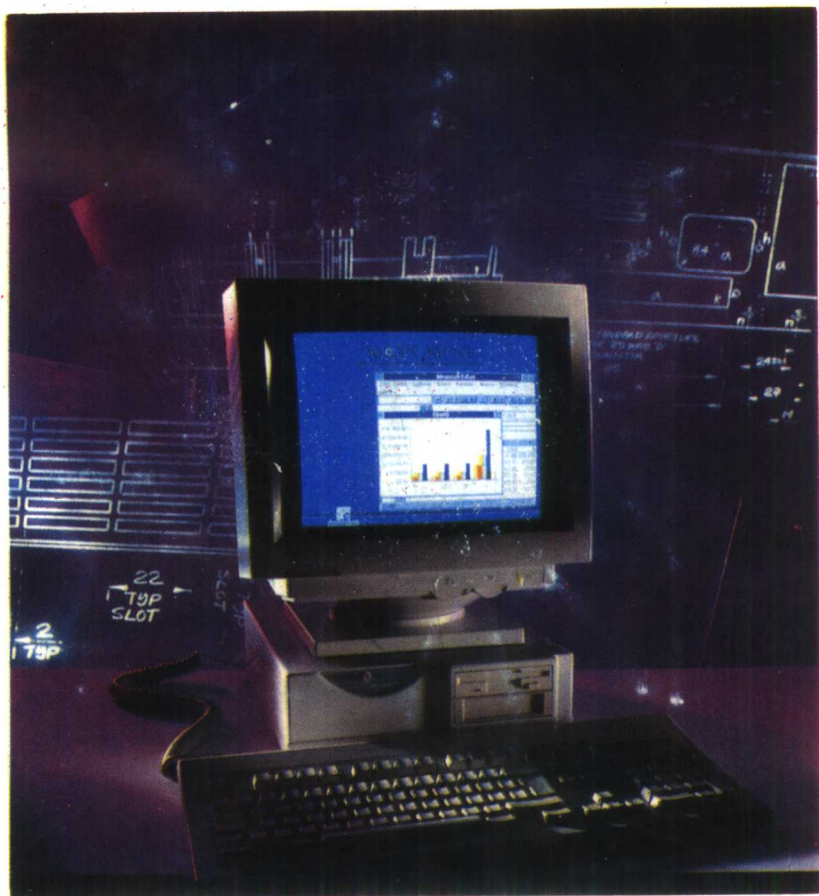


● 计算机基础教育系列教材 ●

汉字 FoxPro 原理及应用

陈福盈 王云宜 主编



中南工业大学出版社

汉字 FoxPro 原理及应用

陈福盈 王云宜 编著

中南工业大学出版社

汉字印刷原理及应用

陈福盛 王云宜 编著

湖南工业大学出版社

中南工业大学出版社出版发行
核工业中南二三〇研究所印刷厂印装
新华书店总店北京发行所经销

*

开本:787×1092 1/16 印张:16.25 字数:408千字

1995年10月第1版 1996年9月1次印刷

印数:0001—4000

*

ISBN 7-81020-795-4/TP·058

定价:18.00元

本书如有印装质量问题,请直接与生产厂家联系解决

厂址:湖南长沙

邮编:410011

前 言

FoxPro 是美国 Fox Software 公司 1991 年推出的最新关系数据库系统软件,是作为微机关系数据库管理系统的更新换代软件推出的。FoxPro 与目前流行的关系数据库管理软件 dBASE⁺、dBASE IV、FoxBASE⁺ 全兼容,首次在微机数据库中引入了集成环境。它不仅是一个编译型的集成式数据库软件,更重要的在于其卓越的强处理功能和高运行速度,提供的丰富的应用软件开发工具,目前正在成为新的数据库管理系统的标准。

FoxPro 围绕着类似于 Macintosh 的强有力的用户界面而设计的。Macintosh 技术即多媒体处理技术,是集图、文、声等多种类信息处理的一种先进技术,它追求文、图、声、象的综合处理能力以克服传统计算机上单一的信息处理模式。当 1992 年美国 Microsoft 公司兼并 Fox 公司后,Microsoft 公司著名的 Windows 平台功能也加入到 FoxPro 之中。针对 DOS 平台和 Windows 平台,FoxPro 也有两类版本,一类是 FoxPro for DOS;另一类是 FoxPro for Windows。本书是以 DOS 平台为主进行阐述的。

FoxPro 采用了独特的 Rushmore 专利查询技术,使查询速度大为提高,在大型数据库中表现了突出优点;采用了图形用户界面使界面屏幕非常美观和生动多样;多窗口技术和菜单操作,使用户需记忆的操作命令数目减至最低;提供的多种程序开发工具,使应用程序开发变得相对容易;程序可编译成 .EXE 文件可独立在 DOS 环境下运行而不必要求系统软件支持等一系列优点,使得它很快得到普及应用。

为了掌握 FoxPro,对于已经具有 dBASE、FoxBASE 编程经历的人,只需要进一步掌握 FoxPro 操作技术及其提供的工具、更多的函数、命令的使用方法,即可进行应用程序的开发工作。而对于没有 XBASE 编程经历的人,则需要学会关系数据库管理系统的基本原理,诸如从二维关系表建立数据库开始,从数据类型、常量变量、函数和命令的使用、特别是数据库管理系统的操作命令和格式的学习入手,进一步结合 FoxPro 的菜单操作,来逐步学习应用程序设计和开发的必要内容。

本书兼顾了前二者的要求,第 1 章简略叙述了数据库的基本概念,第 2 章介绍了在 FoxPro 中用到的公共基础知识,这两章主要是为没有 XBASE 编程经验的用户而写的。从第 3 章开始系统叙述 FoxPro 的基本操作和逐章进行重点内容的阐叙。最后两章则以程序设计技术串通全部章节及应用程序开发,使读者能独立开发、具体应用。

全书按照“保证基础、精选内容、从浅入深、以利教学”的原则编写。全书共 17 章和 3 个附录。第 1、2 章由王云宜编写,7 章至 17 章由陈福盈编写,3、6 章由王伟编写,4、5 章由陈锐编写,陈福盈最后修改定稿,全书各示例程序均已上机通过,读者可以模仿实用。

由于作者水平有限,不妥之处在所难免,恳请专家及读者批评指正。

作 者

1995 年 8 月

目 录

第1章 数据库基本概念	(1)
§ 1.1 信息、数据和数据处理	(1)
§ 1.2 数据模型的分类	(2)
§ 1.3 新一代关系数据库管理系统 FoxPro 简介	(3)
思考题	(4)
第2章 FoxPro 基础知识	(5)
§ 2.1 数据类型	(5)
§ 2.2 常量与变量	(6)
§ 2.3 运算符和表达式	(8)
§ 2.4 函数	(10)
§ 2.5 FoxPro 命令及其格式	(12)
§ 2.6 文件类型及其扩展名	(13)
§ 2.7 记录指针、工作区及别名	(15)
习 题	(17)
第3章 FoxPro 基本操作	(18)
§ 3.1 FoxPro 版本、运行环境、安装启动	(18)
§ 3.2 菜单结构及操作	(19)
§ 3.3 窗口操作及对话框	(23)
§ 3.4 使用 COMMAND 命令窗口	(25)
§ 3.5 获得帮助 Help 和退出 FoxPro	(26)
§ 3.6 FoxPro 的主要功能特色	(27)
思考题	(29)
第4章 建立数据库及其维护	(30)
§ 4.1 数据库结构的设计	(30)
§ 4.2 数据库文件的建立	(30)
§ 4.3 数据库文件的编辑	(42)
习 题	(49)
第5章 数据库的排序和索引	(50)
§ 5.1 建立排序文件	(50)
§ 5.2 建立索引文件	(54)
§ 5.3 排序与索引的异同	(62)
习 题	(62)
第6章 数据库的查询	(63)
§ 6.1 对单个记录的查询	(63)
§ 6.2 对多个记录的查询	(67)
§ 6.3 相关案例查询(RQBE)	(68)

习 题	(75)
第 7 章 报表生成器	(76)
§ 7.1 建立一个快速报表	(76)
§ 7.2 使用页格式设计报表	(79)
§ 7.3 Report 菜单的进一步介绍	(83)
习 题	(93)
第 8 章 邮政标签生成器	(94)
§ 8.1 建立邮政标签	(94)
§ 8.2 打印标签	(97)
§ 8.3 修改现存标签文件	(98)
习 题	(98)
第 9 章 屏幕生成器	(99)
§ 9.1 建立屏幕窗口及 Screen 弹出式菜单	(99)
§ 9.2 使用快速屏幕方式建立屏幕文件	(100)
§ 9.3 使用屏幕程序代码文件	(103)
§ 9.4 使用屏幕设计方式建立屏幕文件	(106)
§ 9.5 字段格式设置和数据有效性检查	(109)
§ 9.6 按键的建立	(116)
§ 9.7 向屏幕添加目标的选择项进一步介绍	(118)
习 题	(122)
第 10 章 菜单生成器	(123)
§ 10.1 人事管理菜单程序设计	(123)
§ 10.2 建立人事管理菜单系统	(123)
§ 10.3 使用用户菜单系统	(131)
§ 10.4 使用 FoxAPP 建立菜单系统	(134)
习 题	(140)
第 11 章 FoxPro 的关系功能	(141)
§ 11.1 数据库的工作区及其选择	(141)
§ 11.2 关系数据库的概念	(141)
§ 11.3 如何关联文件	(143)
§ 11.4 建立多文件报表	(150)
习 题	(153)
第 12 章 键盘宏的使用及数组功能	(154)
§ 12.1 定义键盘宏	(154)
§ 12.2 存贮键盘宏	(157)
§ 12.3 恢复键盘宏	(158)
§ 12.4 键盘宏的进一步介绍	(159)
§ 12.5 FoxPro 的数组功能	(160)
习 题	(163)
第 13 章 文件管理器	(164)

§ 13.1	Files 控制板	(164)
§ 13.2	TREE 控制板	(172)
	习 题	(176)
第 14 章	FoxPro 桌面辅助工具	(177)
§ 14.1	FoxPro 的计算器	(177)
§ 14.2	FoxPro 的日历/日记簿	(179)
§ 14.3	FoxPro 的特殊字符	(180)
§ 14.4	FoxPro 的 ASCII 表	(181)
§ 14.5	屏幕文本的拷贝	(182)
§ 14.6	魔板游戏	(182)
	习 题	(182)
第 15 章	文本编辑器	(183)
§ 15.1	建立一个文本或程序文件	(183)
§ 15.2	EDIT 菜单	(183)
§ 15.3	保存和关闭文件	(187)
§ 15.4	定义文本块	(187)
	习 题	(188)
第 16 章	FoxPro 程序设计	(189)
§ 16.1	FoxPro 程序设计的基本方法	(189)
§ 16.2	命令文件的建立与运行	(190)
§ 16.3	控制流程	(191)
§ 16.4	过 程	(195)
§ 16.5	用户自定义函数(功能块)	(197)
§ 16.6	记忆字段处理技术	(197)
§ 16.7	FoxPro 的窗口技术	(201)
§ 16.8	FoxPro 的菜单技术	(204)
§ 16.9	数据输入程序设计	(211)
§ 16.10	打印报表程序设计	(215)
§ 16.11	程序的调试	(220)
§ 16.12	FoxPro 控制参数的设置	(222)
	习 题	(225)
第 17 章	项目管理器	(226)
§ 17.1	建立一个项目文件	(226)
§ 17.2	项目相关基本数据	(230)
§ 17.3	记录项目相关信息	(231)
§ 17.4	使用项目管理器生成 EXE 文件	(232)
	习 题	(234)
附录 I	FoxPro 文件种类和扩展名	(235)
附录 II	FoxPro 命令集	(236)
附录 III	FoxPro 函数	(245)

第 1 章 数据库基本概念

§ 1.1 信息、数据和数据处理

人类社会已进入信息化社会，人们在政治、经济、军事、文化、教育、科学等各项活动中将产生大量的信息。人们使用了各种物理符号和它们有意义的组合来表示信息，这些符号及其组合就是数据。数据有两类：数值型数据和非数值型数据。如职工人数、产量、工资等属数值型，而文字、图象、报表、声音等为非数值型数据。数据是信息的具体表示形式，信息是数据的有意义的表现。信息与数据相互联系，相互依存，是人类社会一种宝贵的软资源。

随着计算机的运算速度和存贮容量迅速提高，使得人们有可能对大量信息进行保存和加工处理，使信息得到充分的利用，从而促进社会生产力的发展。

数据处理经过了手工处理、机械处理、电子计算机数据处理三个阶段。而计算机管理数据又有人工管理、文件系统管理、数据库管理三种方式。本书所要叙述的数据库管理技术，已是目前最广泛应用的数据管理方法之一。数据库管理系统是帮助人们处理大量信息，实现管理科学化和现代化的强有力工具。由于数据库具有数据结构化、最低冗余度、较高的程序与数据独立性，易于扩充、易于编制应用程序等优点，因而数据库是近年来发展最为迅速的计算机软件，数据库技术是数据信息管理技术的最新成果。

所谓数据库，一般地说是数据的集合，形象地说是存储数据的“仓库”，但是不能把数据库仅仅简单理解为存储数据的集合，而应当理解为存储数据并负责用户访问数据的机构，应视为一个系统，即数据库系统。把它比作图书馆更易理解。大家知道，图书馆是存储图书和负责借阅图书的部门，不能简单地与书库等同起来。图书馆若要很好地为读者服务，首先必须建立完善的书卡，书卡的内容常包括：书号，书名，作者名，出版社名，出版时间，内容摘要及其他细节。其次要按照一定的顺序和规则（物理结构）来分别存放图书，列出各类书籍存放的对应关系。最后是规定图书的借还手续，即读者对图书的访问及管理员对读者访问的响应过程。对于数据库来说，也要建立数据模型，使用户可以根据数据模型访问数据库中的数据（如检索、插入，删除和修改），而不必关心数据在数据库中的物理存储位置，就象读者按书卡填写借书单，而不必知道书籍在书库中存放在什么位置一样。数据要有组织地存放在存储设备上，并建立数据模型到物理存储的对应表（称映射）。使数据库管理系统能够按照用户的访问请求，找到被访问数据的存储位置，数据的应用与数据的存放位置和存储结构无关，正象改变图书的存放位置并不影响读者按书卡借书一样，它是数据的独立性的表现。因此数据库系统应包括数据库和数据库管理软件两部分。而软件包括数据描述语言及其翻译程序，数据操纵语言及其编译程序，数据库例行程序等。分别完成数据库的定义、描述、建立、管理、维护以及通讯等功能。

§ 1.2 数据模型的分类

数据库系统中的数据是按一定的规律存放的，这种规律就是数据库模型，数据库的设计方法根据其模型不同，常见有三种数据模型：

(1) 层次模型：层次模型的结构示意图如图 1.1 所示。

从图可见，每个父结点都有若干子结点，而每个子结点只有一个父结点。它能表示两个实体之间的联系，其层次分明，结构清晰，但对多重从属关系则很难描述。这种事物之间的联系，称一对多的联系。

(2) 网络模型：网络模型比层次模型灵活，某个子结点可以和多个父结点联系，以表达总体的相关性，例如有三种不同的设备，分别由四个不同的厂家供货，其示意图如图 1.2 所示。它们之间的关系为多对多的关系。

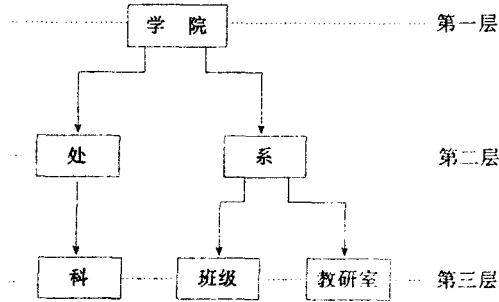


图 1.1 数据库层次模型

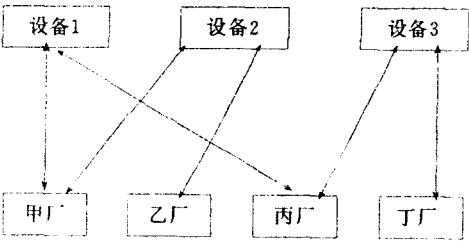


图 1.2 数据的网络模型

(3) 关系模型：这是目前使用得最多的一种模型。关系模型将数据组织成表格的形式，称为二维表，每一个二维表格称为一个关系，其数据可以表示成一对一或一对多的联系。在日常生活中，这种关系模型构成的二维表随时可以见到。如当你走进火车站售票厅时，迎面而来的是两个二维表：票价关系表和里程关系表，它们指明两个车站之间的票价与距离之间的关系。在关系模型中，每一个关系有一个名称，称为关系名，表 1.1 命名了一个职工关系，该关系由 6 个数据或由 6 列组成，表中的每一列称为一个“字段”，而第一个横行是各字段的“型”的集合，构成一个框架。只有给框架的每一个数据项赋值后，得到下面一个“记录”，因而第一行的框架是记录的“型”，其他的行则是各记录的“值”。

表 1.1 职 工 关 系

编 号	姓 名	出生年月	性 别	科 室	职 务
0025	李小明	01/01/70	女	机关	工程师
0038	张 宁	11/12/47	男	第一车间	工 人
⋮	⋮	⋮	⋮	⋮	⋮

为了说明关系型数据库的构成及它能给用户提供的主要功能，我们再建立表 1.2 的工资关系和表 1.3 的科室关系。“工资关系”表示每个职工的工资情况，“科室关系”表示每个科室概要情况，有了这样的数据后，属于以下种类的用户询问都可以得到有效的回答：

表 1.2 工 资 关 系

编 号	月工资	月奖金
0631	105.00	12.00
1053	89.50	10.50
⋮	⋮	⋮

表 1.3 科 室 关 系

科室名	负责人名	职 称
技术科	杨 宁	工程师
生产科	刘建新	高级工程师
⋮	⋮	⋮

(1)各个对象的情况:如某职工的年龄、性别等等。

(2)有关对象间的联系:如某职工的月工资是多少,某科室的负责人是谁等等。

(3)指出具有某种属性的对象:如某科室中女职工,或有工程师职称的人是谁等。

(4)满足某种条件的对象的属性情况:如某职工所在科室的负责人姓名、职称等等。

(5)所有某类对象的统计情况:如各类人员的平均工资是多少?工资低于(或高于)某数额的人数是多少等。

所有这些问题都可以通过上述三个关系之一或通过其中几个或全部,直接或间接地得到答复。

以上按照二维表建立起来的关系数据,在关系数据库管理系统中,将每一个二维表对应的计算机文件称为一个数据库。这些数据本身是集成的、独立的,但可以供给众多不同的用户按自己的需要查询、调用有针对性的内容。当然,作为数据库管理系统,还要提供多种手段供用户建立数据库,随时增、删、修改、编辑已建立数据的内容。还要提供多种查询、索引、制表、打印输出等等操作手段,为用户使用数据和开发应用程序服务。本书的内容就是对如何使用 FoxPro 进行全面的叙述。

§ 1.3 新一代关系数据库管理系统 FoxPro 简介

FoxPro 是由美国 Fox Software 公司 1991 年推出的关系数据库管理系统,用以替代早已广泛流行的 dBASE 和 FoxBASE⁺ 关系数据库管理系统,因而在性能上有其独特的优点。FoxPro 采用了集中环境式多窗口用户界面,采用了多项专利技术及应用程序生成器等多种先进手段,具有运行速度快、用户界面好、开发工具丰富且与 dBASE、FoxBASE⁺ 完全兼容等特点,很快得到了普及应用。

FoxPro 是微机平台上最快的数据库系统,它采用了独特的 RUSHMORE 专利查询技术,复合索引技术,使用 SQL-Select 命令以及利用高档微处理器的最佳性能、自动优化系统,其速度比 dBASE IV 快 16 倍,比 FoxBASE⁺ 快 4 倍。

FoxPro 为了方便用户开发应用程序,提供了许多开发工具,如文本编辑器、屏幕生成器、菜单生成器、报表生成器等等,提供了 RQBE 范例关系查询、项目管理、跟踪调试工具、跨平台转换器、应用程序生成器等等。这些工具的使用,大大地提高了应用程序开发效率。

FoxPro 提供了友好的图形用户界面,提供了非编程用户接口,可以使用户不必要记忆如同 dBASE 那样的众多的操作命令,利用菜单系统用户用点菜方式工作,每个在菜单系统下的选择都能自动在命令窗口生成等效的命令。FoxPro 支持键盘方式和鼠标方式进行操作,在 FoxPro 内,除开可以使用系统本身提供的大量窗口外,用户可以自定义窗口,窗口使用方式灵活,可以放大、缩小、移动、排序、消隐和恢复。

FoxPro 应用程序可以部分或全部编译成 .EXE 文件,从根本上解决了保密问题,这是一个极大的优点,其高级应用程序接口 API 可以使用户直接调用 C 语言或汇编语言函数。系统具有极好的打印输出控制,一个控制功能复杂的打印报表输出可能只需数十行程序就能完成。

FoxPro 突破了 DOS 640KB 内存限制,能在不需要任何高区内存驱动程序的情况下,接管 640KB 以上内存区,且基本 640KB 内存只需留下 200KB 左右的剩余空间即可运行,为高速存贮数据、优化程序结构创造了条件。其兼容性与开发性不仅允许用 dBASE、FoxBASE 编写的应用软件可以不加修改地应用于 FoxPro 环境,其开发性还使得它可以直接与 Microsoft 的 EXECL, LOTUS 1-2-3、PARADOX 等系统交换数据。

为了应用程序能支持汉字, FoxPro 本身已有汉化的版本。如果应用西文版本,则需要能支持直接写屏幕,能处理好西文制表符的汉字系统软件支持,我国已有很多这样的软件如希望电脑公司的 UC DOS, 中国龙汉字系统 ACIOS, 华达汉字系统、联想汉卡、长城汉卡等均可支持处理汉字,不同汉字系统及版本高低在使用中略有区别。

思考题

1. 何谓信息、数据、数据处理?
2. 何谓数据库、数据库管理系统? 数据库中数据模型有哪几类?
3. 关系数据库的主要特点是什么?

第 2 章 FoxPro 基础知识

FoxPro 是为数据处理开发的软件产品,它具有对数据的分类、检索和处理等功能,它要求初学者掌握一些基础知识。

§ 2.1 数据类型

数据类型是指简单数据的基本属性,数据类型是一个十分重要的概念,因为数据操作必须遵循一条基本原则,只有相同类型的数据之间才能进行操作,否则就要出现语法错误。

在第 1 章已经指出,在关系数据库中处理的数据都要表示为一个二维表,例如如果要开发一个人事管理系统,建立的人事管理关系二维表如下:

编号	姓 名	性 别	出生日期	籍贯	婚否	职 称	工 资	简 历
1001	黎晓菲	女	09/10/70	广东广州市	F	技术员	350.00	Memo
1002	汪爱明	女	03/18/48	广西桂林市	T	工程师	520.50	Memo
1003	刘明维	男	08/03/45	湖南长沙市	T	工程师	540.50	Memo
:	:	:	:	:	:	:	:	:

在这个二维表中每一行有 9 个字段,第一行是各字段的字段名称,每一个字段有一个数据类型,第一行形成的“框”是各个字段“型”的集合。表中有三个记录,表示了不同的人,其中各字段都有特定的值,其值必须符合特定的数据类型。整个二维表在计算机中形成一个数据库文件,例如为 RSGL.DBF 文件。

FoxPro 定义了七种类型的数据,它们是:

1. 字符型 (C 型) 数据

它由可打印的字符、数字、空格及其他专用字符组成,汉字也属于字符型数据,一个汉字的宽度为两个字节,其他字符都是一个字节,一个字符型字段的最大宽度为 254 个字节。

在人事管理关系中,姓名、性别、籍贯、职称都是字符型数据,类型标识用 C (Character) 表示。

2. 数字型 (N 型) 数据

数字型数据由正、负号,小数点和 0 至 9 的数字组成,是可以对其进行数学运算的数据,用字母 N (Numeric) 表示。数字型字段的宽度最大为 20 位 (包括小数点和符号位)。在人事管理关系中,编号和工资都属于数字型数据。

3. 浮点型 (F 型) 数据

浮点型数据也是由正、负号,小数点和数字 0 至 9 组成,最大长度也是 20。在这里,浮点的含义是指小数点可以“浮动”,适用于需进行大量科学计算要求数值位数长且要求较高精度的场合。在一般日常生活中,用数字型数据都能满足数值计算的要求,数字型数据的小数

点位置是固定的而已。浮点型数据用字母 F (Float) 标识。

4. 日期型 (D 型) 数据

日期型数据存放日期数据, 类型标识为字母 D (Data), 通常采用美国格式“月/日/年”形式, 月、日、年均均为二位数字, 日期型数据的宽度固定为 8 位。人事管理关系中, 出生日期即为日期型数据。

5. 逻辑型 (L 型) 数据

逻辑型数据表示逻辑判断的结果值, 只有“真”(True) 或“假”(False) 两种结果, 真用 .T. 表示, 假用 .F. 表示, 类型标识为 L (Logical)。人事管理关系中, 婚否字段就用了逻辑型数据。

6. 备注型 (M 型) 数据

备注型数据仅用于数据库备注型字段, 前面指出过字符型数据的最大长度不超过 254 个字符, 当要存储的数据超过这一限额时, 可设置成备注型字段, 备注型字段可以输入任意长度的有关信息, 例如容纳一个较长的报告。备注型数据宽度固定为十个字节, 这 10 个字节不是用来存储该备注文件的真实内容, 而是存放一个块号 (或指针), 指向备注文件真实内容的存放位置。备注文件的真实内容存放在另一个与数据库文件同名而扩展名为 .FPT 的单独文件中, 在列表显示数据库文件内容时, 备注字段中仅列出“Memo”而看不到具体内容, 若要显示“Memo”字段内容, 可用 FoxPro 提供的 LIST 命令, 在字段名表中指出“Memo”的字段名即可。

备注型数据标识用 M (Memory) 表示。

7. 图象型 (P 型) 数据

这是一种特殊的数据类型, 允许用户存放图象信息。这种类型的字段目前只有在 Apple Macintosh 机型上运行的 FoxPro 版本才能支持, 在 FoxPro for Windows 上可以使用。

§ 2.2 常量与变量

程序中出现的任何量均有名和值两个方面。名反映其语法, 即要按一定的语法规则取名; 值反映其语义, 即值有含义。在程序执行中, 其值恒为不变的量称常量, 又称常数; 值为可变的量称变量。对常量而言, 名和值是一致的, 名即为值, 值即为名, 如整数 5, 它既代表名, 又代表值, 名和值合二为一。对变量而言, 名和值是不一致的, 名为标识符, 固定不变, 而值则随类型不同而定, 并在程序执行中不断变化, 故称变量。

1. 常量

FoxPro 中, 常量为数值常量、字符常量、日期常量、逻辑常量。除数值常量直接用数字表示外, 其他常量均需定用定界符: 字符常量由单引号、双引号或中括号定界; 日期常量由大括号定界; 逻辑常量由圆点定界, 如 168、“参考消息”、{02/03/94}、.T. 分别是数值常量、字符常量、日期常量和逻辑常量。日期常量的表示在 FoxPro 2.0 以上版本中才可使用。

2. 变量

FoxPro 中, 变量有两大类: 字段变量和内存变量。变量名均只能由字母、数字、汉字和下划线构成, 除内存变量中的系统内存变量名以下划线“_”打头外, 其他所有变量名一般以字母打头, 长度不超过 10 个字符。要注意的是, 变量名中决不允许有空格, 否则键入后出错。如 A1、A5、AB_1、Name、GZ 为正确的变量名。为避免混淆, 变量名最好不要取成 FoxPro

的命令或函数名。

(1) 内存变量

内存变量是一种独立于数据库文件而存在的变量，或者说是一种临时工作单元。它又可分为两类：一类是系统定义的内存变量，称系统内存变量。另一类是用户根据需要随时定义的内存变量。用户定义的内存变量，常用于存放常数、中间结果和最终结果，也可以作为控制变量来控制应用程序的运行。内存变量中存放什么类型的数据（N、C、F、L、D），该变量就具有相应的属性，即数值型内存变量、字符型内存变量等等，并且通过赋值语句可以改变内存变量的属性。从内存变量的组织形式上看，它可以分为标量（即非数组中的单个内存变量）与数组两类。用户定义的内存变量中的标量，一般通过赋值语句同时建立与赋值，并将该变量的类型定义为所赋数据的类型。用户定义的内存变量中的数组只能为一维或二维。一个数组只占一个内存变量的名额，一般通过数组定义命令 DIMENSION 建立。数组中各元素的初始值为逻辑常量·F·，通过赋值语句可对整个数组同时赋值，也可对各元素分别赋值，并将它们的类型自动定义为相应数据的类型。数组中各元素的数据类型可以不同。

内存变量的作用域有全局和局部两种。全局（Public）内存变量建立之后，除非删除它，否则在各级程序执行中始终起作用。局部（Private）内存变量只在建立它的程序中以及被该程序调用的下级程序中有效，一旦退出该程序，即自行删除。内存变量可以保存到内存变量文件中，也可以从该文件中恢复到内存中。

系统内存变量是 FoxPro 自动生成和维护的“内置”内存变量。这些系统内存变量大多包含有控制 FoxPro 输出和显示格式的信息。系统内存变量不影响@...say/@...To 命令的输出。有些系统内存变量允许用户初始化和确定其返回值，例如，系统内存变量——Peject 用于设置打印输出的走纸方式，系统默认值为“Before”，即先走纸后打印，用户可将其设置为“Non”，即不走纸。没作特殊说明的系统内存变量被默认为全局变量，也可把系统内存变量指定为局部变量。表 2.1 中列出了部分系统内存变量名及其功能。

表 2.1 FoxPro 系统内存变量

系统内存变量	功 能	类型	默认值
_ALIGNMENT	确定输出文本的对齐位置	C	LEFT
_BOX	决定是否在文本周围显示方框	L	·T·
_CALCMEM	保存计算器的存储值	N	0.0
_CALCVALUE	保存或初始化被显示的计算器值	N	0.0
_CLIPTEXT	保存或初始化裁剪板的内容	C	NULL
_CUROBJ	存放当前 GET 命令的目标号	N	0
_DBLCLICK	鼠标器两次按钮之间的时间间隔	N	0.5
_DLARYDATE	保存或初始化 Calendar/Diary 中的日期	D	当前日期
_INDENT	确定回车换行后的起始位置相对于原来起始位置的偏移值	N	0
_LMARGIN	设置? 命令输出时左边界的空白字符数	N	0
_MLINE	确定用 MLINE() 函数输出备注文本时, 输出行号的偏移值	N	0
_PADVANCE	确定是用 FORMFEED 还是用 LINEFEED 作为换页控制符	C	FORMFEED
:	:	:	:
:	:	:	:

(2) 字段变量

字段变量为数据库文件中记录型的组成要素，也即二维表中第一行的标题栏目。根据它们的取值类型，可分为数值型、字符型、日期型、逻辑型、浮点数型和备注型等。

由于字段变量的保存形式即数据库文件，因此它的作用时间随文件打开而开始，随文件的关闭而终止。通常，数据操作是针对当前工作区数据库的字段进行的，故数据工作区也可视为字段变量的作用域。然而，FoxPro 中绝大多数数据操作也可使用非当前工作区的字段，但要在这些字段名前冠以所在工作区或数据库别名。

对一个数据库文件中某一字段变量而言，该数据库文件有多少条记录，它就有多个值。因此，字段变量是多值变量，它的值随记录的不同而不同，只有记录指针所指记录的字段内容，才是它的当前值。只要记录指针一移动，它的当前值就随之改变。相对而言，内存变量是一种单值变量，只要不重新赋值，它的当前值就保持不变。

要指出的是，字段变量和内存变量只要类型相同，就可以互相传递，并可参与相应类型的数据操作。如果类型不同，则必须通过类型转换函数把类型统一后，方可传递或运算。字段与数组相互传递时，一个字段对应数组中的一个元素，一个记录对应一个数组。

§ 2.3 运算符和表达式

运算符常用于数值、关系、逻辑和日期四类数据的运算，它由系统定义。表达式由运算对象和运算符按一定规则组合而成。运算对象指常量、字段变量、内存变量及函数。表达式由用户定义，是大部分语句的重要组成部分。

2.3.1 运算符

FoxPro 定义了算术、关系、逻辑及字符串四类运算符。

(1) 算术运算符（以下按运算优先次序排列）

()	括号	*, /	乘、除
*, * 或 ^	乘方	+, -	加、减以及运算正、负

算术运算符对数值型数据进行运算，产生数值型运算结果，但加号和减号可对日期型数据进行运算，产生日期型结果，即一个日期型数据可以加或减去一个表示天数的数，结果仍为一个日期型数据。两个日期型数据相减，差为两者之间相减的天数。

此外，所有同一级的运算，如 +, -, 或 *, /, 均从左到右依次进行，括号的优先级高于运算。括号无大、小之分，可多层套用。

(2) 关系运算符（其运算优先级相同）

<	小于	<> 或 #	不等于
>	大于	<=	小于等于
=	等于	>=	大于等于

关系运算符又称比较运算，被比较的两个表达式必须具有相同的数据类型，比较的结果只能为逻辑真或逻辑假。

(3) 逻辑运算符（以下按运算优先级次序排列）

()	括号	• AND •	逻辑与
• NOT •	逻辑非	• OR •	逻辑或

逻辑运算符用于一个或多个逻辑表达式进行逻辑运算，只产生逻辑值真与假。

(4) 字符串运算符（其运算优先级相同）

+	连接运算符	==	字符串全等比较
-	连接运算符	\$	字符串比较

+或-均是两个字符串连接成一个字符串，但后者与前者有所不同的是后者将第一个字符串尾部的空格移到第二个字符串的后面。

2.3.2 表达式

前已指出，表达式是运算对象与运算符的有规则结合，而运算对象可能具有不同的数据类型，但运算结果只能是一种数据类型，即表达式的类型。如前述日期型数据可与数值型数据相加减，但结果仍为日期型数据。又如，表达式：婚否 = .T. .AND. 工资 > 100，运算结果为逻辑型数据。若据此来划分表达式的种类，则可分为数值、字符、逻辑、日期及备注五类表达式。关系表达式只不过是逻辑表达式的特殊形式而已，故不单独列出。

特别要指出的是：单个常量、单个字段变量、单个内存变量或函数都是表达式的形式之一，记住这一点以后使用 FoxPro 的命令很有好处。

(1) 数值表达式

其由数值型常量、字段、内存变量及函数与算术运算符连接而成，其运算结果为数值型数据。例如，100，A-B，（设 A、B 为数值型内存变量），工资（设它为数值型字段），工资 + 15 均为数值表达式。

(2) 字符表达式

其由字符串运算符，连接字符型常量、变量、函数或函数表达式所构成，运算结构仍为字符串。例如：“AB ” + “C”，“AB ” - “C”，姓名（设它为字段名）均为字符表达式。这里符号“ ”表示一个空格，下同。

(3) 逻辑表达式

其由逻辑运算符，连接逻辑常量、变量、逻辑函数或关系表达式组成。例如：.NOT. A（设 A 为逻辑变量），P > 1（设 P 为内存变量），年龄 > 30 .AND. 工资 < 100，均为逻辑表达式。在一个复杂的逻辑表达式中，各种运算顺序是：算术运算、关系运算、逻辑运算。

(4) 日期表达式

日期表达式由数值常量、日期字段、日期内存变量、日期函数与算术运算符 + 或 - 连接而成。例如：

A = CTOD("04/16/91")

B = CTOD("04/10/91")

则 A - B, A - 15, CTOD("04/16/91"), CTOD("04/10/91"), {02/10/95} 均为日期表达式。

由运算符、变量、函数、常量组成的表达式及其结果数值举例如下。下面的例子中使用了 FoxPro 的显示命令：?。在 FoxPro 命令窗口中键入以下表达式，FoxPro 则给出其结果数值。

? INT(34.76 * 2 + 735.57 * 40) / 10 - 11

2938.20

? "How " + "are you"

```
How are you
? "are" $ "How are you"
.T.
? {05/08/94}+40
06/17/94
```

§ 2.4 函数

函数是常数、变量之外的另一类数据，是表达式的成份之一，FoxPro 有 200 多个函数（见附录 3），其函数的概念与数学中函数的概念类似，每一个函数有其自变量，函数给出一个确定的对应值。使用 FoxPro 系统提供的函数，可以极大的扩展用户使用功能，增强各类命令或控制语句的工作内容。

FoxPro 函数按功能可分为算术运算函数、字符运算函数、日期和时间运算函数、转换函数、状态测试函数、标识函数、输入函数、系统函数及自定义函数等。在使用函数时要注意函数自变量的数据类型和函数返回值类型和其功能。

2.4.1 函数分类

按照对函数自变量给定的情况，FoxPro 中函数有三大类。

1. 自变量必须由用户给定。大部分函数都属于这种情况，例如：文件测试函数 FILE(<字符表达式>)，其自变量是一个用字符表达式表示的文件名，必须由用户给出文件名后该函数才能测试该文件。

2. 函数自变量用户可以给定也可以不给出。如果不给出自变量，系统给出一个约定的缺省值。例如数据库文件是否结束的测试函数 EOF([<别名>])，如果用户不指定<别名>则系统视为 EOF()，返回当前选定工作区的数据库文件状态。

这一类函数有 10 多个，如 BOF()、EOF()、DBF()、DELETED()、FCOUNT()、MESSAGE()、RECCOUNT()、RECON()、RECSIZE()等。

3. 函数自变量由系统指定，用户不需再给出。这类函数有近 20 个，如 COL()、DATE()、ERROR()、DISKSPACE()、FKMAX()、FLOCK()、ISCOLOR()、LUPDATE()、OS()、PCOL()、REDDKY()、RLOCK()、LOCK()、ROW()、SELECT()、UPDATE()、VERSION()等。这类函数大部分能对系统各方面的情况进行测试，返回系统的状态及有关参数，如 OS()返回支持 FoxPro 运行的宿主操作系统的名字。

2.4.2 函数应用示例

下面举例说明一些函数的使用。

(1) 可用于改变字符数据的函数(缩写 Char exp 代表的是字符型表达式)。这些函数只对字母起作用，对数据和字符字段中的特殊字符无任何影响。

- UPPER(<Char exp>)可将字母变为大写。例如：UPPER(FNAME)全部按大写字母返回 Fname 字段内容，UPPER("hello")返回 HELLO。
- LOWER(<Char exp>)返回小写字母。例如：LOWER(FNAME)按小写字母返回 Fname 字段的内容，LOWER("hello")返回 hello。