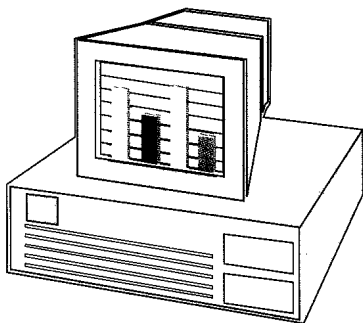




面向21世纪课程建设系列教材

FoxBASE+ 关系型数据库基础

主 编 刘剑辉
副主编 周三元



12.00

机械工业出版社

前 言

FoxBASE + 系统是美国 Fox 软件公司推出的关系型数据库管理系统, 它广泛应用于企、事业管理、办公自动化等各个领域。目前介绍 FoxBASE + 的书籍很多, 但大多为“使用手册”的形式, 初学者不易掌握, 为此我们根据多年数据库教学的经验, 并结合会计电算化的需要编写了本书。

本书第一、二章由王茜编写, 第三、四、七、九章由刘俭辉编写, 第五、六章由周三元编写, 第八章由刘俭辉和曹伟共同编写, 第十章由张福堂编写。全书由刘俭辉负责修订、总纂和定稿, 朱广忠副教授予以审定, 同时张福堂副教授对全书提出了不少有益的建议, 在此深表谢意!

编 者

1998. 1

目 录

前言

第一章 FoxBASE + 基础知识	1
§1.1 数据库系统的基本概念	1
§1.2 FoxBASE + 的主要特点和性能指标	4
§1.3 FoxBASE + 的基本语言要素	6
§1.4 FoxBASE + 命令结构	14
第二章 数据库的建立与显示	15
§2.1 数据库文件的建立	15
§2.2 数据库文件的打开与关闭	19
§2.3 数据库文件的显示	20
§2.4 数据库记录指针的定位及常用文件测试函数的使用	23
第三章 数据库的日常维护	26
§3.1 数据库结构的修改	26
§3.2 数据库文件记录的修改	26
§3.3 数据库文件记录的添加	31
§3.4 数据库文件记录的删除	32
第四章 数据库文件记录的检索和统计	36
§4.1 数据库文件记录的顺序查询	36
§4.2 数据库文件记录的索引查询	38
§4.3 数据库文件的排序	45
§4.4 数据库文件记录的统计与汇总	47
第五章 内存变量与函数	51
§5.1 内存变量	51
§5.2 数组	54
§5.3 FoxBASE + 函数	58
第六章 多重数据库操作与磁盘文件操作	69
§6.1 工作区和多重数据库操作	69
§6.2 FoxBASE + 文件操作命令	76
第七章 程序设计基础	82
§7.1 命令文件的建立与执行	82
§7.2 命令运行环境与参数的设置	84
§7.3 非格式化输入输出命令的使用	87
§7.4 屏幕格式设计和格式化输入输出命令	90
§7.5 程序的分支结构	95
§7.6 程序的循环结构	100

§7.7 过程文件及其调用	104
第八章 应用程序设计	111
§8.1 结构化的系统设计方法	111
§8.2 系统分析与设计实例	113
§8.3 菜单程序设计	117
§8.4 应用程序设计的一般实现方法	123
*第九章 多用户的应用	143
§9.1 多用户应用概述	143
§9.2 MFoxBASE + 常用的多用户命令和函数	144
第十章 FoxPro 数据库系统基础知识	149
§10.1 FoxPro 2.5 数据库系统概述	149
§10.2 FoxPro 菜单、窗口与对话框	151
§10.3 FoxPro 数据库操作	155
§10.4 FoxPro 集成开发环境	160
附录 A FoxBASE + 命令一览表	165
附录 B FoxBASE + 函数一览表	175

第一章 FoxBASE + 基础知识

随着计算机技术的发展,计算机系统从应用于科学计算领域逐渐扩展到应用于数据处理的各个领域,尤其是微型计算机系统在企业中的应用,受到了普遍的关注。计算机应用于企事业管理与应用于一般的科学计算不同,它面临着大量数据的输入、输出及数据的管理和使用等问题。对一个企业来说,能否管理好和充分利用好这些数据资源,将直接影响自己的经济效益。数据库技术就是为了满足这种需要而产生和发展起来的。

§1.1 数据库系统的基本概念

一、数据、信息与数据处理

数据,通常指记录下的原始事实,是反映客观实体的性质、形态、结构和特征的一组符号。如采购一批材料材料名称、供货单位、价格、数量等都是数据,记帐用的会计凭证也是数据。

信息是经过加工处理后对使用者有用的数据,也就是说,信息实质上是经过加工处理后的数据,是对数据的解释和内涵的提取。

数据与信息两者相互依存,数据是表达信息的编码符号,用以度量信息,信息则是数据的内涵,是依附于数据而存在的。两者在概念上不尽相同,但是在使用时人们往往并不严格去区分它们。信息与数据是相对的,一个系统输出的信息可能成为另一系统的输入数据。

将数据加工处理成信息的过程称为数据处理,它泛指数据收集、存储、分类、检索、计算、汇总和传输等一系列活动,数据处理的目的是从大量原始资料中,提取对人们有用的信息作为决策依据,因此,数据处理也可称为信息处理。一项数据处理任务包括几个环节:数据的采集、校验、传输、加工、存储和输出。

数据处理一般是处理非数值计算问题,它涉及的计算公式通常比较简单,不涉及复杂的数学模型,但数据处理量非常大,且数据之间常常存在复杂的逻辑关系,因此数据处理中的基本矛盾就是合理组织数据、有效管理数据。

把数据转换成信息的系统是信息系统,也称为数据处理系统。信息系统是由人员、设备、程序和数据等要素组成的向人们提供有用信息的系统。任何信息系统应具有输入、处理、输出和存储、传输的功能。

二、数据管理技术的发展

数据管理是与数据处理相联系的,所谓数据管理,是指数据的收集、组织、存储、维护、检索和传递等各种操作。这些操作是数据处理业务中的基本环节,任何数据处理,首先是有效地管理好数据,然后才谈得上处理。而数据库技术,就是研究数据管理的,研究如何组织和存储数据,并能提供一种通用的、高效的、使用方便的管理软件,以便最大限度地减轻用户管理数据的负担。

数据管理技术的发展经历了人工管理、文件管理和数据库管理三个阶段。这三个阶段的区分主要表现在程序与数据的关系上。

1. 人工管理阶段

人工管理阶段是计算机用于数据处理的初级阶段,在该阶段,数据的管理是由程序员个人考虑和安排的。程序员在编程时要考虑到数据的逻辑定义和物理组织以及数据在计算机存储设备内的物理存储方式,程序和数据混为一体,在需要引用数据时,直接按地址存取,数据与程序不独立,一旦数据存储发生改变,就必须修改程序。人工管理阶段的程序与数据是一体化的。现在已很少使用。

2. 文件管理阶段

为了摆脱早期直接管理数据方式给程序员造成的负担,50年代后期出现了文件管理系统。程序使用的数据存放在独立的数据文件中,程序和数据文件在逻辑上是分开的。程序员不必涉及数据物理管理的细节,而是通过文件管理系统对数据文件进行存取。

文件管理方式存在许多不足,主要表现在:

- 1) 各数据文件基本上是因程序员自身的程序需要而建立的,数据文件和程序之间存在着密切的依赖关系。数据文件或程序的内容一旦发生变动,另一方通常都需随之作相应变动。
- 2) 由于数据文件是为专门程序设计的,因而容易造成数据冗余。
- 3) 数据文件中的数据可能经常要进行增、删、改等操作,由于用户的需求各不相同,同一数据项可能在多个数据文件中同时存在(数据冗余),修改起来非常繁琐,同时也增加了操作出错的机会并增大了管理的难度。

3. 数据库管理阶段

数据库系统是在文件系统的基础上发展起来的,是以实现数据共享为目标的文件集合。数据库不仅表现为多个文件的集合,而且蕴含了对相关文件的重新组织、设计以实现数据共享的功能。即存储在某个数据库中的数据资源可为多个用户、多种用途所使用。

数据库管理方式有以下几个特点:

- 1) 数据冗余度最小,且数据不依附于用户程序,具有较好的冗余度和较高的数据独立性。
- 2) 数据库的数据由数据库管理系统实行统一管理,各类不同用户均可按需要使用其中的相关数据,实现了数据共享。
- 3) 数据库管理系统配有简单灵活的数据库操作语言、提供了对数据的安全保护、并发操作等一系列功能。

三、数据库系统构成

数据库系统(DBS-DataBase System)由三部分组成:用户应用程序、数据库管理系统和存储在外存储器(通常为磁盘)上的各种数据资源。

1. 用户程序

用户通过自己编写的应用程序来调用数据库中存储的数据,应用程序都是按照用户的实际需要编写的。在数据库系统中还自备用户编程命令,相当于计算机高级语言中的各种语句

2. 数据库管理系统(DBMS-DATABASE Management System)

数据库系统和文件系统最大的不同就是在数据库系统中设置了数据库管理系统。它担负着对数据资源的统一管理,并负责执行用户发出的各种请求命令。

与文件系统不同的是,在数据库系统中用户不能直接与存储的数据打交道,用户对数据库进行的各种数据操作都是通过数据库管理系统实现的。

3. 数据

数据就是指在外部存储介质(一般指磁盘)上存储的数据,它以库文件的形式存放。

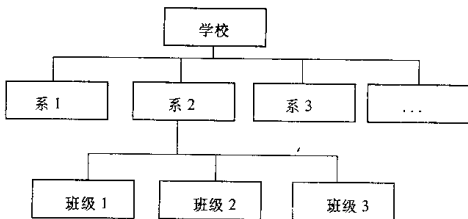
我们平时谈到的数据库实际上涉及了三个层次的概念。数据库(DB - DataBase)是指存储的数据,数据库管理系统担负着对数据资源的管理并接受用户对数据库的访问,数据库系统是采用数据库技术完成特定管理任务的、软硬件结合的人机结合系统。

四、数据模型

数据模型,是指数据之间的逻辑关系,又称为数据结构。常用的数据模型有层次模型、网状模型、关系模型。

1. 层次模型

用树型结构来表示实体之间联系的模型为层次模型。如一个学校的组织结构就是一个层次结构:



层次模型有如下几个特点:

- 1) 有且仅有一个结点无双亲,这个结点为树的根,称为根结点。
- 2) 其它结点有且只有一个父结点。

这种模型的特点是层次分明,结构清晰,适合于表示层次关系比较强的实际问题。

2. 网状模型

当我们处理的问题要同时涉及多种复杂的联系时就需要用到网状模型。网状模型的特点是:

- 1) 可以有二个以上的结点无双亲。
- 2) 可以有一个以上的结点有两个或更多双亲。

网状模型适合于表示数据关系比较复杂的问题,比较成熟的网状模型的数据库管理系统一般用于大中型计算机,微型计算机上基本不使用。

3. 关系模型

关系模型把数据看成是一个满足一定条件的二维表,每个二维表是一个关系。表中的每一行代表一个实体的所有信息,称为一条记录(又叫元组);表中的每一列为不同实体的同一属性,称为字段(又叫数据项),相当于记录的一个记录项,列标题就是对应的字段名。数据项是二维表中不能再分的最小数据单位。

关系模型所满足的条件是:

- 1) 表中不允许有重复的字段名。

2) 表中同一列数据的类型必须相同。

3) 表中行的次序、列的次序均可任意排列,行或列排序的先后次序并不影响表中的数据关系。

例如人事档案管理就是一种二维表。由表 1-1、1-2、1-3 可说明关系间的联系。

表 1-1 职工档案

职工号	姓名	出生年月	性别	部门	职称
0025	李小明	10/11/65	女	保卫科	工程师
0038	张 宁	08/23/59	男	财务科	会计师
...	...	...	...	...	...

表 1-2 工资档案

职工号	基本工资	奖金工资
0025	475.00	210.00
0038	589.00	312.00
...	...	...

表 1-3 干部档案

部门	负责人名	职务
保卫科	李小明	科长
财务科	刘建新	科长
...	...	...

根据以上关系,可以回答出用户的有关提问,例如:

- 1) 各个对象的情况,如:李小明的年龄、性别等。
- 2) 表现关系间的联系,如:李小明的基本工资和奖金。
- 3) 表现关系间的复合联系,如:张宁的科长的工资是多少。

通过使用数据库管理系统实现关系及关系间的关联与映射,能较全面地反映出有关的数据信息。同时,数据库管理系统还提供对数据的各种操作命令,用户可方便地进行对数据库的增、删、改、查等操作。

§1.2 FoxBASE + 的主要特点和性能指标

一、FoxBASE + 简介

FoxBASE + (FoxBASE PLUS) 是美国 Fox 公司于 1987 年 2 月推出的关系型数据库管理系统,它与 DBASE III 及 DBASE III PLUS 完全兼容,同时又作了许多重要的扩充,比 DBASE III 速度更快,功能更强。FoxBASE + 目前使用的版本为 2.1。

二、FoxBASE + 的主要特点

(1) 兼容性 它与 DBASE III 完全兼容,在 DBASE III 环境下编写的程序,不作任何修改就可转移至 FoxBASE + 环境下运行。

(2) 可移植性 由于 FoxBASE + 是采用 C 语言开发的,因此可以方便地移植和推广。同时, FoxBASE + 的多用户版本和单用户版本完全兼容,且用户的应用软件随着硬件和系统的发展也很容易向高档环境移植。

(3) 运行速度快 它是目前市场上运行速度较快且与 DBASE III 兼容的数据库管理系统。

(4) 多种运行方式 FoxBASE + 既可在交互方式下运行,也可在程序方式下运行。在程序方式下既可解释执行,也可通过编译产生目标文件来执行。

三、FoxBASE + 的功能指标 (V2.10)

每个数据库文件记录的最大数	10 亿
每个数据库文件最大字节数	20 亿
每个记录的最大长度	4,000
每个数据库的最大字段个数	128
字符型字段的最大宽度	254
数值型字段的最大宽度	19
日期型字段的固定宽度	8
逻辑型字段的固定宽度	1
备注型字段的固定宽度	10
数值型数据的有效数字	16
内存变量个数	默认 256, 最多 3600
字符型变量的最大长度	254
字符串的最大长度	254
最多可定义的数组个数	3600
数组元素的最大数	3600
每个命令行字符的最大数	254
可同时打开的文件数	默认 16, 最多 48
最多可同时打开的库文件个数	10
最多可同时打开的索引文件个数	每个数据库 7 个
过程文件中子过程的最大数	128

四、FoxBASE + 的组成与运行环境

1. FoxBASE + 系统组成文件

FoxBASE + 系统分为单用户 FoxBASE + 和多用户 FoxBASE + 两种, 多用户 FoxBASE + 也称为 MFoxBASE +。单用户 FoxBASE + 只能在单用户环境下运行 (如 DOS 系统); 而多用户 MFoxBASE + 即可在单用户环境下运行, 也可在多用户环境下运行 (如网络系统或 XENIX 系统)。MFoxBASE + 系统的组成文件有:

MFOXPLUS. EXE	主控模块
MFOXPLUS. OVL	覆盖模块
FOXPCOMP. EXE	编译模块
FOXPHelp. HLP	帮助模块
FOXBIND. EXE	过程文件连接器

其中多用户的 FoxBASE + 最基本的组成模块是:

MFOXPLUS. EXE
MFOXPLUS. OVL

单用户 FoxBASE + 最基本的组成模块是:

FOXPLUS. EXE
FOXPLUS. OVL

2. FoxBASE + 系统运行环境

FoxBASE+ 对硬件系统环境要求并不高,通常普通的 PC 机均可直接使用。如果用户使用 FoxBASE+ 系统时要处理中文信息,应先启动中文操作系统,然后再进入数据库管理系统中。

五、FoxBASE+ 的启动与退出

1. FoxBASE+ 的启动

FoxBASE+ 的启动方法很简单,先进入 FoxBASE+ 系统所在的目录并键入: MFOXPLUS, 确认后出现 “.” 提示,启动成功。此时,系统已在 FoxBASE+ 的控制之下,用户可以使用 FoxBASE+ 的各种命令。

2. FoxBASE+ 的退出

在 FoxBASE+ 的点提示符下输入 QUIT:

QUIT

§1.3 FoxBASE+ 的基本语言要素

一、数据类型

数据类型是指简单数据的基本属性。FoxBASE+ 下,数据类型是一个十分重要的概念,对数据操作必须遵循这样一条基本原则:只有相同类型的数据才能进行相关操作。当用户对不同类型的数据进行相关操作时,将出现语法错误。FoxBASE+ 系统下,用户使用的任何数据都包括型与值两方面的内容,数据的型反映了事物的一般性质,数据的值反映了事物的具体情况。

FoxBASE+ 定义了五种数据类型:字符型数据、数值型数据、逻辑型数据、日期型数据、备注型数据。

1. 字符型数据

字符型数据是指一切可打印的字符,包括 26 个英文字母、10 个阿拉伯数字、各种符号、汉字和空格。字符型数据用字符 C(Character) 来表示。字符型数据不能参加算术运算,但可用 “+” 和 “-” 将几组字符型数据相连,还可按其字符的 ASCII 码值进行大小比较。字符型数据的长度由字符型数据中含有字符的个数决定。字符型数据可定义的最大宽度为 254。

2. 数值型数据

数值型数据是指可以进行算术运算的数据,用字符 N(Numeric) 来表示。数值型数据只能包含阿拉伯数字、小数点、正负号。数值型数据的最大精度为 16 位有效数字。

在定义数值型数据时,必须定义数据总长度和小数位长。数据总长度包括小数点所占的一位。FoxBASE+ 下,用户定义数值型数据的最大宽度为 19。

3. 逻辑型数据

逻辑型数据可用来进行各种逻辑比较。逻辑型数据只有两个值:逻辑真和逻辑假。逻辑真可用 .T.、.t.、.Y.、.y. 表示,逻辑假可用 .F.、.f.、.N.、.n. 表示。逻辑型数据用字符 L(Logical) 表示。逻辑型数据的长度固定为 1。

4. 日期型数据

日期型数据是专门用来表示日期的,固定长度为 8。为便于管理,FoxBASE+ 系统提供了八种日期格式,可由用户按需要选定。系统默认日期型数据的格式为 “MM/DD/YY” 即 “月/日/年” 格式,符合英语书写格式。日期型数据用字符 D(Date) 表示。

日期型数据可以进行加减运算。如果一个日期型数据加上一个数,则系统理解为这个日期加上了一个天数,得到一个新的日期,它仍然是日期型数据。如果一个日期型数据与另一个日期型数据相减,则得到的是这两个日期相差的天数,是一个数值型数据。

5. 备注型数据

备注型数据用字符 M(Memory)表示,其固定长度为 10,用于存放 FoxBASE+ 的内部指针,而实际数据则存放在备注型文件中,这个文件与数据库文件同名,扩展名为 .DBT。对于备注型数据的说法不统一,有的称其为便笺型、明细型、记忆型数据等。备注型数据是 FoxBASE+ 中一种特殊的字符型数据,可被显示、打印和编辑,但不能参加任何类型的运算。

在使用数据库存储数据时,有时各个记录的数据长度相差明显,如职工档案中,每个人(即每个记录)的姓名、性别等项目长度相同,而个人简历和评语等项目有长有短,差别较大,备注型字段就是为存储这类数据而设置的。

二、常量

常量是固定不变的数据。FoxBASE+ 下,常量有四种类型:字符型、数值型、逻辑型、日期型。

1. 字符型常量

字符型常量是用字符定界符括起来的字符型数据,亦称为字符串。FoxBASE+ 有三种字符定界符:单引号、双引号和方括号。定义字符串的定界符必须成对出现,字符串的长度由定界符内的字符个数决定,字符串的最大长度为 254 个字节。

例 1-1 字符型常量的书写及其长度。

'112.35' 长度为 6

"ABC" 长度为 3

[丹东] 长度为 4

2. 数值型常量

数值型常量可以是带符号或不带符号的整数或小数,数值型常量的长度由整数、小数点、小数(负数还包括负号)所占的位数决定,长度不超过 19 位。

例 1-2 数值型常量的书写及其长度。

12.35 长度为 5

23 长度为 2

-150 长度为 4

3. 逻辑型常量

逻辑型常量的固定长度为 1,其值可以是 .T.、.t.、.Y.、.y.,它们均代表真,也可以是 .F.、.f.、.N.、.n.,它们代表假。

4. 日期型常量

日期型常量的格式系统默认为月/日/年。月、日、年分别为两位数字。日期型常量的固定长度为 8。

在 FoxBASE 中日期型常量不能独立存在,一般要先定义为字符型常量,然后通过专门的转换函数转换为日期型数据。

例如 CTOD("08/09/81")为日期型常量,代表值为 81 年 8 月 9 日,长度为 8。

注意:在日常生活中,我们经常会遇到用阿拉伯字母表示的物理参数,用计算机处理这类

信息时,有的被定义为数值型数据,而有的又被定义为字符型数据。FoxBASE + 下使用阿拉伯数字表示数据时,其类型的定义通常遵循这样一条原则:凡能或需要进行算术运算的数据,应将其定义为数值型数据,否则可定义为字符型数据。

例 1-3 数据的类型定义。

设备编号	'051432'	定义为 C 型
设备数量	25	定义为 N 型
电话号码	'444961'	定义为 C 型
年 龄	16	定义为 N 型

三、变量

变量是在操作过程中其内容可以发生变化的数据。FoxBASE + 下,变量分为内存变量和字段名变量两种。变量名可以由字母、汉字、数字和下划线组成,且必须以字母或汉字开头,变量名的长度不能超过 10 个字符。

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,它用于存放数据处理过程中需要用到的各种常量、中间结果和最终结果,以实现与数据文件的信息交换。因此,内存变量是一种临时工作单元,需要时可以临时定义,不用时又可以释放掉,每个内存变量都是独立的,变量的内容可以根据需要随时更新。

内存变量按其存储数据类型不同,可分为四种:C、N、L、D。内存变量的类型是由其存放数据的类型决定的,即在给内存变量赋值时,其类型为“见值识型”的。在 FoxBASE + 下,系统可使用内存变量的默认数目为 256 个,需要时最多可为 3600 个。

(1)字符型内存变量 内存变量的内容为字符型数据时,该变量就是字符型变量。字符型变量长度可达 254 个字节。给内存变量赋字符型数据时必须加字符型数据的定界符。

例 1-4 字符型内存变量的定义。

```
. A = "123.5"
. B = '财政'
. C = {01/01/98}
```

(2)数值型内存变量 内存变量的内容为数值型数据时,该变量就是数值型变量。数值型内存变量可存储的数值精度为 16 位有效数字,长度最大为 19 位。

例 1-5 数值型内存变量的定义。

```
. A = 123.5
. N = -20
```

(3)逻辑型内存变量 内存变量的内容为逻辑值时,该变量就成为逻辑型内存变量。当给内存变量赋的值为 .Y.、.y.、.T.、.t. 或 .N.、.n.、.F.、.f. 时,该变量为逻辑型变量,其保存的值为逻辑真 .T. 或逻辑假 .F., 长度固定为 1。

例 1-6 逻辑型内存变量的定义。

```
. L1 = .y. L1 的值为 .T.
. L2 = .f. L2 的值为 .F.
```

(4)日期型内存变量 存储日期型数据的变量为日期型变量,其长度固定为 8。

例 1-7 日期型内存变量的定义。

. D1 = CTOD(C)

. D2 = DATE()

2. 字段名变量

字段是构成库文件的最基本数据单元。一个库文件中,不同记录的同一字段中存放的数据内容通常是不同的。当用户对数据库操作时,字段名的内容随当前记录的变化而变化,因此字段名为变量。字段名变量有 C、N、L、D 和 M 五种类型。每个字段的数据类型,由专门的建库命令定义和说明。

四、运算符

为便于相应类型的数据进行各种运算操作,FoxBASE + 定义了多种运算符:算术运算符、字符运算符和逻辑运算符等。

1. 算术运算符

算术运算符可以对数值型数据进行常规算术运算,并产生数值型运算结果。算术运算符由六种符号组成。

(): 优先运算符

+ : 加号(单目运算为正号)

- : 减号(单目运算为负号)

* : 乘号

/ : 除号

** 或 ^ : 幂

各个运算符进行运算的优先顺序是:先运算幂,后乘除,最后运算加减,括号优先。遇到同级的运算符,其优先顺序排列则按照自左向右的规则进行运算。

例如 计算 $100 * 10 * * - 2 + 10$ 时,先计算 $10 \div 2$,其值与 100 相乘,最后才与 10 相加,结果本算式的值为 11。

对于日期型数据可以用“+”和“-”进行运算。

例 1-8 日期型数据的运算。

DATE() - CTOD("01/01/98") 其值为数据型数据

DATE() + 20 其值为日期型数据

2. 字符运算符

字符运算符用来对字符型数据进行相连操作,即将多个字符串连接成一个字符串

+ : 将两个字符串连接成一个新的字符串

- : 把首字符串末尾的空格移到后面字符串末尾后将两字符串连接

例 1-9 字符型数据的相连。

'liaoning' + 'dandong' = 'liaoning dandong'

'liaoning' - 'dandong' = 'liaoningdandong'

3. 逻辑运算符

经过运算能够产生逻辑值的运算符,称为逻辑运算符。不同类型的逻辑运算符可以连接不同类型的数据。

(1)比较运算符 比较运算符用于对两个数据的大小进行比较,产生逻辑型的结果数据。被比较的数据可以是数值型数据、字符型数据和日期型数据。数值型数据是按其数值大小进行比

较的,字符是按其 ASCII 码值的大小进行比较的,而汉字是按其机内码(大致与汉语拼音顺序相同)进行比较的。

FoxBASE+ 的比较运算符有:

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

例 1-11 比较运算符的使用。

```
.. A = 10
. C = "ABCD"
. D = CTOD("01/01/98")
. ? A > 5
.T.
. ? C > a
.F.
. ? D = DATE()
.F.
```

注意:比较运算符两边连接的数据类型必须相同,字符比较运算时不要随意添加空格,因为字符比较时,空格也参加比较。虽然各比较运算符的优先级别相同,但用户可以用括号改变运算顺序。

(2)字符串包含运算符\$ FoxBASE+ 系统用它来表示两个字符串之间的包含与被包含关系。\$运算符只能连接字符型的数据,它的使用格式为:

<字符串> \$ <主字符串>

如果主串中包含了串则运算结果为真,反之则为假。

例 1-12 包含运算符的使用。

```
.. C1 = "A"
. C2 = "ABCD"
. ? C1 $ C2
.T.
. ? C1 $ "abcd"
.F.
```

(3)逻辑运算符 逻辑运算符用来对逻辑型数据进行各种逻辑运算,它只能连接逻辑型数据。逻辑运算符有:NOT、AND、和 OR。逻辑运算符的功能特见表 1-4。

表 1-4 逻辑运算符的功能特性

操作符	优先级别	名称	意义
.NOT. .AND. .OR.	最高 其次 最低	逻辑非 逻辑与 逻辑或	条件不成立时结果为真 两个条件同时成立时结果为真 条件之一成立时结果为真

逻辑运算符和其它运算符一样,也可以利用优先运算符()来改变各个逻辑运算符之间的先后操作顺序。

例 1-13 逻辑运算符的使用。

```
. A = 10
. C = "财"
. L = . F.
. ? A > 10. AND. C$ "财政". OR. . NOT. L
. T.
```

五、函数

为提高数据库系统对数据的管理能力,FoxBASE+ 系统提供了上百个具有特殊功能的函数,系统对每个函数的自变量个数、类型、值域有明确规定,函数是一种本身具有运算功能的特殊数据。对于每个函数,FoxBASE+ 系统给定了专门的名字。当用户使用函数时,只需提供合适形式的参数就可调用。

下面列出几个常用函数的使用格式和功能,至于完整的函数介绍参见本书第五章。

函数名称及其格式	功能
INT(<数值表达式>)	取整数部分
ROUND(<数值表达式 I> , <小数位>)	对数值进行四舍五入
LEN(<字符串>)	求字符串的长度
SPACE(<数值表达式>)	生成空格
STR(<数值表达式> [, <长度> [, <小数位>]])	将数值转换成字符串
SUBSTR(<字符串表达式> , <起始位置> [, <长度>])	字符串截取
DTOC(<日期型表达式>)	将日期转换成字符串
CTOD(<字符串表达式>)	将字符串转换成日期
DATE()	求当前系统日期

从上面给出的函数中我们看到,有些函数要求在括号内必须提供参数,这种函数被称为有参函数。而有些函数不必提供参数,被称为无参函数。函数不管是否带有参数,括号都不可缺少。

例 1-14 函数的简单使用。

```
. ?INT(123.56)
123
. ?ROUND(123.56,1)
123.60
. ?LEN("财政")
4
. ?LEN(SPACE(10))
10
. ?STR(123.56,7,1)
123.6
. ?STR(123.56,5,1)
```

123.6

```
. ?STR(123.56, 4, 1)
```

```
124
```

```
. ?STR(123.56, 2, 0)
```

```
*
```

```
. ?SUBSTR("ABCD123", 3, 2)
```

```
CD
```

```
. ?SUBSTR("财政", 3)
```

```
政
```

```
. D = CTOD("01/01/98")
```

```
01/01/98
```

```
. ?DATE()
```

```
01/07/98
```

```
. ?DATE() - D
```

```
6
```

说明:STR()函数在对数值型数据进行转换时,如果要求的保留位数小于转换数值的整数位数时,系统将输出“*”表示溢位。

六、表达式

由相关运算符将常量、变量、函数等数据按一定规则连接起来的整体被称为表达式。

一个表达式,无论长短,经过运算后,总能得到一定的运算结果,也就是说,任何一个表达式像函数一样都能给出一个回送值,因此,表达式也是一种数据。

按表达式运算后得到数据类型的不同来划分,表达式包括算术运算表达式、字符运算表达式、日期表达式和条件表达式。

1. 算术运算表达式

算术运算表达式是指一个数值型数据或用算术运算符和括号将数值型数据按一定规则连接而成的式子,算术运算表达式运算的结果为数值型数据。

例 1-15 算术运算表达式的计算。

```
. ?LEN("计算机") + 10 * * 2
```

```
106
```

2. 字符运算表达式

一个字符型数据或用字符运算符把字符型数据连接而成的式子,称为一字符运算表达式,字符运算表达式的运算结果为字符型数据。

例 1-16 字符运算表达式的计算。

```
. ?"计算机" + STR(12.56, 3)
```

```
计算机 13
```

3. 日期表达式

一个日期型数据或用日期型数据加减一个数值而构成的表达式称为日期型表达式。

例 1-17 日期型表达式的使用。

```
. ?DATE()
```


01/07/98

. ?DATE() - 10

12/28/98

4. 条件表达式

条件表达式是运算结果为逻辑值的表达式。条件表达式通常分为关系表达式和逻辑表达式两种。

(1)关系表达式 关系表达式是由比较运算符或包含运算符将相关数据连接起来的表达式。关系表达式的运算结果是逻辑型数据。

关系表达式的一般形式是：

1) <算术运算表达式> <比较运算符> <算术运算表达式>

2) <字符运算表达式> <比较运算符> <字符运算表达式>

3) <日期表达式> <比较运算符> <日期表达式>

在关系表达式中,首先进行算术、字符、日期等运算,然后再进行关系运算。表达式中使用比较运算符的两边表达式必须类型相同。

对于字符运算表达式进行关系运算时,还可以使用包含运算符\$,其一般形式为:

<字符串 1> \$ <字符串 2>

若字符串 1 包含在字符串 2 中,则运算结果为真,否则运算结果为假。

例 1-18 关系表达式的计算。

. ?10> 20

. F.

. 性别="男"

男

. ?性别="男"

. T.

. ?"女"\$性别

. F.

注意:性别="男"作为命令直接使用时,"="完成给左边的变量性别用右边数据赋值的功能;而性别="男"作为表达式使用时,"="为比较运算符,完成左右两边数据的比较,返回一个逻辑值。

(2)逻辑表达式

用逻辑运算符连接逻辑型数据的表达式为逻辑表达式。逻辑表达式的值仍为逻辑型数据。系统在运算逻辑表达式时,先进行算术、字符、日期运算,再进行关系运算,最后才进行逻辑运算。

例如,要查找某职工档案中男同志中基本工资高于 500 的人或部门为财务科的女同志,其条件表达式的书写格式为:

性别="男". AND. 基本工资> 500. OR. 部门="财务科". AND. 性别="女"

七、FoxBASE + 的文件分类

FoxBASE + 的各类数据和程序都是以文件的形式存储在存储设备中的。为方便用户操作,FoxBASE + 定义了 9 种不同类型的文件,每种文件按相应的格式进行存取。这 9 种文件分别