

# *FoxPro 2.5 for Windows* 高级应用程序开发指南



—— 如何开发全功能的数据库应用程序

[美] Nelson King 著

王晓伟 张振宇 译  
任洪江 校



- 利用菜单和程序设计来掌握FoxPro的数据管理
- 在FoxPro for Windows中扩充图形功能
- 构造功能全面的强有力应用程序



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONIC INDUSTRY

FoxPro 2.5 for Windows

## 高级应用程序开发指南

——如何开发全功能的 FoxPro 2.5 应用程序

|     |             |   |
|-----|-------------|---|
| 〔美〕 | Nelson King | 著 |
|     | 王晓伟 张振宇     | 译 |
|     | 任洪江         | 校 |

电子工业出版社

(京)新登字 055 号

## 内 容 简 介

本书系统介绍了开发 FoxPro 2.5 for Windows 高级应用程序的技术和方法,主要目的在于帮助读者完成从菜单用户到应用程序开发者的过渡。本书以循序渐进的方式对 FoxPro 2.5 for Windows 这一优秀的微机数据库管理系统进行了详尽细致的介绍,并附以大量代码进行了如何开发高级应用程序的说明。全书分为四个主要的部分,分别是“使用 FoxPro 菜单系统”、“基础程序设计”、“应用程序的开发”和“建立完整的应用程序”。为方便读者学习,本书还随附一张软盘,向读者提供了示例代码。

本书适于使用 FoxPro 2.5 for Windows 的所有用户,无论是初学者还是已有编程经验的高手,本书均会对读者有所帮助。

本书英文版书名为“FoxPro 2.5 for Windows: Developing Full-Scale Applications”,由美国 M&T Books 出版公司出版。该书的中文出版版权于 1994 年 10 月经美国远东图书公司(Far East Books, Inc.)授予电子工业出版社,翻译及文字处理工作由美国远东图书公司完成并经电子工业出版社进行了审定。未经出版者同意,任何人不得以任何手段复制或抄袭本书的内容。

Copyright © 1994 by M&T Books.

Copyright © of Chinese version 1994 by Publishing House of Electronic Industry.

## 《FoxPro 2.5 for Windows 高级应用程序开发指南 ——如何开发全功能的数据库应用程序》

[美] Nelson King 著

王晓伟 张振宇 译

任洪江 校

责任编辑:王世忠

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

中国科学院印刷厂印刷

\*

开本:787×1092 毫米 1/16 印张:35.5 字数:908.8 千字

1995 年 2 月第一版 1995 年 5 月第二次印刷

印数:6000—12000 册 定价:72.00 元

ISBN 7-5053-2765-8/TP·873

# 引 言

这本书写给那些想去开发 Windows 环境下 FoxPro 2.5 功能的朋友们,它的主要目的在于帮助读者完成从菜单用户到应用程序开发者的过渡。这不是件容易的事情,但结果会大受欢迎。目前急需寻找建立客户商业应用程序的有效方式,也急需具有工作技能的人员。FoxPro for Windows 软件开发环境提供了完成该任务的数据管理和编程工具。但是请记住,由于 FoxPro for Windows 的丰富、复杂和功能强大,掌握它需要很长的学习过程。

这本书也许不是给初学者的,它不一一解释每个术语,只是粗略地叙述数据库概念,并假设读者熟悉 FoxPro for Windows 的启动、菜单使用、窗口使用和其他简单操作。

在开始接触 FoxPro for Windows 时,很容易进入那些看上去似乎非常重要的部分,比方说屏幕生成器,但不会意识到成功地使用它们至少需要两个先决条件:深入理解数据库的管理和 FoxPro 的编程技术。入门教程使得使用屏幕生成器看起来很容易,但接触了象代码片(code snippet)那样的东西以后,就会问代码片是什么?这正是许多专职程序员在 FoxPro 2.0 for DOS 中首次见到代码片所问的问题。他们熟悉代码段(code fragment),但是却不知道 Fox 用代码片做什么。学习编程就是学习这些技术。

这说明了学习象 FoxPro for Windows 这种复杂的软件开发系统具有相当的难度。许多重要的语言要素和编程工具具有易于掌握、使用简单的特性,满足初学者的使用,但它们同时具有特别完备的功能,更适于专家使用。不幸的是,参考手册并不对此进行区分,只是列出所有的东西,让用户自己去归类了解所使用的内容,这对于没有经验的人是很困难的。

本书试图以几种方式进行归类。首先,遵从自然进程,它按专业知识的大致递增方式和人们使用 FoxPro 的模式,分为四个部分。尽管本书从一般意义上来讲并不是一个教程,但其中采用了“搭积木”的方式。

- **使用系统菜单:**它从菜单选项开始来探索基本概念。它提供了数据库管理和 Xbase / FoxPro 编程语言的介绍。
- **基础编程:**它在命令窗口和简单程序文件中使用了 Xbase/FoxPro 语言的大多数重要命令和函数。
- **开发应用程序:**它进入了程序设计的更为复杂的方面,特别是用户接口,覆盖了 FoxPro 的有力工具。
- **建立完整的应用程序:**它涉及了建立大规模应用程序所包括的所有问题,例如安全性、网络操作、数据字典等。

第二,本书通过一系列编程和数据库问题,强调了上下文关系,例如:

- 为什么它在这儿,它来自何处?
- 它用于什么方面?
- 它不能用于什么方面?

• 它有替换吗?

第三,本书强调了数据库管理,尤其是创建数据库应用程序中固有的决策过程。这个过程从设计开始,接着编程、测试、安装,甚至于销售和训练。即使接触一个比较简单的数据库项目,也要考虑很多步骤和事情。

总而言之,本书的对象是那些想学习 FoxPro 编程和想在 Microsoft Windows 下建立自己应用软件的人员,读者可在任何应用层次上使用此书,它可提供 FoxPro for Windows 数据库管理方面的一些见识、提示、技巧和系统性的观点。

书中的大量代码适用于个人使用,其中包括代码片、例子和辅助函数。随附的软盘包含了“系统维护”模块的全部内容:

- 数据字典管理
- 打印/输出管理
- 安全性系统
- 系统代码管理

它包含了多于 1MB 的 FoxPro for Windows 代码,各模块相互独立,并且很容易插入用户自己的应用程序。本书第四部分的绝大篇幅不管是在说明编程概念还是在描述应用方法上,都与该软件密切相关。充分利用盘上已完成的软件,可以省去键入许多代码。

在计算机世界中,几乎没有比数据库管理更复杂的事情了,作为最终用户系统和完整软件开发环境的 FoxPro for Windows 就更是如此。用户从表面上所看到的仅是所含主要软件的问题:面向对象的程序设计、客户/服务器数据库、用户接口设计以及应用开发等等。

FoxPro 2.5 for Windows 具有微机事务中最快的数据库机制,即数据组织和操作程序。FoxPro 的速度已是名声在外,经受了一次又一次的挑战,直到今天,它仍能维持着全部的性能优点。

FoxPro for Windows 提供了好几种保持这种优点的方法。最简单的方法是使用 System Menu 的 ENDDO,这包括了各种基本数据管理操作,如创建文件、编制索引和数据输入等。第二种方法继承了 DOS 版本的部分特性,它通过 Command Window 的 ENDDO 来进行,在那里,数据库管理命令可以直接显示或执行(假设知道那些命令是什么)。第三种方法是编写程序,FoxPro for Windows 实现了 Xbase 编程语言的一个超级集合,它包括多于 620 条命令、函数、操作符,被许多工具所支持。使用 FoxPro for Windows,既能建立小型的个人应用程序,也能建立几乎可任意大小、任意复杂程度的专业应用程序。

# 目 录

## 引 言

### 第一部分 使用 FoxPro 系统菜单

|                          |               |
|--------------------------|---------------|
| <b>第一章:文件组织 .....</b>    | <b>( 5 )</b>  |
| 1.1 表结构窗:创建 DBF .....    | ( 5 )         |
| 1.2 使用观察窗口 .....         | ( 10 )        |
| <b>第二章 数据输入和编辑 .....</b> | <b>( 25 )</b> |
| 2.1 变量 .....             | ( 25 )        |
| 2.2 表达式 .....            | ( 26 )        |
| 2.3 输入工具:Append .....    | ( 36 )        |
| 2.4 输入工具>Edit .....      | ( 38 )        |
| 2.5 编辑正文 .....           | ( 41 )        |
| 2.6 正文菜单 .....           | ( 41 )        |
| <b>第三章 表的操作 .....</b>    | <b>( 47 )</b> |
| 3.1 基本操作选项 .....         | ( 47 )        |
| <b>第四章 数据检索和输出 .....</b> | <b>( 51 )</b> |
| 4.1 检索的准备 .....          | ( 51 )        |
| 4.2 RQBE .....           | ( 56 )        |
| 4.3 报表生成器 .....          | ( 70 )        |
| 4.4 标签设计器 .....          | ( 93 )        |
| 4.5 汇总计算 .....           | ( 93 )        |
| 4.6 Copy To 选项 .....     | ( 97 )        |
| <b>第五章 辅助菜单项 .....</b>   | <b>( 99 )</b> |
| 5.1 键盘宏 .....            | ( 99 )        |
| 5.2 帮助菜单 .....           | ( 101 )       |
| 5.3 运行菜单 .....           | ( 102 )       |

|                               |              |
|-------------------------------|--------------|
| <b>第六章 FoxPro 数据库机制 .....</b> | <b>(103)</b> |
| 6.1 Rushmore 技术 .....         | (103)        |

## 第二部分 基础程序设计

|                            |              |
|----------------------------|--------------|
| <b>第七章 命令窗口的程序设计 .....</b> | <b>(111)</b> |
| 7.1 使用命令窗口 .....           | (112)        |
| 7.2 命令窗口的典型用法 .....        | (112)        |
| 7.3 一个典型的编辑过程 .....        | (130)        |
| 7.4 在命令窗口中使用函数 .....       | (140)        |
| 7.5 命令窗口中的综合考虑 .....       | (156)        |
| <b>第八章 良好的传统编程技术 .....</b> | <b>(159)</b> |
| 8.1 编程的配置 .....            | (159)        |
| 8.2 文件、过程和用户定义函数 .....     | (168)        |
| 8.3 程序控制结构 .....           | (181)        |
| 8.4 屏幕和 READ .....         | (190)        |
| 8.5 变量和数组 .....            | (209)        |
| 8.6 更加基本的命令和函数 .....       | (226)        |
| 8.7 调试 .....               | (236)        |
| 8.8 建立短程序 .....            | (247)        |
| 8.9 样本程序 .....             | (247)        |
| 8.10 建立用户库 .....           | (252)        |

## 第三部分 应用程序的开发

|                          |              |
|--------------------------|--------------|
| <b>第九章:组织应用程序 .....</b>  | <b>(257)</b> |
| 9.1 应用程序开发周期 .....       | (257)        |
| 9.2 应用程序的组成部分 .....      | (261)        |
| <b>第十章:分析和设计 .....</b>   | <b>(263)</b> |
| 10.1 分析一个项目 .....        | (263)        |
| 10.2 表的设计 .....          | (266)        |
| 10.3 应用程序设计 .....        | (274)        |
| 10.4 用户接口设计 .....        | (283)        |
| <b>第十一章:用户接口元素 .....</b> | <b>(287)</b> |
| 11.1 用户接口目标 .....        | (287)        |

|             |                  |              |
|-------------|------------------|--------------|
| 11.2        | 颜色               | (288)        |
| 11.3        | 文本和字体            | (295)        |
| 11.4        | 窗口               | (296)        |
| 11.5        | 菜单               | (319)        |
| 11.6        | 屏幕元素             | (319)        |
| 11.7        | 用户消息             | (351)        |
| 11.8        | 键盘元素             | (354)        |
| <b>第十二章</b> | <b>项目管理器</b>     | <b>(363)</b> |
| 12.1        | 使用项目管理器          | (364)        |
| 12.2        | 建立应用程序           | (369)        |
| <b>第十三章</b> | <b>屏幕生成器</b>     | <b>(373)</b> |
| 13.1        | 几点初步的说明          | (373)        |
| 13.2        | 屏幕生成器与 READ 命令   | (375)        |
| 13.3        | READ 命令          | (376)        |
| 13.4        | 建立一个基本屏幕         | (379)        |
| 13.5        | 屏幕生成器            | (426)        |
| <b>第十四章</b> | <b>菜单生成器</b>     | <b>(431)</b> |
| 14.1        | 菜单的基本组成部分        | (431)        |
| 14.2        | 使用菜单生成器          | (437)        |
| <b>第十五章</b> | <b>编程元素</b>      | <b>(443)</b> |
| 15.1        | READ 命令的使用基础     | (444)        |
| 15.2        | 各编程元素之间的协调       | (445)        |
| 15.3        | 事件驱动编程           | (457)        |
| 15.4        | 集成 SQL           | (463)        |
| <b>第十六章</b> | <b>特殊的数据元素</b>   | <b>(469)</b> |
| 16.1        | 使用备注型字段          | (471)        |
| 16.2        | OLE 与 GENERAL 字段 | (474)        |
| 16.3        | DDE              | (477)        |
| <b>第十七章</b> | <b>多用户系统和网络</b>  | <b>(481)</b> |
| 17.1        | 多用户环境            | (481)        |
| 17.2        | 锁定问题             | (482)        |
| 17.3        | 网络的设置            | (494)        |

## 第四部分 建立完整的应用程序

|                               |              |
|-------------------------------|--------------|
| <b>第十八章 通用的应用程序组成部分 .....</b> | <b>(501)</b> |
| 18.1 应用程序的启动.....             | (501)        |
| 18.2 应用程序测试.....              | (507)        |
| 18.3 错误捕获与恢复.....             | (513)        |
| 18.4 应用程序文档.....              | (513)        |
| 18.5 应用程序的分发.....             | (528)        |
| <b>第十九章 辅助的应用程序组成部分 .....</b> | <b>(533)</b> |
| 19.1 磁盘上的软件.....              | (533)        |
| 19.2 数据字典.....                | (534)        |
| 19.3 系统代码.....                | (541)        |
| 19.4 安全性.....                 | (543)        |
| 19.5 输出系统.....                | (553)        |
| <b>附录 附带磁盘说明 .....</b>        | <b>(559)</b> |

## 第一部分 使用 FoxPro 系统菜单



同其它主要 Windows 下的数据管理系统一样, FoxPro 也提供了一个交互式用户接口。对大多数人而言, 均要使用标准的 FoxPro for Windows 菜单系统, 如图 1.1 所示。

标准菜单符合所有 Microsoft Windows 产品的惯例。如果读者已经对 Windows 使用很熟悉了, 就会发现操作 FoxPro for Windows 菜单很容易。

**注意:** 这个菜单能通过编程来修改。因此, 标准菜单不包括那些能建立和连接到系统菜单, 或通过 Run 菜单项可调用的程序、屏幕、报表等。

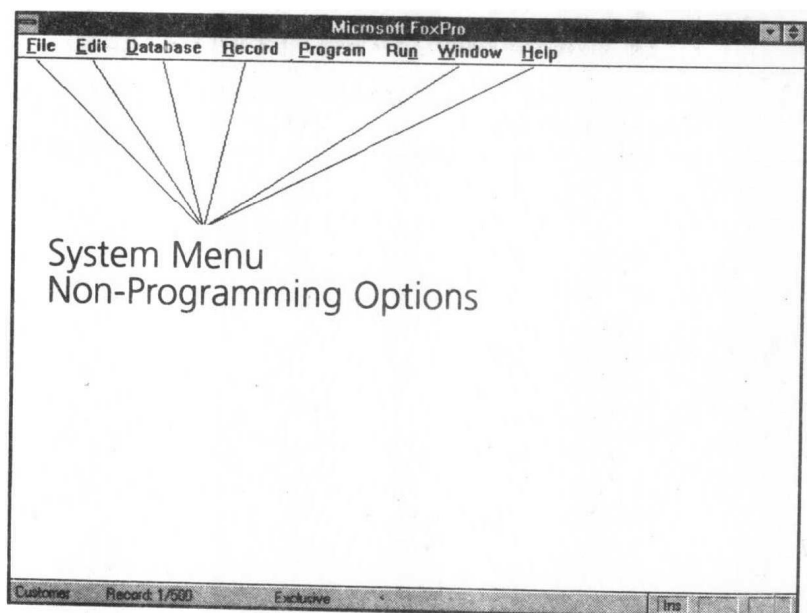


图 1.1 Microsoft FoxPro for Windows 的系统菜单

在学习菜单使用时, 会发现它们有点奇怪的混淆, 比如报表放于数据库选项下, 删除放于记录选项下。事实上尽管数据库这个词它还在系统菜单中, 但它已与数据文件区分开了, 主要原因是, 菜单选项是旧的 DOS FoxPro 菜单和 Windows 方法的混合产物。

然而, 一个菜单系统用户友好的方法仍需要数据库概念的基本知识, 同时对用户而言, 这些概念是被屏蔽的。系统在许多时候认为, 用户知道在干什么。

本部分按数据库管理的概念和实践, 强调了菜单系统的选项, 而不是详述它们的用法, 这在 FoxPro for Windows 手册中很容易找到。它把 FoxPro for Windows 菜单系统看作是数据库管理的方便访问点, 即使读者很快去从事 FoxPro for Windows 的编程(第二到第四部分), 菜单系统仍然非常有用。大多数人, 包括从事编程的人, 都使用系统菜单、命令窗口和编程来进行许多数据库工作。

在继续讲述之前, 先解释几个术语和约定。

表 1.1 公共术语和约定

|        |                                                                                                                     |
|--------|---------------------------------------------------------------------------------------------------------------------|
| 命令动词   | 命令(Command)是 Xbase 语言最基本的组成部分,它常被表示为动词,例如 DO、COUNT 等,大多数系统菜单由命令动词组成,本书中,命令动词以大写字母表示。                                |
| 命令子句   | 许多命令动词带有修改和增加动词动作的附加子句,例如,COUNT..FOR、BROWSE..FIELDS 等。在系统菜单中,许多命令子句在菜单弹出式窗口中以按钮选项出现。注意,学习命令也意味着学习子句,子句数量大约是命令数量的五倍。 |
| 函数和操作符 | 函数和操作符是语言的辅助组成部分,函数的形式与数学函数大体相似,它们提供获取信息、执行子过程等的方式,如 DATE()和 SEEK()。操作符并不显眼,但同等重要,包括=、+、-、*、>和<>。                   |
| 菜单项    | 系统菜单选项(可在菜单中进行选择)的第一个字母以大写字母表示。                                                                                     |
| 样本代码   | 本书中,使用标准打字机格式字形表示用户想要打印的东西。所有样本代码用以下方式表示:SET ALTERNATE ON。                                                          |

# 第一章 文件组织

## 1.1 表结构窗口:创建 DBF

首先要介绍的是数据文件。

进入 FoxPro for Windows 要做的第一件工作,要干的第一件事是创建数据文件并把一些数据放入其中。在许多方面,数据库管理的根本点在于把信息放入某类结构化的格式中,使之能访问和操作。即使在深入到编程和应用开发时,也不可能远离数据文件,它是象 FoxPro for Windows 这类的程序所关心的东西。否则,可以用 Basic 或 C++ 来工作。在 FoxPro for Windows 中,正如其它所有的 Xbase 产品一样,数据文件被称为 DBF(数据库文件,这样称呼的原因是 DOS 文件的扩展名是 DBF)。建立一个 DBF 很容易,用鼠标或键盘激活系统菜单,选取 File New 项 (ALT-F,N),然后选取 TABLE/DBF 项,按下 New 按钮,操作如图 1.1 所示。

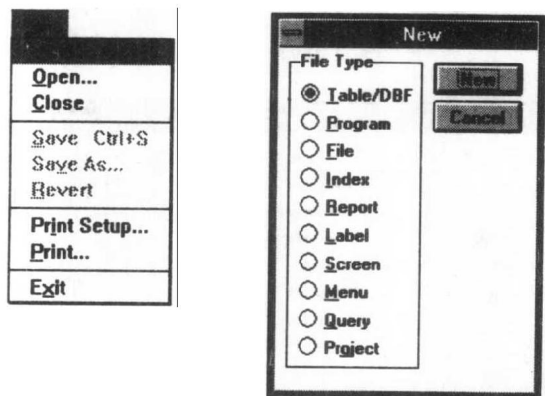


图 1.1 文件菜单

向上弹出表结构窗口(图 1.3),便可开始工作。此时,用户需要指定文件名,定义字段名、类型、长度等等。

掌握一些关于 DBF 的技术细节对读者很有帮助。事实上,如果想使东西可见的话,心理上的“图片”是有用的,这类似于图 1.2。

DBF 有开始(标题)、中间(数据记录)和结尾(记录终止处)三个部分。标题包括了有关 DBF 文件的信息,其中有文件中的记录数目、记录中的字段名及其属性。接下来是按行可见的记录。例如,如果文件包括了一个邮寄表,那么每个记录就是一个邮寄地址的信息。FoxPro for Windows 在理论上可以在一个 DBF 中处理 1 亿个记录。

| FILE HEADER |         |         |         |         |
|-------------|---------|---------|---------|---------|
| FIELD 1     | FIELD 2 | FIELD 3 | FIELD 4 | FIELD 5 |
|             |         |         |         |         |
|             |         |         |         |         |
|             |         |         |         |         |
|             |         |         |         |         |
|             |         |         |         |         |
|             |         |         |         |         |
|             |         |         |         |         |

图 1.2 典型的数据库结构

描写文件、记录和字段的传统方式是用行(记录)和列(字段)表示的表格,磁盘上的文件实际看上去不象这种图表,但它确实具有相同的顺序。在许多其它数据库系统,尤其是结构化查询语言(SQL)中,文件本身就称作表。在 FoxPro for Windows 中,Microsoft 正在把文件的使用变为表的使用。更加清楚的是,磁盘上存储的数据仍称为文件,表指的是文件的逻辑结构。

### 1.1.1 表结构窗口的操作

事实是,图 1.3 所示的表结构窗口是 FoxPro for Windows 最重要的菜单屏幕。在这里,用户可以定义表的字段,然后把表结构存到磁盘上的 DBF 文件中。

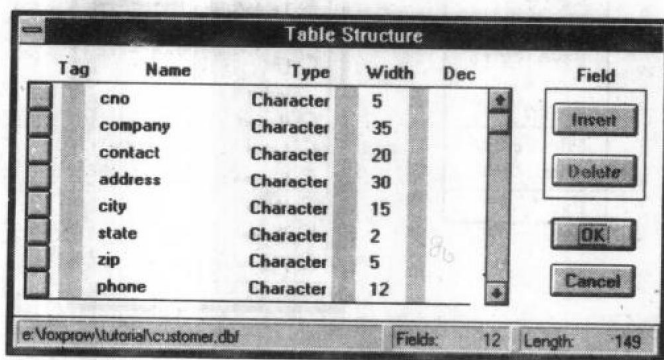


图 1.3 表结构窗口

### 1.1.2 命名数据文件

如果数据库系统中只有三个文件,则怎么称呼它们都没有关系。另一方面,如果这是一个较大的系统,包括不同种类的文件(比方说 100 到 200 个),系统由不同的人来开发和维护,那么文件命名就不再是无关紧要的事了。而且,由于 FoxPro for Windows 在 DOS 中只有一个根目录,所有文件名必须受到文件名最多八个字符、扩展名最多三个字符的限制。要是

能为文件取名为 Cedar Hills accounting payables 就好了,但不能这样做。可以取名为 CHACCPAY.DBF。每个人都在构造可记忆的文件名,因此,有必要制订一些约定,并遵守这些约定。

例如,用户可能还要使用 DOS 命令来拷贝文件,所以要设计文件名来支持这个过程。文件扩展名(.DBF,.CDX 和.FPT)很有帮助,但也可引入项目、类型等来区分文件。例如,文件名的前两个或前三个字母来表示项目或模块。如果项目用于实验室设施,则所有文件名可以 LS 打头;如果所有文件都属于通用分类模块,则可用 GL 打头。

```
GLMAIN.DBF |  
GLUPDATE.DBF | GLACCTS.DBF  
COPY GL*.DBF F:\GL\DBF
```

这是第一次建议读者根据项目大小和参加人数建立约定,以后还会有许多这类的建议。

### 1.1.3 表设计的准备工作

必须明白,在大型专业应用程序中,可能整个项目时间的三分之一要用于表设计和输入字段信息的准备工作中。这是因为,每一个表和字段表示了数据库系统中数据类型的定义。而且,在得到指定的类型信息之前,通常必须进行访问、分析、设计和讨论。这个过程很大程度上不属于本书的范围,但不可低估其作用。做好准备工作,就增加了项目成功的可能性。

表字段是 FoxPro for Windows 数据库系统的最根本组成部分,决不能轻视。不仅字段的选择重要,而且有关命名、类型和长度的决策对整个数据库管理和软件开发过程都有影响,许多人往往先把尝试性的表结构(字段设计)画在纸上,这并不好。现在,程序中定义字段的方法,如在 FoxPro for Windows 中,采用了很多期望值。缺省是很有用的,就象同时观察许多文件结构的一种方式,或是注释字段名和其它字段信息的一种方法。这就是数据字典的范围,它将在第四章中讨论。

不管开发什么样的表和字段的设计方法,在设计变为真实文件之前都需认真思考。这并不是说修改一个 FoxPro for Windows 文件很困难,至少,在文件被数据填满前,这是件很容易的事情。但是在本章有关关系数据管理系统的讨论中,读者将会看到,使用 FoxPro for Windows 的功能来组织信息是有很多事情要做的,做好它需要智慧、勤奋、经验和一些运气。

### 1.1.4 创建数据字段

什么是字段?在使用一个包含客户邮寄表的例子中,期望能得到什么?当然是客户名,或许是以名字和姓分开的方式列出的,这样,可按名字排序和查找。或许“客户”是一家公司,那么就需要表中包括公司名称。接下来,我们需要客户的地址,它又包括象街道、城市、邮区号等不同部分,这些东西对于定义客户很有帮助。在数据表中,每一项将是一个字段。

一个表的记录具有同样的字段,它们被看作是列。在 FoxPro for Windows 中,一个表中最多可有 255 个字段。每列(字段)中的数据必须类型相同,内容近似。举例来说,如果要定义一个含有姓氏的字段,那么每个记录中的该字段都包含某人的姓氏,或者为空。按照应该装入的数据类型,赋予每个字段一个名字和几个属性。

### 1.1.5 字段名

开始输入字段到表中时,这大概是在决定了应包括什么字段之后,需要给每个字段一个名字、数据类型和宽度。在一段时间内,即使是在单一项目中,也将建立许多字段。不要生成太多的字段,实践证明,建立字段的命名约定是比较好的。

- 使字段易于记忆,即字段名能帮助记忆字段的内容。由于字符名限制在 10 个字符以内,所以这并不总是件容易的事情。
- 使用逻辑的和一致的缩写形式。假定想用一个字段来表示售货的总价格,则可称呼该字段为 TCGS 或 TOTALCOGS。第一种方法采用了每个字的首字母,第二种方法在缩写的同时试图保留一些,每种方法都能运转。与所有的编程约定一样,没有哪种方式能做所有的事情。某些方式可能在某些环境下显得更好。约定的关键不是用户如何去多做,而是前后一致地去做。
- 在不同的表中,相同的内容使用相同的字段名。如果需要在两个不同的表中使用同一个姓氏字段,那么可在两个表中统一将该字段称为 LASTNAME。这曾经会导致问题出现,但现在随着别名和别名操作符(如 INV. LASTNAME)的一致使用,重叠字段名就变得较为安全了。这样做的好处包括易于进行相似文件间数据传送和建立文件间的关系。更重要的是,它使得事情变得清楚,因为字段表示的内容实际是相同的。

### 1.1.6 数据类型

数据分类是数据管理首先要做的事情。使用户感到惊讶的是理解  $5+6=11$  很困难,他们可能会很难理解为什么  $5+6=11$ ,但理解“5”+“6”=“56”就不困难了。原因就在于 5 和“5”是不同的,前者是个数字,可用数字做数学运算,而后者是字符,就象字母 b,你不能进行运算,“5”+“6”式中的 + 号表示连接操作,即把字符连在一起。

数据类型很大程度上是对计算机限制的适应。计算机是如此地严格,以致于必须精确地告诉它工作信息的种类。人们已经接受了限制条件,并制订了超出它们的有用范围。数据类型成了数据库“模型”和取得真实世界信息的一个要素,它正变得越来越重要,因为人们已把声音、图象等引进了数据库中。字段的数据类型指明了我们能用来干什么,例如,在本书后续部分中,读者将遇到许多仅用于特定数据类型的函数。

在 FoxPro for Windows 中,共有七种数据/字段类型,如表 1.1 所示。

实际上,大多数字段是字符型的。字符并不仅限于字母,空格、符号和数字都可成为字符数据。下一个最常用的字段是简单的数值型,它几乎用于所有的业务数字中,这两种数据类型的使用约占全部字段的 85%。

另外五种字段类型是专用类型。从纯数据库管理的角度来看,日期型字段是最奇怪的类型之一,它展示给用户的是象 01/01/93 这样的东西,但内部却存储为数字。在使用日期字段或变量时,必须记住,它不是由字符构成的。

备注型字段和通用型字段则自成一统。备注型字段曾是 Xbase 语言的自由形式的正文字段。备注型字段的数据在一个分离的文件中存储,其扩展名为 .FPT,其长度和结构不受任何限制,包括可用的磁盘空间。这使它可方便地放置任何非标准字段类型信息。借助于