按钮	136
13.7.2	建立 Next 和 Previous 按钮	137
13.8	出错处理	139
13.9	小结	139
	复习思考题	139
第十四章	增加屏幕功能	141
14.1	建立弹出控制	141
14.2	建立一个通用的按钮屏幕	142
14.3	修改用户输入屏幕	144
14.4	组合屏幕	145
14.5	小结	147
	复习思考题	147
第十五章	建立菜单	148
15.1	启动菜单设计窗口	148
15.2	指定菜单整体特性	148
15.3	定制菜单	150
15.3.1	定制顶层菜单	150
15.3.2	定制第二层菜单	151
15.4	测试菜单	153
15.5	保存菜单	153
15.6	小结	154
	复习思考题	154
第十六章	用项目管理器集成应用程序	155
16.1	创建新项目	155
16.1.1	打开项目窗口	155
16.1.2	向项目中增加文件	156
16.1.3	建立项目	158
16.2	创建应用程序	158
16.3	运行应用程序	158
16.4	小结	160

复习思考题	160
第十七章 使用 FoxPro 宏和内部辅助程序	161
17.1 宏	161
17.1.1 缺省宏	162
17.1.2 建立宏	163
17.1.3 编辑宏	166
17.1.4 删除宏	166
17.1.5 保存宏	167
17.1.6 缺省宏文件	167
17.2 文件管理器 Filer	167
17.2.1 文件管理器窗口	167
17.2.2 标记文件	169
17.2.3 使用 Files Like(选择文件)对话框	170
17.2.4 查找文件	171
17.2.5 排序文件	172
17.2.6 目录树形结构	173
17.3 计算器 Calculator	173
17.4 日历/日记簿 Calendar/Diary	175
17.5 特殊字符表 Special Characters	175
17.6 ASCII 字符图 ASCII Chartt	175
17.7 屏幕捕获程序 Capture	176
17.8 小结	177
复习思考题	177
第十八章 定制和优化 FoxPro	179
18.1 启动开关和装入程序	179
18.2 启动文件	180
18.3 FoxPro 中的内存	181
18.3.1 内存类型	182
18.3.2 内存和 FoxPro 标准版本	182
18.3.3 内存和 FoxPro 扩充版本	182
18.4 优化 FoxPro 的其他考虑	183
18.4.1 CONFIG. SYS	183
18.4.2 启动速度	183
18.4.3 文件和目录	184
18.4.4 空闲磁盘空间	185
18.4.5 使用 EMS 的 RAM 盘和磁盘 cache	185
18.4.6 数学协处理器	186
18.4.7 内存常驻程序	186
18.4.8 FoxPro 装入程序	186

18.4.9 多用户环境中的 FoxPro 和临时文件	186
18.4.10 命令设置	187
18.5 小结	188
复习思考题	188
附录 A 最终程序清单	189
附录 B FoxPro 简捷键和控制键	197
附录 C FoxPro 出错消息	198

第一章 数据库背景知识

1.1 什么是数据库

许多人提到数据库时只是想到计算机里的数据库,其实我们身边就有各种各样的数据库。日常所用的英文字典,就是一个典型的数据库,它能帮助用户确定一个英文单词的拼写、含义和各种习惯用法,又快又方便。其他的如亲友通信录、电话号码簿、食谱、电视节目报等也是典型的数据库。总之,数据库是关于某个主题的信息集合,可以对其加以组织,使用户能够容易地查找、添加和修改其中的数据。

前面所举的几个数据库例子,都是以纸张为存储媒介的,它们是广泛意义上的数据库。若将它们组织成计算机文件,即以计算机磁盘为存储媒介,就变成了计算机中所使用的数据库。数据库计算机化后,就可以利用计算机对其进行处理了。本书以后章节中所指的数据库,一般都是指计算机化的数据库,即以磁盘文件形式存储在计算机中的数据库。

1.2 谁使用数据库

对数据库用户的要求要比电子表格和字处理用户高得多。由于数据库系统的复杂性以及数据库管理系统的专业性,所以,常常需要用户学会一门计算机语言,才能充分地发挥其功效。因此,开发数据库的公司或企业经常聘请或委托专业开发人员开发相应的数据库应用程序。也就是说,有各种层次的数据库用户,既有可以对数据库什么也不了解的数据录入员,也有非常熟悉数据库管理系统的数据库开发人员,具体如下:

- 数据录入员和初学者。负责使用数据库来完成任务,如管理客户文件、把邮件合并生成规范化信件等。通常情况下,这些用户使用别人开发的应用程序来工作,一般不需对数据库有太多的了解。
- 中级用户。负责使用数据库处理信息,如建立一些关系较复杂的报表等,但常常需要通过查看手册来进行。
- 高级用户。负责根据他们所掌握的知识设计开发使用方便的软件产品。许多中级用户都想达到这种程度,但常常因这种或那种原因而不能如愿。现在,有了 FoxPro,他们有可能如愿了。随着对 FoxPro 的熟悉,许多人都惊叹 FoxPro 的强大而灵活的开发工具。
- 开发人员。负责为其他用户(例如录入员)建立数据库应用程序,供他们使用。开发人员也许和产品关系不是最大,但必须拥有进行系统设计的专门知识。

1.3 为什么要使用数据库

假定某厂有人事科、劳资科和情报科,人事科负责管理全厂每个职工的人事信息,具体

包括职工号、姓名、性别、出生日期、文化程度、政治面貌、技术职称、入厂日期、所在科室、家庭住址和其他等；劳资科负责管理全厂每个职工的月工资和奖金信息，具体包括职工号、月份、工资、奖金和说明等；情报科负责管理全厂所有技术图书信息，具体包括书号、书名、作者、出版社、出版日期、定价等，除此之外，情报科还负责记录每本图书的借阅情况，即哪本书什么时候由哪个职工借阅的。

事实上，一个工厂要管理的信息远远不止这些，但就是这些信息，如果用人工管理或文件管理手段，其工作量也是非常大的，并且常常费力不讨好。比如说，有一个职工因故调出该厂，看到批文后，人事科便删去了有关这名职工的记录（如注销该职工卡片），但劳资科和情报科并不知道此事，这样，劳资科的台帐上还有这名职工，这还不会造成什么危害，至多不过是一条废记录而已。然而，情报科的图书借阅登记台帐上也许还记载着这名职工借阅了几本书没有归还，按照规定，应该先办归还图书手续，然后再办调离手续，事实上却经常因为没有强制性手段而违反这个规定。

造成这些弊端的主要原因是数据冗余大，即同样的信息重复在多处存放，很难保证这些信息总是一致的。在本例中，有关职工的信息，人事科、劳资科和情报科都要保存，而这些部门又是平行单位，根本不可能要求他们同时修改同一条职工记录，以保证信息的一致性。

将这些信息组织成数据库后，情况就不同了。有关职工的信息只存放在一个数据库中，人事科、劳资科、情报科不再需要保存这些信息。需要使用时，只要调用这个数据库即可。注意，这三个科室调用的是同一数据库。这样，数据的一致性就自然得到了保证，要么全修改，要么全不修改。

消除冗余，对数据集中管理是使用数据库的最大好处。对数据库应用程序开发人员而言，还有一个数据独立性强的好处。总之，使用数据库的基本思想就是要对所有数据实行统一的、集中的、独立的管理，数据独立于程序而存在，并可以提供给各类不同用户共享使用。

1.4 三种数据库

数据库的历史和整个计算机工业的历史紧密相关，在整个数据库发展过程中，起码出现了三种类型的数据库：层次数据库、网络数据库和关系数据库。

在60年代第一台现代化的计算机开发成功，其上就有层次数据库 IBM System/360 在运行。由于这些早期的计算机非常昂贵，其中计算机耗电和数据存储又要占去大部分份额，所以，当时最关心的就是空间占用和执行速度。

层次数据库之所以流行起来，就是因为它们执行速度快和存储空间利用率高。在这样一个数据库中，数据的组织方式就像一棵倒立的树——一系列用枝条连接起来的结点。这些系统的使用并不容易，为得到一个数据，用户必须使用一个复杂的网状关系。当数据用一种自顶向下的一致的方式来处理时，若数据间的关系不需要变化，这种方法还是可取的。问题是，实际情况常常不是这样的。

在70年代，DEC 等公司开发出了小型机，同时，层次数据库发展成了网络数据库。在这样的数据库系统中，允许更复杂的数据关系存在，除了自顶向下的父子关系外，还支持预先定义关系的完整性等高级特点。当然，要想处理数据库中的数据，用户必须清楚网络

关系。

在 70 年代中期,在 IBM 公司供职的 E. F. Codd 发表了一篇研究论文,提出了一种新的数据库结构,即基于数学方式的关系数据库。按照这种理论,IBM 公司开发出了关系数据库 SYSTEM/R,后来发展成了 SQL/DS 和 DB2。1978 年,Oracle 公司推出了第一个商品化的关系数据库系统 Oracle。

关系数据库系统的核心是表,表是与某个事物相关的信息的集合。比如,有关职工的人事信息就可构成“职工”表,有关图书的信息可构成“图书”表。表又依次分成记录,记录给出了表中某一项的信息,记录又分成若干字段,字段中存放的是记录中某一方面的信息。记录就好像是日常生活中所用的卡片,而字段就像是每张卡片上的“姓名”、“年龄”等项。在“图书”表中,每一条记录都对应于一本图书,记录中的每一个字段对应于每一个具体的图书信息项,如“书号”、“书名”等。

用过 dBASE 的人都知道,在 dBASE 中没有“表”这个概念。在数据库专家眼中,数据库是由许多表组成的,但在 dBASE 看来,问题可以简化,它直接把表称为数据库,就相当于每个数据库仅由一个表组成。FoxPro 也沿用了这种习惯,没有使用“表”概念,应注意区别。

早期的关系数据库的执行性能要明显低于层次和网络数据库,但是,随着近年来硬件制造工艺水平的提高和各种软件开发新技术的引入,关系数据库的性能已得到了很大的改进,即使拿 PC 机上实现的关系数据库与其他层次和网络数据库相比,也逊色不了多少。目前,商品化的数据库系统基本上都是关系型的。

随着 PC 机的普及,PC 机上的关系数据库系统在整个数据库系统中的份量变得越来越重要。

1.5 Xbase 历史

谈起 PC 机数据库系统,就不能不谈及 dBASE。dBASE 是 Ashton-Tate 公司(该公司现已并入 Borland 公司)的系列产品,具体有 dBASE I, dBASE II, dBASE III plus 和 dBASE IV 等。dBASE 软件在微机数据库领域获得了非凡的成功,成了事实上的标准。其他厂家推出的微机数据库软件也纷纷声称自己与 dBASE 兼容,是 dBASE 式的产品,使用了 dBASE 语言,结果导致说起 dBASE 时反而不知具体指的是哪一家产品,造成了很大的混乱。为了消除这种混乱,于 1990 年推出了术语 Xbase,它专指各种与 dBASE 兼容的软件产品和程序语言。

为了帮助大家弄清各种 Xbase 产品,有必要将 Xbase 的发展过程简单介绍一下。

在 70 年代后期,计算机爱好者们意识到了 PC 将会被大量用于商务应用。当时,虽然借助于 Microsoft BASIC 语言(PC 机上使用的第一个应用程序编程语言),人们能够建立起实用的应用系统,但是,在程序中存储和检索数据还比较困难。人们迫切希望能有一种数据库管理系统,它能集中控制和管理计算机的数据资源,使得各个应用程序可以共享数据资源,它能提供一些强有力的开发工具,使得应用程序开发更为容易,更具效率。鉴于此种情况,许多软件开发商和软件开发者在寻求着一种理想的数据库管理系统。

在数据库软件的早期,关系型数据库管理系统与层次型数据库管理系统都是可选的方案之一。但是,层次型数据库不灵活,设计数据库时需要许多专门知识,并且一旦开发完一个应用系统,就几乎不可更改了。而关系型数据库则灵活得多,数据库可以根据需要很容易地

进行修改。因此,随着时间的推移,层次型数据库逐渐被淘汰。

1978年,Wayne Ratliff正在美国加州Pasadena的一家实验室工作。当时有一个称为JPL-DIS的文件系统,Wayne就是利用这个文件系统作为模型开发了一个用于PC机的关系型数据库系统,称为Vulcan。1980年他联合了两个软件销售商Hal Lashlee和George Tate,重新开发了这个软件与文档,改名为dBASE II,并将该产品向迫不及待的用户发布了。

利用BASIC来编制商用程序,常常因为文件的处理与数据库操纵不易实现而受挫。著名的以“点为提示符”的dBASE II给PC数据库市场带来了一场革命性的变化。只要学习一些简单的命令,用户就能够建立信息的逻辑存储结构,并且能够快速检索它们。有经验的程序员还可以将这些命令存入文件,之后作为程序来运行,程序可以带有输入屏幕、数据有效性检查、逻辑判断和打印格式报表等功能。以前很难想像会有“使用如此简单的方法就能完成任务”的程序语言,但是,在1982年dBASE II做到了。

随着dBASE II的应用越来越广泛,其缺点也越来越明显,最明显的缺点是用户需要学习和记住越来越多的命令,另一方面,人们希望开发更好更复杂的应用程序。于是,又有了一些新产品陆续推出,以满足这些要求。但是,在微机数据库市场dBASE II一直保持着强有力的领导者的地位,它成功地为大多数用户解决了大部分问题。

1984年,Ashton-Tate公司推出了dBASE III,它一出现便成了事实上的数据库软件标准。dBASE III既有强大的数据处理功能,又有数据库操纵的灵活性,使用它提供的工具,用户不用编程就能建立自己的查询、报表和标签。

dBASE III及dBASE III plus是所有dBASE产品中最持久的产品。虽然几年之后数据库市场发生了不小的变化,但是它的销售额仍然超过其他产品。其成功的关键在于它集中地解决了大多数用户都关注的问题。它继续面向低档微机并且已经成为很多公司和政府机构的既定标准。

1986年,Fox Software公司推出了FoxBASE,这是一个比dBASE快很多倍的产品,它曾作为加速dBASE III plus程序执行的辅助产品而销售,结果极其成功。开发者们马上就采纳了FoxBASE,因为FoxBASE能以更快的速度运行它们现有的应用程序,同时缩短了开发周期。FoxBASE如此完美地运行dBASE III plus程序,惊动了Ashton-Tate公司,甚至要以“模仿”为由控告Fox Software公司。

1988年下半年,由于一再延期交付dBASE IV,1.0版又有一些严重的毛病,以及迟迟没有推出相应的编译器,导致最新的dBASE IV引入后,Xbase的市场占有率反而开始下降。而1989年和1990年则是Fox Software公司最为成功的两年,许多开发者和最终用户纷纷选择Fox Software公司推出的功能比FoxBASE更强的FoxPro,一致称赞FoxPro的执行速度和图形用户界面。其他Xbase产品的销售在这期间也不错,但是总的来说,整个数据库市场由于dBASE IV的失败而失去了很多用户。

1991年与1992年是一段团结与合并的时期。1991年Borland收购了Ashton-Tate公司,1992年Microsoft收购了Fox Software公司,Computer Associates公司收购了生产Clipper的Nantucket公司,Borland又于92年下半年继续收购了生产Arago和其他Xbase产品的Word Tech公司。大公司的加入,增强了Xbase产品的可信度。1993年,又有许多Windows环境下的Xbase产品面世,进一步增强了Xbase产品。

Fox Software公司并入Microsoft公司后,又在原FoxPro 2.0版本的基础上,推出了最新

的 FoxPro 2.5 版本。作为 Xbase 产品,它能直接运行 dBASE III 和 dBASE IV 应用程序,又新增了图形用户界面和丰富的工具,是目前速度最快、功能最强的 PC 机数据库管理软件。

在 1.2 节已介绍过,有许多用户由于各种原因不能成为数据库开发人员,但他们非常希望能成为数据库的高级用户,以便更好、更充分地使用数据库。在 FoxPro 出现之前,只有极少数用户能克服受挫失望情绪和重重困难,最终达到目的。无论是成功的用户还是未成功的用户,最后得出了同一个结论:如果说成为中级用户已属不易,那么成为高级用户几乎已是高不可攀。请想想,有多少非专业人员能记住那么多的命令和复杂的程序设计知识?

FoxPro 2.5 正是面向这种需求而出现的,它提供了许多功能极其丰富的工具。比如说,只要在菜单或图标上圈圈点点,就可完成基本的数据库操作,无需记忆任何命令。也就是说,成为 FoxPro 中级用户非常容易。再比如说,利用 FoxPro 提供的屏幕生成器或报表生成器,再记忆一些非常简单的命令(注意,数量极少),就能制作出非常漂亮的屏幕和报表,同专业人员开发出来的屏幕和报表相比,毫不逊色。实际上,由于 FoxPro 2.5 以上版本所提供的丰富的开发工具,已使 FoxPro 的使用过程和 FoxPro 的程序设计过程完全地融合到了一起。因此,现在有很多开发人员已“退步”到了高级用户的水准。

1.6 数据库基本概念

1.6.1 数据工作区

数据库文件在使用之前先要打开。在 FoxPro 中,可以同时打开和使用多个数据库文件。为了使这些同时使用的文件不致于互相干扰,并便于识别,FoxPro 采用不同的数据工作区把它们分隔开。FoxPro 的扩充版本共有 225 个工作区,标准版本有 25 个工作区,每个工作区只能打开一个数据库文件,因此允许同时打开的数据库文件数最多也是 225 个。

数据工作区是以编号或名称来标识的。数据工作区的编号分别为 1~225 或 1~25。

FoxPro 的用户虽然可以同时使用多个数据工作区,但在任一时刻只有一个工作区是活动的,称为当前工作区。当前工作区中的数据库文件称为当前数据库文件。FoxPro 中绝大多数数据操作都是针对当前数据库进行的,如果需要同时使用其他工作区数据,必须在该数据名称之前冠以所在工作区的编号或名称或下面就要介绍的数据库别名。

FoxPro 初始状态时默认 1 号工作区为当前工作区,用户也可指定其他工作区为当前工作区。

1.6.2 数据库别名

FoxPro 对非当前工作区的数据进行操作时,固然可以用该工作区名称来标识它,但是由于工作区的名称是固定的,例如 2 号工作区的名称总是 B,因而不利于数据与操作功能的相对独立。于是,FoxPro 让每个已打开的数据库文件都有一个别名(Alias),用户就可以用别名代替工作区名,来标识非当前工作区数据库。由于别名是在数据库打开时,由用户任意指定的,不仅灵活,而且与所在工作区没有必然的联系,因此容易实现数据与操作功能的独立。例如,假定某个数据库文件的别名被指定为 S2,那么无论它是在哪个工作区打开的,只要选择该数据库(通过别名选择),就会自动将相应的工作区置为当前工作区。

数据库别名,由一个不超过8位的不含空格的字符串构成。数据库别名一般是在数据库文件打开时由用户命名的,如果用户不命名,则FoxPro自动把该数据库的文件名(不包括扩展名)指定为它的别名。

对于不想学习FoxPro编程的一般用户而言,可以不必理会数据库别名,因为选择数据库和选择工作区是通过很形象的菜单实现的。

1.6.3 记录指针

数据库文件一般都有很多条记录,FoxPro进行数据库操作时,虽然可以对数据库中全部记录或部分记录进行连续处理,但是在处理中总是逐条记录进行的。因此,对于一个正打开着的数据库文件来说,在任一时刻都有且只有一条记录是数据处理的当前对象,它或者正等着处理,或者正在处理,这条记录被称为当前记录。FoxPro为每个数据工作区分别设置了一个记录指针,用以标识各工作区中的当前记录。记录指针中存放的是记录在数据库文件中的序号,即记录号,因此,如果某工作区中没有数据库文件打开,不存在当前记录,该工作区记录指针的值就为0,如果有数据库文件打开,那么记录指针的值就是当前记录的记录号。当记录指针向前移动,超过数据库文件中最后一条记录,指向文件尾时,它的值比数据库文件中最大的记录号(相当于数据库文件中的记录数)大1。当记录指针向后退,退过数据库文件中首记录,指向文件头时,它的值为1。

在浏览窗口中浏览数据库记录时,屏幕上将会出现一个标记,用以标出记录指针的当前位置。

1.6.4 数据类型

数据库操作的对象是大量的信息,信息的表达形式就是数据。数据按其构造、处理方法、用途及基本属性,分为若干不同的类型。数据库操作的基本原则就是,只有相同类型的数据之间才能进行操作,因此,数据类型是一个非常重要的基本概念。

FoxPro数据类型有6种:字符型、数值型、浮点型、日期型、逻辑型和备注型。这些类型的构成和使用规则如下:

(1) 字符型(Character):字符型数据由可打印的字符组成,字符型数据之间不能进行数学运算,但可以用“+”或“-”号把几个字符型数据连接起来,还可以按其字符的ASCII码进行大小比较。字符型数据含有的字符个数称为该字符型数据的长度(或宽度),最大长度是254,缺省为10。字符型数据一般用于表示非计算和非日期型的信息,例如产品名称、图书名称、职工号、邮政编码等。在定义数据库字段时,这是缺省的字段类型。

(2) 数值型(Numeric):数值型数据只能由阿拉伯数字、小数点和正负号构成。数值型数据之间可以进行数学运算,数值的最大精度是16位有效数字。数值型数据的最大长度是20,其中包括可选的符号位和小数点,缺省长度是8。数值型数据一般用于表示需要进行计算的数据,如数量、金额、成本、比率、利息等。

(3) 浮点型(Float):浮点型数据基本上与数值型数据相同,只是其最大精度超过了16位有效数字。浮点型数据一般用于表示数据精确度要求比较高的科学和工程数据,如扩展系数、原子重量等。

(4) 日期型(Date):日期型数据一般用于表示日期的数据,如起始日、截止日、生日等。