

理想的
办公套件

中文 Office 95 系列丛书之四

中文 Visual FoxPro 3.0 使用开发指南

彭江平 罗宜晶 主 编
晏峻峰 黄玉成 副主编



P. H. UESTC

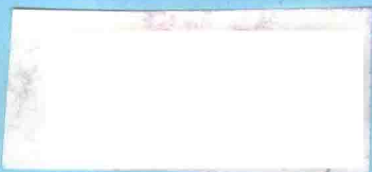
电子科技大学出版社

理想的办公套件——中文 Office 95 系列丛书之四

73.67
C771
73.87
C771

图说号

Handwritten notes in the top left corner, including the characters "中文" and some illegible scribbles.

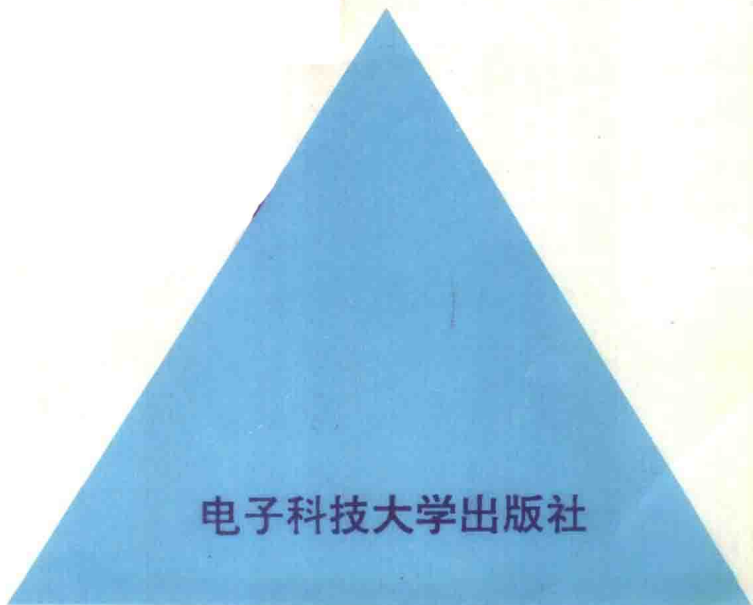
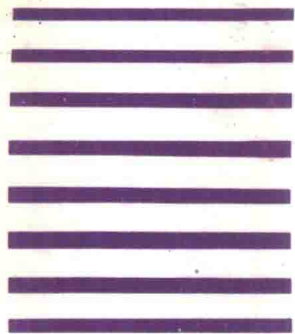


58

中文

Visual FoxPro 3.0 使用开发指南

彭江平 罗宜晶 主 编
晏峻峰 黄玉成 副主编



电子科技大学出版社

中文 Visual FoxPro 3.0 使用开发指南

理想的办公套件——中文 Office 95

系列丛书之四

彭江平 罗宜晶 主 编

晏峻峰 黄玉成 副主编

•
电子科技大学出版社出版

(成都建设北路二段四号) 邮编 610054

成都青羊福利东方彩印厂

新华书店经销

•
开本 787×1092 1/16 印张 23.5 字数 567 千字
版本 1997年2月第1版 印次 1997年2月第1次印刷
印数 1—5000 册
中国标准书号 ISBN 7-81043-539-6/TP · 210
定价: 28.80 元

前 言

中文 Visual FoxPro 3.0 是一个动态关系数据库开发环境, 无论对于新的编程人员, 还是有经验的开发者, 它都是非常友好且功能强大的。

Visual FoxPro 3.0 既具有 Visual 系列的功能强大、直观易学、面向对象等优点, 又兼具 Windows 和 FoxPro 的长处, 提供了向导、设计器和生成器等工具, 使数据库的管理工作变得更容易, 提供的 SQL 和 ODBC 技术又使之能更方便地与其他数据库产品及软件进行移植。

本书内容分为上下两篇。

上篇为基础篇。主要介绍中文 Visual FoxPro 3.0 的基础知识, 库、表操作, 报表及标签的制作与使用, 菜单设计, 实用程序设计初步等内容。还涉及了关系数据库与视图的有关知识, 特别是比较详细地介绍了面向对象的有关概念。

下篇为提高篇。介绍了最新的关系数据库的有关内容与技术, 主要包括结构查询语言 (SQL)、事件驱动应用程序设计、项目管理器、开放数据库互连 (ODBC)、客户/服务器 (Client/Server) 功能及 Visual FoxPro 3.0 与其他软件的数据交换。

最后为附录, 包括 Visual FoxPro 3.0 的主要命令集、函数集等内容, 读者可从中查询自己关心的有关问题。

随着中文 Visual FoxPro 3.0 的推广应用, 它必将成为最流行的桌面数据库产品。

参加本书编写的人员有: 彭江平、罗宜晶、晏峻峰、黄玉成。其中, 第五、六、七、十、十一、十三章及附录二、三、四的内容由彭江平编写, 第一、八、九章及附录一、五、六、七的内容由罗宜晶编写, 第二、三、四章的内容由晏峻峰编写, 第十二章的内容由黄玉成编写。

由于作者水平有限, 再加上时间仓促, 书中难免存在错误和不足之处, 敬请读者批评指正。

编 者

1996 年 11 月

1995.11.11

目 录

上篇 基础篇

第一章 中文 Visual FoxPro 3.0 基础知识

1.1	进一步完善的数据库概念	3
1.2	中文 Visual FoxPro 功能简介	6
1.2.1	快速开发应用程序	6
1.2.2	功能上的深入强化	7
1.2.3	建立客户/服务器解决方案	8
1.2.4	应用程序的交互使用	10
1.2.5	早期版本的升级	10
1.3	中文 Visual FoxPro 的安装、启动与退出	11
1.3.1	中文 Visual FoxPro 的安装	11
1.3.2	中文 Visual FoxPro 的启动与退出	16
1.4	中文 Visual FoxPro 3.0 的系统界面	18
1.4.1	中文 Visual FoxPro 3.0 的窗口组成和基本操作	18
1.4.2	菜单系统	20
1.4.3	工具栏	28
1.4.4	Visual FoxPro 的设计器 (Designer)、向导 (Wizard)、生成器 (Builder)	29
1.4.5	命令 (Command) 窗口	34
1.4.6	文件类型	34
1.5	Visual FoxPro 帮助的获取	36
1.6	本章小结	36

第二章 库表操作

2.1	表操作	39
2.1.1	表的创建	39
2.1.2	表结构的修改	46
2.1.3	表数据的输入	47
2.1.4	表的浏览与编辑	49
2.1.5	表的索引	50
2.1.6	查询数据	52
2.2	库操作	60
2.2.1	数据库的创建、打开、关闭	60

2.2.2 数据库设计器	61
2.3 本章小结	66

第三章 报表及标签的制作与使用

3.1 报表	67
3.1.1 制作报表的准备工作	67
3.1.2 制作报表的方法	67
3.2 标签	90
3.2.1 标签向导	90
3.2.2 标签设计器	93
3.3 标签与报表的打印	94
3.4 本章小结	95

第四章 菜单设计

4.1 菜单设计器	96
4.1.1 “菜单设计器”的启动	96
4.1.2 “菜单设计器”的使用方法	97
4.2 使用系统主菜单上的“查看”选项	101
4.3 使用系统主菜单上的“菜单”选项	102
4.4 菜单的使用方法	104
4.5 本章小结	104

第五章 工作环境、工具栏的定制与键盘宏的定义

5.1 工作环境的定制	105
5.1.1 利用选项对话框设置环境	105
5.1.2 设置的显示及保存	106
5.1.3 各类选项的设置	106
5.2 “工具栏”的定制	118
5.2.1 “工具栏”对话框	118
5.2.2 “工具栏”的显示与隐藏	119
5.2.3 “工具栏”的定制	119
5.2.4 自定义“工具栏”	120
5.3 键盘宏的定义	120
5.3.1 宏的生成	121

5.3.2 编辑宏	122
5.3.3 宏集合	123
5.4 本章小结	123

第六章 实用程序设计初步

6.1 程序设计的基本概念与过程化程序设计简介	124
6.1.1 过程化程序设计与面向对象程序设计	124
6.1.2 过程化程序设计的基本操作	125
6.2 COMMAND窗口及常用命令	126
6.2.1 基本命令	126
6.2.2 与文件有关的命令	127
6.2.3 使用数据命令	128
6.3 过程化程序设计初步	130
6.3.1 与用户对话: 输入/输出命令	130
6.3.2 程序控制	136
6.3.3 过程及变量的作用范围	142
6.3.4 自定义函数	143
6.4 其他	144
6.4.1 运行已有的XBASE程序	144
6.4.2 编译应用程序成EXE文件	144
6.5 本章小结	145

第七章 关系数据库与视图

7.1 数据库的基本概念	146
7.1.1 基本概念	146
7.1.2 数据库系统的特点	147
7.1.3 数据库的分类	147
7.2 数据规范化	148
7.2.1 多对一关系	148
7.2.2 多对多关系	151
7.2.3 其他类型的关系	152
7.3 实例	153
7.3.1 建立关系数据库ReadBook1.dbc	153
7.3.2 应用关系数据库	154
7.4 使用视图更新数据	154
7.4.1 本地视图的创建	154
7.4.2 远程数据的查看与更新	156
7.5 视图的定制	159
7.6 多表查询与更新	162
7.6.1 查询多个表和视图	162

7.6.2	在视图中使用多个本地表	164
7.6.3	在视图中使用多个远程表	165
7.6.4	在视图中合并本地和远程数据	165
7.7	创建视图的SQL命令简介	165
7.7.1	单数据表的本地视图	165
7.7.2	多数据表的本地视图	165
7.7.3	视图的使用命令	166
7.8	本章小结	166

第八章 Visual FoxPro 面向对象程序设计基础

8.1	面向对象技术的形成与发展	167
8.2	面向对象的有关基本概念	168
8.3	面向对象技术在数据库系统中的应用	171
8.3.1	数据库发展的历史	171
8.3.2	面向对象数据库系统(OODB)的特征	172
8.3.3	OODB的优势	172
8.3.4	OODB应用开发工具	173
8.3.5	OODB的现状与前景	174
8.4	面向对象的程序设计方法在VFP 3.0中的应用	174
8.4.1	VFP 3.0的类的概念	175
8.4.2	VFP 3.0中类的层次	176
8.4.3	类的创建与处理	178
8.4.4	对象处理	178
8.4.5	VFP中的事件	180
8.4.6	面向对象的程序设计步骤	181
8.5	VFP程序设计示例	181
8.6	本章小结	183

下篇 提高篇

第九章 结构查询语言(SQL)

9.1	关于SQL查询	187
9.2	SQL SELECT	188
9.3	SQL SELECT命令	189

9.3.1	选项[ALL/DISTINCT]	190
9.3.2	选项[<alias, >]<select item>[AS <column name>]	190
9.3.3	FROM <table>[<alias>] [, <table>[<alias>] ...]	190
9.3.4	选项INTO (destination)	191
9.3.5	选项TO FILE (filename) [ADDITIVE]	191
9.3.6	选项WHERE	191
9.3.7	其他选项	192
9.3.8	SELECT命令使用实例	192
9.3.9	其他SQL命令	193
9.4	本章小结	193

第十章 事件驱动应用程序设计

10.1	Visual FoxPro的对象的层次与事件机制	194
10.1.1	对象的层次	194
10.1.2	对象的引用方法	195
10.1.3	设置对象的属性	196
10.1.4	事件与方法	196
10.2	本章用到的数据说明	196
10.2.1	数据库READBOOK.DBC	196
10.2.2	自由表READ1.DBF	197
10.2.3	其他	197
10.3	利用VFP的可视化工具与事件机制进行事件驱动应用程序设计 ..	197
10.3.1	表单、文本框与标签的应用——单数据表维护表单	197
10.3.2	编辑框的应用	204
10.3.3	复选框对象的应用	206
10.3.4	命令组与选项组应用	208
10.3.5	组合框应用示例	210
10.3.6	列表框应用示例	213
10.3.7	微调控制器的应用	215
10.3.8	计时器应用示例	217
10.3.9	图像对象应用示例	218
10.3.10	OLE对象的应用	219
10.3.11	表格控件的应用	221
10.3.12	页框控件的应用	231
10.3.13	应用程序表单的设计过程	233
10.4	应用可视类库进行事件驱动程序设计	234
10.4.1	利用VFP的可视化设计工具与已有的可视类库	234
10.4.2	可视化地建立自己的可视类库	237
10.4.3	用编程的方式动态地建立自定义类	238
10.4.4	用编程的方式应用可视类库	241
10.5	VFP的事件循环机制	242
10.5.1	FoxPro 2.X的事件循环机制的实现	242

10.5.2	VFP的事件循环机制	242
10.5.3	应用示例	243
10.6	本章小结	244

第十一章 项目管理器

11.1	关于“项目管理器”	245
11.1.1	“项目管理器”初览	245
11.1.2	“项目管理器”的使用	246
11.1.3	建立项目或应用程序	251
11.2	建立一个示例应用程序	253
11.3	本章小结	254

第十二章 Visual FoxPro 3.0 与其他应用程序之间的信息共享

12.1	基本概念	255
12.2	交互式地实现VFP与其他应用程序之间的信息共享	256
12.2.1	添加OLE对象到表单	256
12.2.2	在表中添加OLE对象	256
12.2.3	显示数据表中的通用字段中的OLE对象	257
12.3	自动化地实现VFP与其他应用程序之间的信息共享	258
12.3.1	OLE对象示例	258
12.3.2	创建对象	259
12.3.3	操作OLE对象	261
12.4	本章小结	265

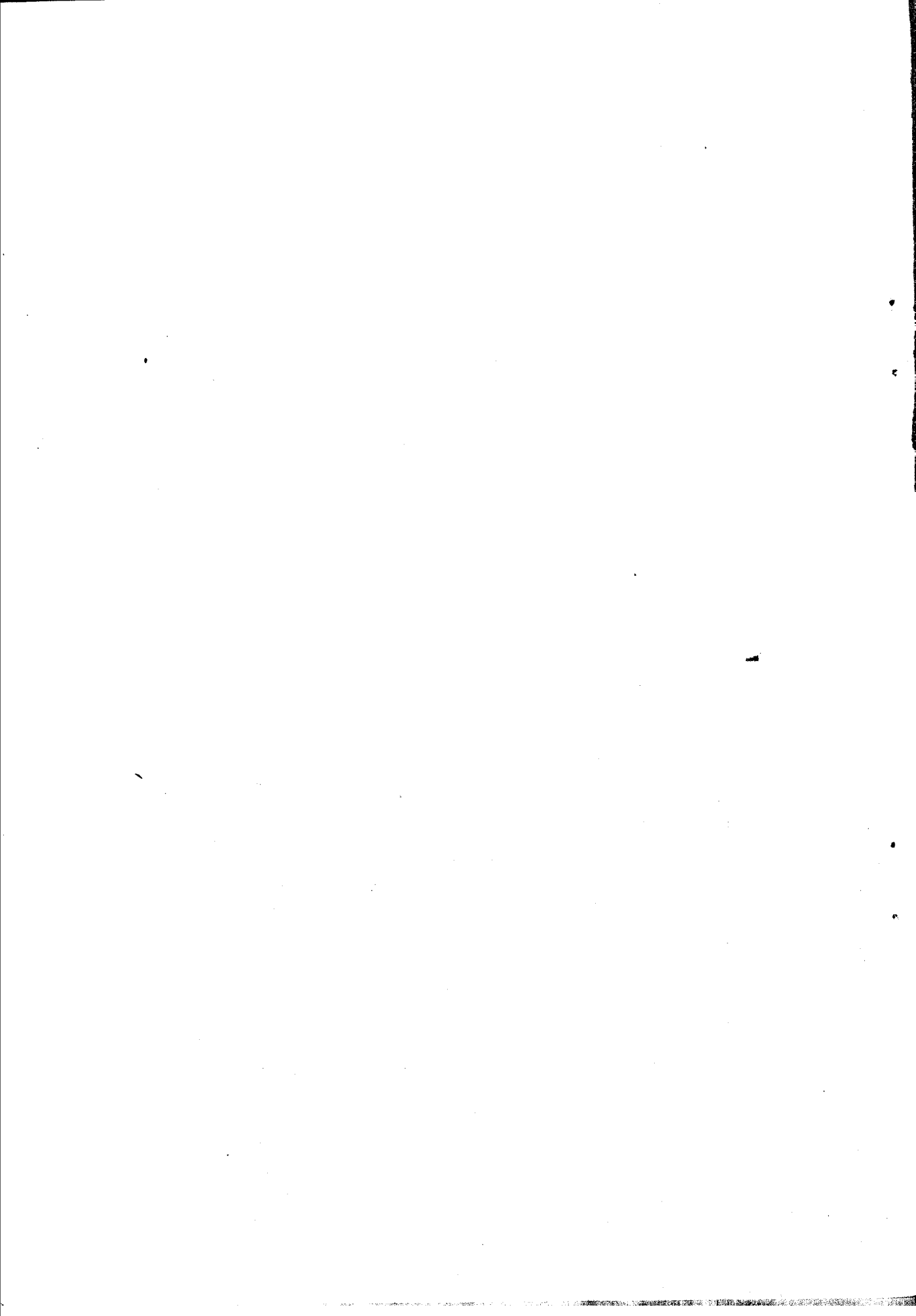
第十三章 开放数据库互连及客户/服务器功能

13.1	ODBC结构概述	266
13.1.1	数据库的接口	266
13.1.2	ODBC的体系结构	270
13.1.3	ODBC的系统要求	270
13.1.4	在应用程序中使用ODBC	271
13.2	ODBC驱动程序	272
13.2.1	什么是ODBC驱动程序	272
13.2.2	ODBC驱动程序的结构	272
13.2.3	ODBC驱动程序的分类	275
13.2.4	驱动程序的选择	277
13.3	ODBC数据源的管理	278
13.3.1	如何安装和使用ODBC管理器	278
13.3.2	如何配置日志登记项	279

13.3.3	如何通过ODBC管理器创建数据源	280
13.4	创建ODBC的链接	281
13.4.1	建立与数据源的链接	281
13.4.2	如何与数据源脱离链接	284
13.4.3	设置、获取链接选项	284
13.5	客户/服务器(Client/Server)功能	285
13.5.1	SQL Pass Through技术	286
13.5.2	视图技术	287
13.6	本章小结	292
附录一	有关Visual FoxPro 3.0的语言规则	293
一、	符号表示和约定	293
二、	操作数据	293
三、	命名规则	298
附录二	Visual FoxPro 3.0主要命令集	301
附录三	Visual FoxPro 3.0的主要函数集	334
附录四	Visual FoxPro 3.0主要系统内存变量集	347
附录五	FoxPro 2.6和Visual FoxPro 3.0之间的差异	349
一、	语言差异	349
二、	表格和数据库差异	351
三、	界面差异	351
四、	功能差异	354
附录六	运行已有的FoxPro 2.6源代码	356
附录七	Visual FoxPro 3.0标准术语英汉对照表	357

上篇 基础篇

- ☐ 中文 Visual FoxPro 3.0 基础知识
- ☐ 库表操作
- ☐ 报表及标签的制作与使用
- ☐ 菜单设计
- ☐ 工作环境、工具栏的定制与键盘宏的定义
- ☐ 实用程序设计初步
- ☐ 关系数据库与视图
- ☐ Visual FoxPro 面向对象程序设计基础



第一章 中文 Visual FoxPro 3.0 基础知识

- 进一步完善的数据库概念
- 中文Visual FoxPro功能简介
- 中文Visual FoxPro的安装、启动与退出
- 中文Visual FoxPro 3.0的系统界面
- Visual FoxPro帮助的使用
- 本章小结

目前,在我国广泛使用的数据库管理系统产品主要是XBASE类产品,如dBASE系列产品、FoxBASE等,这些数据库系统简单实用、容易掌握。但是,其缺点如用户界面不友好、开发效率低、执行速度慢等却给人们进一步的应用带来了许多麻烦,更何况,它们天生就不是“真正”的数据库,并不支持完整的关系数据库的概念。相比之下,某些“真正”的数据库产品,如Oracle、Ingres等,又都需要较高级的硬件设备,学习难度也比前者高得多。但是毋庸置疑,它们对关系数据库各种概念的支持又要比前者完善得多。这样看来,在使用的方便性和概念上的完备性之间似乎存在着一道不可逾越的鸿沟,用户只能在这两者之间作出取舍。真的是这样吗? Visual FoxPro 3.0(简称VFP)的推出为我们提供了另外一种选择。

英文VFP 3.0是1995年8月推出的,汉化版本发表于1996年5月。如今的VFP不仅继承了FoxPro的优良传统,而且还充分利用了许多最新的计算机理论和技术,如可视化编程和面向对象方法。借助于工具栏、对象和可视控件,VFP可以自动完成界面的设计并执行各种任务。VFP还在完善关系数据库概念上前进了一大步,加入了Table、事务、触发器等内容,可视化工具和加强的链接性使用户可以与大多数后台数据库的客户/服务器应用程序链接,使32位的ODBC驱动程序能够集成来自各个系统的数据,包括客户/服务器数据库的数据、本地数据及其他应用程序的数据。VFP可直观地、创造性地建立应用程序,可以从工具栏上访问OLE 控件和定制控件,从而利用Microsoft Office 和其他应用程序的各种功能。由上可知,与FoxPro 2.5/2.6 相比,VFP无论是在数据库理论,还是在具体实现上都有了显著的提高。

1.1 进一步完善的数据库概念

Visual FoxPro 3.0对XBASE(数据库)的内涵进行了很大的改进和充实。除了把DataBASE(数据库)和Table(表)的概念分清以外,还引入了视图。同时,触发器的使用和关联的加强也增强了VFP 对数据库中数据完整性和一致性的控制能力。此外,事务管理和ODBC 驱动程序的加入又大大方便了网络应用系统的设计。至此, Visual FoxPro 3.0将

XBASE数据库提升到完整的关系数据库范畴，是一个真正的DataBASE，完全可以被看作是一个后端的数据库引擎(Engine)。

正如XBASE系列用户早已熟悉的，在FoxPro 2.6 以前版本的XBASE系列数据库产品中，数据库就是一个二维的表结构，数据库和表这两个概念没有什么分别，两者实际上所指的是同一个东西，而VFP 则对其进行了严格的区分。

在VFP中，表是指同一类记录的集合。它是描述实体集的，所以是一个描述实体集的所有符号的集合。表一般已经作为一个单独的文件存在磁盘上。例如，所有的客户记录组成一个客户数据表CUSTOMER.DBF。VFP的表在概念上等同于XBASE系列产品中的所谓数据库文件(DBF文件)，表文件的文件名同样沿用以.DBF为后缀的默认规则。

而数据库则是指一个或多个表的集合。数据库用来描述一个或多个实体集及实体集与实体集间的关联，所以，数据库是一个描述多个实体集及其关联的所有符号的集合。例如，客户表、雇员表、产品表、订单表等组成一个销售公司的经营数据库。在VFP中，数据库文件以.DBC为后缀。数据库概念的结构如图1.1所示。

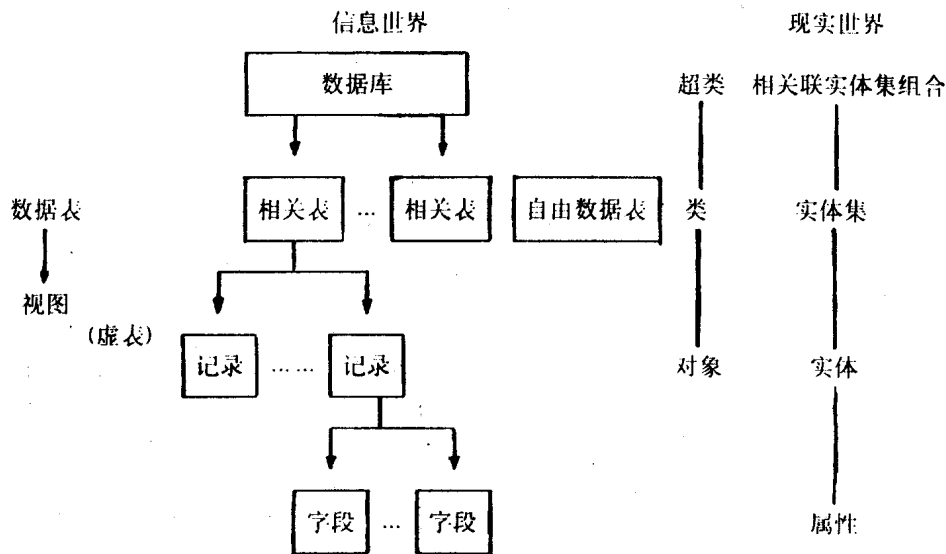


图1.1 数据库概念的结构图

数据库是由表组成的，而一个表又不能同时由多个数据库共同拥有。因此，当表被包含到某个数据库中时，表文件中将被记下标识，表示已被某数据库使用，这样，其他数据库就不能再包含该数据表。隶属于数据库的表称为相关表(Associated Table)，相关表既可与数据库配套使用，也可脱离数据库的约束，作为一个独立表使用。没有包含到任何数据库中的表为自由表，既可以单独使用，也可以被某个数据库所包含。由于XBASE 数据库产品建立的DBF文件，都被VFP当成自由表看待，因此自由表的提出，也继承和保护了用户已有的数据库应用成果。

视图是从一个或几个表中导出的表。视图也称为虚表，它本身不独立存储在数据库中，而只在数据库中保存其定义。但在用户看来，视图和数据表是一样的。在VFP中，视图又分为本地视图和远程视图两种。

VFP 把原来的表文件升级为数据库，又引入了视图，并自己完成对相关表的管理，就可以使用户在开发信息管理系统时把真正需要的数据存入数据库，而对于某些情况，只要在需要的时候产生视图即可。并且，又因为数据库把逻辑上相关的表直接组织到一起，也

保证了整个数据库中数据逻辑的完整性和一致性。

使用以前的XBASE产品开发管理系统时,如何保证数据库中数据的完整性和一致性一直是一个难以解决的问题。为了保证数据的正确性,不得不编写大量的Valid函数和其他检查过程。但是,由于这种检查大部分只限于数据的输入阶段,程序和数据库中数据的联系毕竟不十分紧密,因而常常有种种违背数据的完整性和一致性原则的操作发生。实际上,要想彻底解决数据的完整性和一致性问题,仅靠这些是远远不够的。VFP通过引入触发器、硬关联和引用一致性设置,基本上解决了这一问题。

所谓触发器(Trigger),其实是主动数据库系统中所涉及到的概念。主动数据库系统,就是不但提供了传统数据库功能,而且还可以自动识别应用程序定义情况,并自动触发应用程序定义动作的数据库系统。触发器就是用来实现这种数据库系统主动性的一种机制,它又可以根据用户所关心的情况分成相应的几类。在关系数据库系统中,如Oracle,触发器一般可分为Insert Trigger、Delete Trigger和Update Trigger等三种类型,分别对应于数据库中所进行的插入、删除和更新操作。在触发器中,事先由用户定义好要执行的动作,然后,当系统探测到有这三种操作即将进行时,就会执行这些动作,完成对数据库中数据满足完整性条件的判断,以及其他可能的动作,如显示一个字符串等。触发器还可以监控对指定数据对象的访问,找出对保护数据的非法访问,并执行有关动作。

在Visual FoxPro 3.0中,引进了关系数据库的三类完整性:实体完整性、引用完整性和触发器。前两类是数据库本身自我约束的完整性规则,由系统自动支持。

1. 实体完整性规则

这条规则要求表中记录的主键值不能是空值。如果出现空值,那么,主键值就起不了唯一标识记录的作用。

2. 引用完整性规则(Reference Integrity)

简单地说,设置数据库中两个以上表之间对关键字的一种记录维护关系,如同步修订外部关键字或同步进行删除,这样即可确保表之间的记录的完整性。引用完整性规则适用于VFP所提供的硬关联功能。

3. 用户自定义的触发器

这是针对某一具体数据库的约束条件,由应用环境决定。它反映某一具体应用所涉及的数据必须满足的语义要求。VFP利用Stored Procedure存储处理程序和对触发器的支持提供了定义和检验这类完整性的机制,以使用统一的、系统的方法处理它们,而不要由某个应用程序担负这一功能。

VFP还为网络应用系统的开发提供了初露端倪的事务管理。当在网络上运行信息系统时,必须采用一定的机制解决数据访问和存储的冲突。例如,同一项记录同时经不同工作站进行修改,若这里有两个以上的人已经将该项记录进行修改而没有回存,那么,当回存时,就有可能造成数据访问冲突,产生“写丢失”或者“脏读”现象,即两人都写,而后写者覆盖了前者所写的内容;或一人读,一人写,在前者刚刚读完一个数据之后,后者却把它给改成了另一内容。这样,如果前者再读第二次的话,就会产生不一致的现象。为解决这一问题,XBASE系统开发人员常常需要专门开发解决数据维护冲突的程序,这就加重了程序设计人员的工作负担。

而VFP强化表与数据库的功能,允许用户对表设置网络操作时缓冲区的处理环境,可自动对修改欲回存的记录或表进行检查、扫描、分析、锁定、更新、释放等。另外,借助Buffering可以无需使用变量维护记录,先对Buffering维护记录,再用Commit或Revert

对记录进行真正的回存或还原。这样，VFP 便像后端关系数据库一样，让用户在开发网络应用系统中，变得轻松方便，更能确保信息的一致性与完整性。

以上只是从理论上探讨了VFP 对关系数据库概念的完善和支持，而这种理论上的完善在实际设计系统时则表现为整个数据库具有严谨的结构，数据逻辑的安全性、完整性和一致性得到了有力的保证。

1.2 中文Visual FoxPro功能简介

作为新一代数据库管理系统(DBMS)，VFP 给用户提供了前所未有的强大功能——更快的速度、更强的能力和更卓越的灵活性，并引起了XBASE的又一次革命。Visual FoxPro所提出的新的对象风格和事件模式，将帮助用户比以往任何时候都要快速、简捷地建立和修改应用程序。

1.2.1 快速开发应用程序

在Visual FoxPro中提供的向导、生成器、工具栏以及设计器等一系列开发工具，使用户可以快速创建应用程序。Visual FoxPro对象和事件模式的使用，又可以快速开发、原型化和实现真正的无模式应用程序。

1. 利用向导和生成器

当希望快速产生设计结果时，向导的使用可以让用户更高效地完成预定的任务。所谓向导是Microsoft提出的一种通过向用户提出简单问题的方式来创建复杂文件的实用程序，它集完美的视觉感受和齐全的功能于一体，用户可以在屏幕上进行绘制、设置指定的字段等等，然后，根据自己的需要对其进行相应的调整和处理。在Visual FoxPro中，系统提供了大量的实用向导，并逐步执行用户确认的指令，以帮助其在Visual FoxPro中胜任一般工作。例如，在创建表的过程中，“表向导”提供了必要的指导，参考“表单向导”可以帮助用户创建表单，而“查询向导”则展示了如何创建标准查询。有关向导的其他详细信息，可以在联机帮助中参考“使用向导”主题进一步了解。

作为一种可视的工具，生成器可以帮助用户按照预定的详细说明进行控制设计。例如，“列表框生成器”就是一个选项卡式的对话框，用于设计表单上的列表框，在这种生成器中还可以设置列表框的一般属性。有关Visual FoxPro 生成器的详尽信息，可以在联机帮助中搜索“使用生成器”主题。

2. 使用快捷工具栏

遵循Microsoft标准应用程序的样式，Visual FoxPro还提供了便捷的工具栏。工具栏包含用户经常使用的对象以及要完成普通任务的按钮。要完成指定的任务或使用特殊的对象时，应该选择适当的按钮。

工具栏可以进行定制，它们主要来源于Visual FoxPro，也可以在自己设计的应用程序中创建自定义工具栏。有关工具栏的详细信息，请在帮助中搜索“定制工具栏”主题。

3. 通过设计器开发应用程序组件

利用Visual FoxPro设计器进行开发工作，则用户得到的对应用程序的控制要比使用向导或生成器为多。在Visual FoxPro中，其设计器提供了美观的图形界面，这样可以直