



中等职业教育国家规划教材
全国中等职业教育教材审定委员会审定

数据库应用基础

—— FoxPro for Windows

(计算机及应用专业)

主编 魏茂林



高等教育出版社

中等职业教育国家规划教材

全国中等职业教育教材

数据库应用基础

—FoxPro for Windows

(计算机及应用专业)

主 编 魏茂林
责任主审 宋方敏
审 稿 柏子阳 朱树春



高等教育出版社

内 容 提 要

本书是中等职业教育国家规划教材,根据教育部 2001 年颁布的数据库应用基础教学基本要求编写,同时参考了有关行业的职业技能鉴定规范及中级技术工人等级考核标准。本书系统地介绍了 FoxPro 2.5b for Windows 基础知识、数据表的基本操作和程序设计方法。全书共分 10 章,第 1~6 章分菜单和命令两种方式详细介绍了创建 FoxPro 数据表,记录的显示、修改、增删、排序、索引、查询、统计汇总、多重数据表的操作及 SELECT 查询命令与 RQBE 关系查询等;第 7~10 章介绍了程序的基本结构、子程序、过程文件、函数、输入输出设计、用户界面设计及简单应用程序的编写。每章后配有大量例题、思考与练习、上机实习等内容。本书选材力求新颖、浅显易懂,突出学生基本技能的培养,同时有利于教师组织教学。

本书可作为中等职业学校计算机及应用专业及相关专业教材,也可作为数据库培训教材和读者自学用书。

图书在版编目(CIP)数据

数据库应用基础 FoxPro for Windows/魏茂林主编.

北京:高等教育出版社,2002.7(2005 重印)

中等职业教育国家规划教材

ISBN 7-04-010887-9

I. 数… II. 魏… III. 关系数据库-数据库管理系统,FoxPro-专业学校-教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2002)第 037262 号

数据库应用基础——FoxPro for Windows

魏茂林 主编

出版发行 高等教育出版社

社 址 北京市西城区德外大街 4 号

邮政编码 100011

总 机 010-58581000

经 销 北京蓝色畅想图书发行有限公司

印 刷 北京印刷集团有限责任公司印刷二厂

购书热线 010-58581118

免费咨询 800-810-0598

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

网上订购 <http://www.landaco.com>

<http://www.landaco.com.cn>

开 本 787×1092 1/16

印 张 17.5

字 数 410 000

版 次 2002 年 7 月第 1 版

印 次 2005 年 12 月第 10 次印刷

定 价 21.30 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 10887-00

中等职业教育国家规划教材出版说明

为了贯彻《中共中央国务院关于深化教育改革全面推进素质教育的决定》精神，落实《面向 21 世纪教育振兴行动计划》中提出的职业教育课程改革和教材建设规划，根据教育部关于《中等职业教育国家规划教材申报、立项及管理意见》（教职成[2001]1 号）的精神，我们组织力量对实现中等职业教育培养目标和保证基本教学规格起保障作用的德育课程、文化基础课程、专业技术基础课程和 80 个重点建设专业主干课程的教材进行了规划和编写，从 2001 年秋季开学起，国家规划教材将陆续提供给各类中等职业学校选用。

国家规划教材是根据教育部最新颁布的德育课程、文化基础课程、专业技术基础课程和 80 个重点建设专业主干课程的教学大纲（课程教学基本要求）编写，并经全国中等职业教育教材审定委员会审定。新教材全面贯彻素质教育思想，从社会发展对高素质劳动者和中初级专门人才需要的实际出发，注重对学生的创新精神和实践能力的培养。新教材在理论体系、组织结构和阐述方法等方面均作了一些新的尝试。新教材实行一纲多本，努力为教材选用提供比较和选择，满足不同学制、不同专业和不同办学条件的教学需要。

希望各地、各部门积极推广和选用国家规划教材，并在使用过程中，注意总结经验，及时提出修改意见和建议，使之不断完善和提高。

教育部职业教育与成人教育司

二〇〇一年十月

前 言

本书是根据教育部 2001 年颁布的中等职业学校计算机及应用专业教学指导方案中的数据库应用基础教学基本要求编写的，被列为中等职业教育国家规划教材。

本书以初学者为对象，系统地介绍了 FoxPro 2.5b for Windows 数据表的创建，记录的显示、修改、增删、排序、索引、查询、统计汇总、多重数据表的操作、SELECT 查询命令与 RQBE 关系查询等数据表的基本操作和程序的基本结构、子程序、过程文件、函数、输入输出设计、用户界面设计及简单应用程序的编写。本书数据表操作部分给出了菜单和命令两种操作方式，菜单方式直观、方便；命令方式操作快捷，同时为下一步学习编写应用程序打下基础。

本书内容的组织与编排比较符合学生的认知规律，简单了解数据表的基本知识之后，首先进行数据表的创建、数据记录的基本操作，然后进行基本程序结构的设计学习，体现了由简单到复杂的认知规律。本书把内部函数的讲解安排到程序设计部分，这样处理避免了学生一开始学习新课程就接触大量枯燥的函数，给学生很大的思想压力，不利于后面内容的学习。

本书大量的例题是围绕学生熟悉的典型事例进行编排的。教师处理教材例题时，可以采取任务驱动方式，先提出要解决的任务，启发学生寻找解决问题的方法，获取解决问题的途径。

本书在编写时注意启发、引导学生去思考问题，章节中随例题安排了一些思考题，是对例题求解的进一步引申，或要求学生注意的问题，达到举一反三的目的。本书围绕章节内容，给出了大量的练习和上机实习，加深对数据库知识的理解和应用，同时又加强了对学生技能的培养。

为了便于读者继续学习数据库的其他课程，例如，Visual FoxPro 7.0、Access2000 等，本书将数据库称为数据表，请读者在阅读本书时注意。

本书的教学参考课时数安排如下：

章	教 学 内 容	讲授	上机	小计	说 明
1	FoxPro 基础知识	6	2	8	数据表基本操作部分
2	数据表的基本操作	6	4	10	
3	数据表的排序与查询	4	4	8	
4	数据统计与汇总	2	2	4	
5	多重数据表操作	3	2	5	
6	RQBE 关系查询与 SELECT 查询操作	3	3	6	
7	FoxPro 程序设计基础	5	4	9	程序设计部分
8	子程序、过程文件和函数	5	3	8	
9	基本输入输出设计	3	3	6	
10	用户界面设计	3	3	6	
	机 动		2	2	
	合 计	40	30	72	

本书由青岛市教育局职业教育教研室魏茂林担任主编，第1~6章、第8章、第9章由魏茂林编写，第7章由济南市第六职业中专王勇进编写，第10章由青岛食品学校姜全生编写。本书经全国中等职业教育教材审定委员会审定，由南京大学计算机系副主任宋方敏担任责任主审，由南京大学柏子阳、朱树春老师审稿，高等教育出版社另聘请了北京市宣武职教中心学校马开颜老师审稿。

本书在编写过程中得到了省、市教育主管部门的指导，并提出了许多宝贵意见，在此一并表示感谢。

由于作者水平有限，书中难免存在一些不足，真诚欢迎各学校在使用过程中提出宝贵意见。

编 者
2002年1月

目 录

第 1 章 FoxPro 基础知识	1
1.1 数据库的基本知识	1
1.2 FoxPro 关系数据库简介	2
1.3 FoxPro 的启动与退出	2
1.4 常量、变量和表达式	3
1.5 FoxPro 命令的使用	12
本章小结	13
思考与练习	14
上机实习	16
第 2 章 数据表的基本操作	18
2.1 数据库的设计	18
2.2 建立数据表的结构	19
2.3 打开和关闭数据表	22
2.4 显示与修改表结构	23
2.5 输入与显示记录	26
2.6 浏览与修改记录	34
2.7 指针移动与删除记录	41
2.8 数据表文件的复制操作	47
2.9 数据表与数组之间的数据交换	54
本章小结	57
思考与练习	58
上机实习	62
第 3 章 数据表的排序与查询	65
3.1 数据表的排序	65
3.2 顺序查找记录	67
3.3 过滤查找记录	69
3.4 数据表的索引	70
3.5 快速查找记录	79
本章小结	81
思考与练习	82
上机实习	85
第 4 章 数据统计与汇总	87
4.1 统计记录个数	87

4.2 数值求和与求平均值	89
4.3 统计计算	92
4.4 数据分类汇总	93
本章小结	96
思考与练习	96
上机实习	98
第 5 章 多重数据表操作	99
5.1 工作区的使用	99
5.2 数据表的连接	103
5.3 数据表的关联	105
5.4 数据表的更新	111
本章小结	112
思考与练习	113
上机实习	116
第 6 章 RQBE 关系查询与 SELECT 查询操作	118
6.1 RQBE 关系查询	118
6.2 SELECT 查询操作	130
本章小结	138
思考与练习	139
上机实习	140
第 7 章 FoxPro 程序设计基础	143
7.1 命令文件的建立与运行	143
7.2 常用的交互语句	147
7.3 顺序结构程序设计	148
7.4 分支结构程序设计	151
7.5 循环结构程序设计	157
本章小结	161
思考与练习	162
上机实习	167
第 8 章 子程序、过程文件和函数	171
8.1 子程序	171
8.2 过程文件	173
8.3 程序之间的数据传递	176
8.4 内存变量的属性	178
8.5 常用函数和自定义函数	182
本章小结	193
思考与练习	194
上机实习	197

第 9 章 基本输入输出设计	201
9.1 SET 命令设置	201
9.2 格式化的输入输出	204
9.3 屏幕格式文件	211
9.4 颜色的设置	213
9.5 窗口设计	215
9.6 输出格式设计	219
本章小结	230
思考与练习	231
上机实习	233
第 10 章 用户界面设计	236
10.1 常见的菜单设计	236
10.2 屏幕生成器的使用	243
10.3 常见控件的设计	245
10.4 应用程序举例	249
10.5 程序的编译	255
本章小结	257
思考与练习	257
上机实习	257
附录 常用 FoxPro 命令一览表	261

第 1 章 FoxPro 基础知识

随着信息时代的到来,数据库技术得到了广泛的应用和发展,无论是像 Oracle、Sybase、Informix、SQL Server 这些大型的数据库管理系统,还是像 PowerBuilder、FoxPro 这些微机上常用的数据库开发系统,在计算机数据管理方面都发挥出了巨大的作用。

1.1 数据库的基本知识

1.1.1 数据库

数据就是记载客观事物的状态、特征、特性的符号。数据无处不在,如考试成绩、身高、收入等,除了数字以外,还有字符、图形、图像和声音等计算机能处理和存储的数据。

数据库(DataBase)就是按一定的组织结构存储在计算机内可共享的相关数据的集合。一个数据库可以由一个或多个数据表组成。数据库中的数据是按照一定的数据模型进行组织、描述和存储的,与用户的应用无关,具有较小的数据冗余、较高的数据独立性和易扩展性,并能为用户所共享。

1.1.2 数据库管理系统

数据库管理系统简称 DBMS(DataBase Management System),是用来建立、存取、管理和维护数据库的软件系统。它是数据库系统的核心软件,数据库系统的不同用户对数据库的建立,数据的编辑、修改、检索、统计汇总,打印报表及数据库的并发控制等操作,都是由数据库管理系统来完成的。它是用户与数据库之间的接口。如学校学生成绩管理系统,包括成绩的录入、查询、统计及报表打印等功能。

1.1.3 数据库系统

数据库系统就是指计算机系统中引用了数据库技术后的完整系统,主要由计算机硬件(主机、键盘、显示器、打印机等)、软件(操作系统、数据库管理系统、开发工具等)、数据库、用户(数据库管理员、终端用户等)四个主要部分组成。

1.1.4 关系数据库

各个数据对象以及它们之间存在的相互关系的描述,称为数据模型。数据模型的构造方法决定了数据库的设计方法及数据库中数据之间的联系方式。常见的数据模型有关系模型、层次模型和网状模型,根据这三种数据模型建立的数据库分别是关系数据库、层次数据库和网状数据库。

关系数据库是把每一个实体看成一个二维表,用二维表来组织和存储数据,它是目前应用

广泛的一种数据模型。每个二维表又称为关系。关系型数据库管理系统是支持关系模型的数据数据库管理系统。

在关系数据库管理系统中，关系数据库通过一个二维表来表示数据之间的联系。表中的每一列称为一个字段，表的第一行是字段名，从第二行开始每行是一条记录。例如，我们可以按照表 1-1 所示的学生基本情况表来建立一个关系数据库，表中的学号、姓名、性别等称为字段，每个字段都有惟一的一个名字，并且每个字段中所有的数据都必须是同一种数据类型；从第二行开始每行是一条记录，一个数据库中存储多条记录。需要特别说明的是，在本教材中将数据库称之为数据表。

表 1-1 学生基本情况表

学号	姓名	性别	出生日期	专业	入学成绩	团员	简历
000101	周讯阳	男	85-02-12	电子商务	450.0	是	
000121	王大鹏	男	85-09-01	电子商务	435.5	否	
000205	李晓莉	女	84-12-24	电器维修	378.0	是	
...	

1.2 FoxPro 关系数据库简介

FoxPro 2.5b for Windows 是 Microsoft 公司于 1993 年开发的关系数据库管理系统。它能和 FoxBASE、FoxPro 以前的版本兼容，但在功能上有新的突破和创新。

FoxPro 采用了 Windows 风格的图形界面，支持鼠标操作，通过菜单窗口可以实现对数据库的基本操作；提供了功能强大的软件开发工具，通过菜单生成器、屏幕生成器和报表生成器等工具，可以提高应用程序开发者的工作效率；具有支持对象的链接与嵌入技术(OLE)，可共享采用 Windows 应用程序提供的文档、表格、图形、图像、声音等资源。

1.3 FoxPro 的启动与退出

在 Windows 系统中安装好 FoxPro for Windows 后，系统自动建立“FoxPro for Windows”文件夹，它包括 3 个图标：FoxPro for Windows、FoxPro Q&A、FoxPro Readme，第一个图标用来启动 FoxPro，后两个图标是有关 FoxPro 一些帮助内容的文件。

1.3.1 FoxPro 的启动

单击 Windows “开始”按钮，选择“程序”菜单中的“FoxPro for Windows”程序组，单击其中的“FoxPro for Windows”命令，启动 FoxPro，屏幕上显示 FoxPro 的窗口界面，如图 1-1 所示。

FoxPro 窗口组成与其他 Windows 应用软件的窗口组成基本类似，其操作方法也大同小异。但在 FoxPro 里还有一个小窗口——命令窗口，其标题为“Command”。在这个命令窗口中可以输入执行 FoxPro 的操作命令，完成相应的操作。

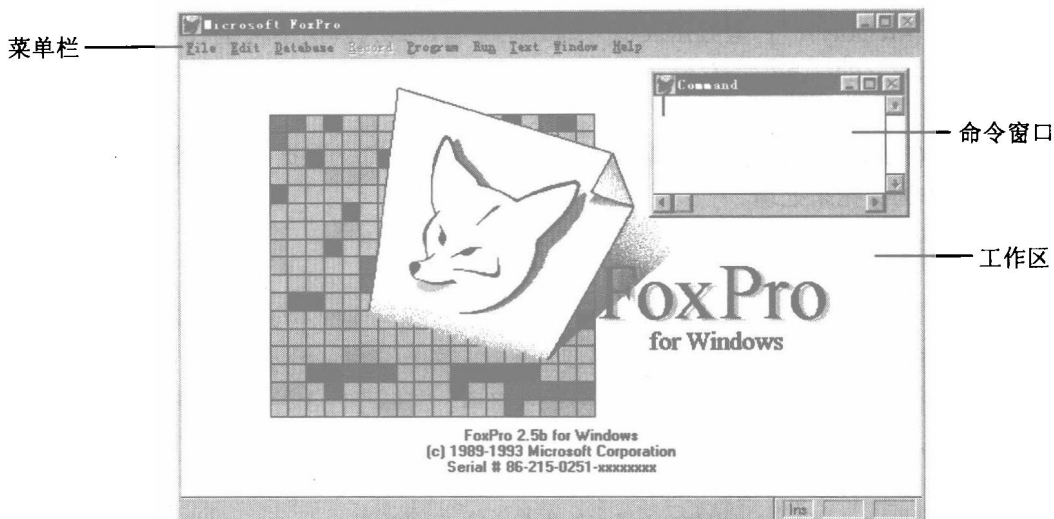


图 1-1 FoxPro 2.5b 界面窗口

1.3.2 FoxPro 的退出

退出 FoxPro 有以下几种方法:

- (1) 用鼠标双击标题栏中最左边的控制按钮;
- (2) 按 Alt+F4 键;
- (3) 在 Command 命令窗口中键入 QUIT 命令, 按回车键执行该命令。

1.4 常量、变量和表达式

FoxPro 中的数据类型有多种, 常用的有字符型、数值型、浮点型、日期型、逻辑型、备注型和通用型。

(1) 字符型(Character): 由英文字母、汉字、数字、空格和各种符号组成的字符串, 其最大长度不能超过 254 个字符(或 127 个汉字)。例如, 学生基本情况表中的“学号”、“姓名”字段均为字符型。

(2) 数值型(Numeric): 数据中包括正负号、数字及小数点。数值型数据的最大长度为 20 位(包括小数点和正负号所占的位数)。例如, 学生基本情况表中的入学成绩是数值型。

(3) 浮点型(Float): 与数值型相同, 包括正负号、数字及小数点。其最大长度也为 20 位, 能精确到小数点后面 15 位。

(4) 日期型(Date): 用花括号括起来表示日期的数据类型, 格式是 yy-mm-dd(年-月-日)或 mm-dd-yy(月-日-年), 年份可以是两位或四位, 可以用 SET 命令或直接通过 Command 菜单的 View 窗口来设置其日期格式及年月日之间的分隔符。如 2001 年 12 月 18 日可以表示为 {01-12-18}、{2001-12-18} 或 {12-18-01} 等。本教材采用系统默认的 yy-mm-dd 日期格式。

(5) 逻辑型(Logical): 它的值只有真和假两种。用 .T. 或 .Y. 表示真, .F. 或 .N. 表示假, 其存储长度为 1 个字节。

(6) 备注型(Memory): 它是数据表特有的数据类型, 是为解决字符型数据最多可容纳 254 个字符的限制而设立的。其数据存储在备注型文件(扩展名为.FPT)中, 存放的字符长度一般只受磁盘空间的限制。

(7) 通用型(General): 它是数据表特有的数据类型, 将外部的数据文件如文档、图表、图像、声音、视频等作为数据来处理。

1.4.1 常量

FoxPro 中的常量与数学中常量的含义相同, 是指其值始终保持不变的量, 又称为常数。FoxPro 中常量分为字符型常量、数值型常量、逻辑型常量和日期型常量 4 种类型。

(1) 字符型常量: 用定界符括起来的字符型字符串。定界符有 3 种: 单引号 “'”、双引号 “”” 和方括号 “[]”, 例如, "Student"、'中国'都是字符型常量。如果某一种定界符本身是字符型常量的组成部分, 则应选择另一种定界符作为该字符串的定界符。例如 "做"四有"人才" 是非法字符型常量, 而[做"四有"人才]、"做'四有'人才"则是合法字符型常量。

(2) 数值型常量: 指一般数学中的常数, 可以是整数或实数。如 5.62, -12.8 等。

(3) 逻辑型常量: 只有真、假两个值。逻辑“真”用.T.、.t.、.Y.或.y.来表示, 逻辑“假”用.F.、.f.、.N.或.n.来表示。

(4) 日期型常量: 用花括号 “{ }” 括起来的日期型数据。例如, {2001-12-18}表示 2001 年 12 月 18 日。

1.4.2 变量

变量是指其值可以变化的量, 如同数学中的 x 、 y 、 z 变量的含义。FoxPro 中的变量分为内存变量和字段变量两种类型。

1. 内存变量

内存变量是用来临时存放数据的存储单元。它独立于数据表, 常用来保存运算的中间结果或用于控制程序的流程, 又称为简单内存变量。内存变量分为系统内存变量和用户自定义内存变量两类。系统内存变量由 FoxPro 系统生成, 用于控制显示、打印输出等, 系统内存变量名由下划线 “_” 开头。

(1) 内存变量的定义。每个内存变量都有一个名字, 内存变量名可由字母、汉字或下划线开头, 由字母、数字或下划线组成, 中间不能有空格, 内存变量名长度不超过 10 个字符。例如, D_2、数量、NL 都是合法的内存变量名。

内存变量有字符型、数值型、日期型、逻辑型和屏幕型 5 种数据类型。屏幕型内存变量不能进行运算, 只能用于保存和恢复屏幕的画面。

内存变量的值决定了内存变量的类型, 改变内存变量的值, 可以改变内存变量的数据类型。

(2) 内存变量的赋值与显示。在 FoxPro 中给内存变量赋值, 常用 “=” 和 STORE 命令。

例 1-1 将 3.14 和 “青岛之夏” 分别赋给内存变量 PI 和 qd, 可以使用下列命令:

```
STORE 3.14 TO PI 或 PI=3.14
```

```
STORE "青岛之夏" TO qd 或 qd="青岛之夏"
```

赋值后内存变量 PI、qd 的数据类型分别为数值型、字符型。

使用 STORE 命令可以同时将一个数值赋给多个内存变量，各变量之间用逗号间隔。

例 1-2 给多个变量赋同一数值，可以使用下列命令：

```
STORE 453.72 TO SL,DA,J8
```

赋值后变量 SL、DA 和 J8 的值都是 453.72。

```
STORE 453.72-200 TO MI
```

赋值时先计算表达式 453.72-200，再将结果 253.72 赋值给变量 MI。

用“?”或“??”命令可以随时把赋值后的内存变量的值显示出来。“?”或“??”命令后面跟要显示的变量、表达式等。如果同时显示多个变量或表达式的值，中间用逗号间隔。

例 1-3 显示例 1-1、例 1-2 给内存变量 PI、qd 和 MI 所赋的值。

```
?PI      &&一行中显示一个变量的值
```

```
3.14
```

```
?PI,qd   &&一行中显示两个变量的值，各变量之间用逗号间隔
```

```
3.14  青岛之夏
```

```
??PI     &&显示一个变量的值
```

```
3.14
```

```
?PI+10   &&显示表达式的值
```

```
13.14
```

?和??两个命令的区别是：?命令在屏幕光标的下一行显示各表达式的值；而??在光标的同一行显示各表达式的值。

要查看内存中全部变量的赋值情况，可以使用 LIST 或 DISPLAY 命令。

格式：LIST|DISPLAY MEMORY [LIKE <通配符>]

功能：显示当前内存中各变量的赋值情况。

说明：

LIST 与 DISPLAY 命令的区别是：LIST 从头至尾显示完毕后结束，而 DISPLAY 命令是分屏显示，每显示一屏后暂停，按任一键继续显示。

通配符可以使用“?”或“*”，一个“?”代替一个字符，一个“*”代替多个字符。

例 1-4 显示例 1-1 和例 1-2 所定义的内存变量。命令操作如下：

```
LIST MEMORY LIKE *
```

屏幕显示：

PI	Pub	N	3.14	(3.14000000)
QD	Pub	C	"青岛之夏"		
SL	Pub	N	453.72	(453.72000000)
DA	Pub	N	453.72	(453.72000000)
J8	Pub	N	453.72	(453.72000000)
MI	Pub	N	253.72	(253.72000000)

如果要显示以 S 开头的所有内存变量，使用下列命令：

```
LIST MEMORY LIKE S*
```

屏幕显示：

SL Pub N 453.72 (453.72000000)

对于屏幕型变量的定义和调用分别用 SAVE SCREEN TO 和 RESTORE SCREEN 命令。

2. 字段变量

字段变量是指数据表中的任意一个字段，它是构成数据表文件最基本的数据单元。在一个数据表中，一个字段可以有很多值，假设一个数据表中有 15 条记录，该表中的每一字段可以有 15 个取值，当用字段名作变量时，它的值随着记录指针的移动而不断变化，所以称它为变量，而且是一种多值变量。如在学生基本情况表中，第一、二两条记录的姓名分别是“周讯阳”和“王大鹏”，当记录指针指向第一条记录时，姓名是“周讯阳”，记录指针指向第二条记录时，姓名是“王大鹏”。这里的“姓名”就是字段变量。字段变量的数据类型是在定义数据表的结构时确定的，具体定义请参看第 2 章的有关内容。

在使用变量时，如果内存变量名与当前数据表文件中的字段变量有相同的名字时，则字段变量优先被使用。如果需要强调内存变量时，则需要是在内存变量名前加上“M.”或“M->”，以避免产生二义性。例如，M.XM，表示 XM 是内存变量。

3. 数组

数组就是多个变量共同使用同一个名称来存储数据，由多个数组元素组成，利用数组的下标来区分每一个数组元素。例如，数组 TX(5) 的数组名为 TX，下标为 5，该数组只有一个下标，称为一维数组，它有 5 个数组元素：TX(1)、TX(2)、TX(3)、TX(4)和 TX(5)。

数组 S(3,4)的数组名为 S，下标分别为 3 和 4，该数组有两个下标，称为二维数组，它有 3 行 4 列共 12 个元素：

S(1,1)、S(1,2)、S(1,3)、S(1,4)

S(2,1)、S(2,2)、S(2,3)、S(2,4)

S(3,1)、S(3,2)、S(3,3)、S(3,4)

数组也是一种内存变量，数组元素的数据类型可以是字符型、数值型、日期型或逻辑型，而且这些类型的数据可同时存在于一个数组中。例如，数组的第一个元素为字符型，第二个元素为数值型等。

(1) 数组的定义。

格式：DIMENSION | DECLARE <数组名 1> (<下标 1>[, <下标 2>])[, <数组名 2> (<下标 3>[, <下标 4>])]…

功能：定义一维或二维数组。

说明：

DIMENSION 和 DECLARE 命令的用法完全相同，数组的命名与简单内存变量命名的规则相同，数组下标可以放在一对圆括号或方括号中。

数组的下标值最小为 1，下标如果有小数，则只截取整数部分。如果省略<下标 2>，则定义的是一维数组，否则定义的是二维数组。

例 1-5 分别定义一个一维和二维数组。操作命令如下：

```
DIMENSION X(5),Y(2,3)
```

上述命令定义了 X 和 Y 两个数组，X 是一个一维数组，其数组元素分别表示为 X(1)、X(2)、X(3)、X(4)和 X(5)；Y 是一个二维数组，其数组元素分别表示为 Y(1,1)、Y(1,2)、Y(1,3)、Y(2,1)、

Y(2,2)和 Y(2,3)。

下面分别是数组 X 和 Y 各元素在内存中的排列情况。

X(1)	X(2)	X(3)	X(4)	X(5)	Y(1,1)	Y(1,2)	Y(1,3)
					Y(2,1)	Y(2,2)	Y(2,3)

数组元素在内存中是按标值的升序来存储的，对于一个二维数组是按行排列的。在表示一个二维数组元素时，既可以用一个下标，也可用两个下标。例如，数组 Y 中的第 4 个元素可以表示为 Y(2,1)，也可表示为 Y(4)。

(2) 数组的赋值。数组定义后各元素的初始数据类型为逻辑型，其值均为.F.。给数组赋值的方法与给简单内存变量赋值的方法相同，可以使用“=”或 STORE 命令。如果只给数组名赋值(不指明下标)，则数组中所有元素均赋同一个数值；如果指明下标，则只给指定下标的数组元素赋值。

例 1-6 下面是给数组元素赋值的操作。

```
DECLARE X(5),Y(2,3)
```

```
LIST MEMORY LIKE *
```

屏幕显示：

```

X          Pub      A
( 1)          L      .F.
( 2)          L      .F.
( 3)          L      .F.
( 4)          L      .F.
( 5)          L      .F.
Y          Pub      A
(1,1)        L      .F.
(1,2)        L      .F.
(1,3)        L      .F.
(2,1)        L      .F.
(2,2)        L      .F.
(2,3)        L      .F.
```

```
X=56
```

```
Y(1,1)=78.12
```

```
Y(1,2)=DATE()
```

```
Y(1,3)="青岛"
```

```
STORE .T. TO Y(2,1),Y(2,2)
```

```
LIST MEMORY LIKE *
```

屏幕显示：

```

X          Pub      A
( 1)          N      56      ( 56.00000000)
```



```

      ( 2)      N   56      ( 56.00000000)
      ( 3)      N   56      ( 56.00000000)
      ( 4)      N   56      ( 56.00000000)
      ( 5)      N   56      ( 56.00000000)
Y      Pub      A
      (1,1)     N   78.12   ( 78.12000000)
      (1,2)     D   01-11-21
      (1,3)     C   "青岛"
      (2,1)     L   .T.
      (2,2)     L   .T.
      (2,3)     L   .F.

```

```

DIMENSION Y(2)
LIST MEMORY LIKE *

```

屏幕显示:

```

Y      Pub      A
      ( 1)      N   78.12   ( 78.12000000)
      ( 2)      D   01-11-21

```

从上例可以看到 X 数组中每个元素的值都为 56, Y 数组各不同元素其数据类型可以不同, 同一数组元素, 前后两次赋值的数据类型也可以不同; 可以将一个数组重新定义, 改变它的维数, 原数组的值保留不变, 增加的数组元素自动赋逻辑假(.F.)值。

1.4.3 内存变量的存取操作

前面已经介绍了使用“=”、STORE 命令对内存变量的赋值及使用 LIST|DISPLAY MEMORY 命令显示内存变量状态的操作, 下面将进一步介绍内存变量的保存、恢复及清除操作。

1. 内存变量的保存

由于内存中存储变量的个数及空间是有限的, 为了提高系统的运行效率, 可以将系统暂时不用的变量保存起来, 然后释放所占用的内存空间, 当系统需要时, 再把它调入内存。

格式: SAVE TO <内存变量文件名> [ALL LIKE|EXCEPT <通配符>]

功能: 将当前内存中的变量存入指定的磁盘文件中, 扩展名默认为.MEM。

说明: 若命令中省略任选项, 则将当前内存中的所有变量(系统变量除外)存入指定的内存变量文件名中; 若选择 ALL LIKE <通配符>选项, 则将与<通配符>匹配的内存变量保存到指定的磁盘文件中; 若选择 ALL EXCEPT <通配符>选项, 则将与<通配符>不匹配的内存变量保存到指定的磁盘文件中。

例 1-7 将例 1-1 定义的内存变量保存到内存变量文件 BL 中。操作命令如下:

```
SAVE TO BL
```

则当前磁盘上建立了 BL.MEM 文件。

例 1-8 将以 P 开头的内存变量存放在 BL1 文件中, 其他变量存放在 BL2 文件中。操作命令如下: