

第一章 数据库管理系统概述

随着计算机技术的发展,计算机的应用不断向着非数值计算的各个领域渗透和发展,特别是70年代出现的数据库技术,使计算机更为广泛地应用于国民经济各部门。数据库能有效、合理地存储各种数据,准确而快速地提供信息,因而数据库是数据处理的重要工具,是管理信息系统、办公自动化系统和决策支持系统的核心部分。在当今的信息社会中,数据库技术必将起着越来越重要的作用。

本章主要阐述有关数据库系统的基本概念,如数据库的特点、数据库管理系统和数据模型等;简单介绍微机数据库管理系统FoxBASE+。

1.1 数据库的基本概念

一、数据与数据处理

人类生活在信息社会中,人们的活动离不开对反映客观世界的各种数据的收集、处理、存储和使用。数据(data)是一种物理符号序列,用来记录和描述客观事物。如用来表示事物数量(销售量、单价、工资、职工人数等)的数字是数据;用来表示各种事物的名称(厂名、姓名等)和符号也是数据。这里所说的数据不仅指数字,还可以指文字、图形和声音。现代的计算机可以接受几乎所有种类的数据。信息(information)则是经过加工并对人类社会实践和生产及经营活动产生决策影响的数据。人们起初所收集到的数据经常是杂乱无章的,它并不能带给我们信息,只有对原始数据进行加工提炼后所得到的数据,才为我们提供有用的、新的信息。信息和数据既有联系又有区别,数据是信息的来源,是产生信息的根据;而信息则是数据加工处理后的结果,但并非所有数据都能转化为信息。

信息是通过对数据进行处理而产生的。数据处理是指对各种类型的数据进行收集、存储、分类、计算、加工和传输等操作过程。通常数据处理也称为信息处理。

二、数据管理技术的发展

数据处理的中心问题是对数据的管理,即用计算机对数据进行组织、存储、检索和维护等数据管理工作。随着计算机技术的发展,数据管理技术也经历了人工管理、文件管理和数据库管理三个阶段。

1. 人工管理阶段(50年代中期以前)

这一阶段是计算机管理的初级阶段,对数据的管理是由程序员个人来考虑和安排的,程序员在编制程序时还要考虑数据的存储结构、存储方式、存储地址和输入/输出格式等,当数据的存储位置或输入/输出格式等发生变化时,相应的程序也要随之改变。这个阶段的特点是:数据和程序是紧密地结合为一个整体,一组数据对应于一个程序,数据不具有独立性。程序和数据的关系如图1-1所示。

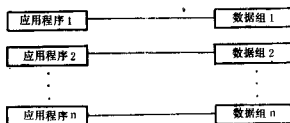


图 1-1

2. 文件系统阶段(50年代后中期到60年代中期)

在这阶段软件有较大的进展,出现了操作系统,一个专门管理数据的软件——文件系统包含在操作系统中,该软件将数据按一定规则组织成为一个有效的数据集合,称为数据文件或文件,在这个阶段数据可以文件形式长期存放在外存设备上,并且数据的存取等操作都有文件系统自动进行管理,文件管理系统成为应用程序和数据文件之间的接口,其关系如图 1-2 所示。

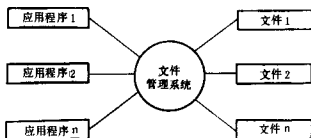


图 1-2

这个阶段的特点是程序与数据分离,数据可长期保存在外存上,由于文件本身基本上是对应一个或几个应用程序,因而存在数据独立性差、共享性差、冗余度大、空间浪费、文件不易扩充等缺点。

3. 数据库系统阶段(60年代后期)

随着计算机技术的发展,硬件方面已有了大容量的磁盘,由于计算机应用于企业管理,数据量急剧增加,迫切要求对数据进行集中控制、提供数据共享,因而研制产生了一种新的数据管理方法即数据库技术,它克服了前面各种管理方式的缺点,使数据管理技术进入一个新阶段。在这阶段,数据库管理系统成为用户与数据的接口,其关系如图 1-3 所示。

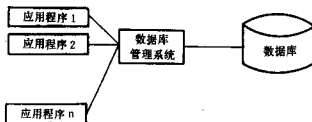


图 1-3

数据库系统的主要特点如下：

(1)实现数据共享,允许多个应用程序存取数据库中的数据。

(2)减少数据的冗余度。在前述文件系统中,用户根据各自的应用而建立文件,使文件之间有大量数据是重复的,称为数据冗余。在数据库系统中,从整体观点来组织数据,数据不是面向个别应用,而是为多种应用所共享,因而数据库系统减少和避免了不必要的冗余。

(3)维护数据的一致性。数据的不一致性主要是由于数据冗余引起的,如果冗余能为系统所控制,则系统就能保证或提高数据更新时数据的一致性。

(4)实现数据的独立性。应用程序与数据之间不存在依赖关系,即数据存储结构或存取方式的改变,程序不受影响。

(5)有较高的数据安全性和完整性。数据库系统对数据实行集中控制和管理,并有防止数据被非法使用和破坏而采取的各种数据保护措施,保证数据的完整性和安全性。

(6)提供数据排序、统计、分析、制表等多种操作。

三、数据库系统(DATA BASE SYSTEM)

数据库系统是一个复杂的系统,可以形象地把它与一个图书馆系统相比拟。图书馆系统有书库、图书管理系统、管理员和用户,书籍是按一定规律摆放在书架上,系统的任务是保管书籍并按管理规则随时为众多用户提供各种服务。

一个数据库系统与图书馆系统类似,它的任务是保存和管理数据,并为用户提供各种服务,但它是一个计算机化的数据管理系统,因而数据库系统通常由数据库、数据库管理系统、计算机硬件与软件系统和用户组成,如图 1-4 所示。

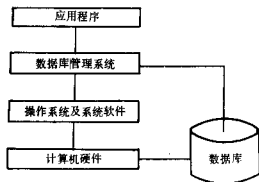


图 1-4

1. 数据库(DATABASE)

数据库是结构化的相关数据的集合。这些数据是按一定的结构和组织方式存储在外存(如磁盘)上,并具有最小的数据冗余,可供多个用户共享。

2. 数据库管理系统(DBMS; DATA BASE MANAGEMENT SYSTEM)

数据库管理系统是在操作系统支持下工作的管理数据的软件,它是整个系统的核心。它负责对数据库中数据进行统一管理,提供以下功能:对数据进行定义;建立数据库;进行增、删、改、查询等操作;数据库的维护、控制;对数据的排序、统计、分析、制表等功能,各种功能都有相应的命令来实现,使用户操作非常方便。

3. 计算机的硬件和软件系统

数据库系统是建立在计算机系统上的,它所需要的最基本的硬件资源是CPU、内存、磁盘及其它外设,在软件方面包括操作系统、作为主语言的高级语言(如 COBOL、PASCAL 和 C 语言等)及其编译系统、描述用户应用需求的各种应用程序等。

4. 用户

通常有三类用户:对数据库系统进行日常维护的数据库管理员(DBA);使用数据库中数据的终端用户;用高级语言和数据操纵语言编制应用程序的程序设计师。

四、数据库语言

数据库语言是用户与数据库管理系统之间的媒介,数据库语言包含数据描述语言(描述和定义数据库结构)和数据操纵语言(对数据进行各种操作)。用户通过使用数据操纵语言对数据库中数据进行存取、查询、统计等操作。数据操纵语言的使用方式有两种:

1. 宿主型:将数据操纵语言嵌入到高级语言中,如嵌入到 COBOL、PASCAL 或 C 等高级语言中,称这种高级语言为主语言,而嵌入的数据操纵语言称为子语言。

2. 自含型:数据操纵语言独立完成对数据库的各种操作,如查询、更新等,以方便非程序员的用户使用数据库中的数据。这种语言功能齐全,效率较高。

五、数据库模型

按照数据库组织方式不同,数据库通常可分为层次数据模型、网状数据模型和关系数据模型。

1. 层次数据模型

层次数据模型是一种树型结构,就象一棵倒置的树,如图 1-5 所示。其特点如下:

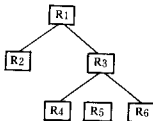


图 1-5

(1) 每个数据元素可与下面任一层多个数据元素相联系,但只能与它上面一层中的一个数据元素相联系。

(2) 最高一层的数据元素称为根,根元素是唯一的,它只有下属的数据元素,不能有向上联系。

2. 网状数据模型

广义地讲,任意一个连通的基本层次联系的集合为网状模型,如图 1-6 所示,其特点如下:

(1) 可以有一个以上的数据元素无向上联系。

(2) 至少有一个数据元素与上层中多个数据元素相联系。

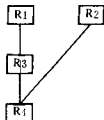


图 1-6

3. 关系数据模型

关系数据模型是一种以数据理论为基础构造的数据模型。它用一张二维表来描述数据之间关系,这张表称为关系,并赋予它一个名字,称为关系名。表 1-1 为一张商品表,在关系模型中即为一个关系,其关系名为商品表,表中每一行为关系的一个元组(tuple),表中每一列为关系的一个属性(attribute)。关系模型中关系应具有如下特点:

- (1) 关系中每一个属性都是不能再分的基本概念。
- (2) 各元组的相同列具有相同的数据类型。
- (3) 每个属性(列)被指定一个不同的名字即属性名,在一个关系中属性名不能重复。
- (4) 每个元组(行)的内容是不相同的,不允许重复。
- (5) 行、列顺序可任意变动,不影响表格信息。

满足以上条件所建立的二维表才能称为关系。表 1-1 符合以上条件,因而是一个关系。

表 1-1 商品表

商品码	商品名称	规格	产地	日期	进口否	单价	备注
0101	西湖彩电	54CD8A	杭州	02/01/93	.F.	2900.50	
0102	熊猫彩电	54P10	南京	01/02/94	.F.	2980.60	
0202	松下空调	1 匹	日本	01/10/94	.T.	6888.80	
0203	科龙空调	1 匹	顺德	02/20/93	.F.	2980.50	
0301	索尼音响	A11K	日本	04/01/94	.T.	4555.50	
0401	荣风电扇	490mm	杭州	06/06/94	.F.	265.50	
0402	华生电扇	400mm	上海	02/05/94	.F.	448.40	
0601	容声冰箱	203L	顺德	05/12/93	.F.	2511.10	
0602	上菱冰箱	203L	上海	06/24/94	.F.	2699.80	
0603	西冷冰箱	190L	杭州	02/17/94	.F.	2108.80	

关系模型是建立在数学理论基础上的,因而有很多关系运算,其中选择、投影和连接三种关系运算尤为重要,使用频繁。

选择:按照一定条件在给定关系中选取若干元组(即选取若干行)。

投影:选择确定的几个属性,并消去其它属性(即选择若干列)。

连接:按照一定条件将两个关系的元组相连(即连接两张表)。

在三种数据模型中由于关系模型具有简单而规范的数据结构,又有严格的数学理论基础,因而成为当前数据库领域应用最广泛的数据模型。

以关系模型为基础的数据库管理系统,称为关系型数据库管理系统(relational data base

management system, RDBMS), 在目前市场上应用较普遍的关系型数据库管理系统有 dBASE III、FoxBASE、FoxPro、INFORMIX、ORACLE、SYBASE 等。

1.2 FoxBASE+关系数据库管理系统

一、FoxBASE+简介

FoxBASE+是美国 Fox Software 公司于 1987 年 2 月推出的关系数据库系统,1987 年 7 月进行修改,成为 FoxBASE+2.0 版,于 1988 年 7 月又推出了 FoxBASE+2.10 版。FoxBASE+不仅与 dBASE III 完全兼容,并且速度更快,功能更强。其主要特点如下:

1. 兼容性。FoxBASE+和 dBASE III PLUS 完全兼容,通过系统提供的有关命令,可将 dBASE III 环境下编写的程序不作任何修改地转移至 FoxBASE+环境下运行。

2. 可移植性。FoxBASE+采用 C 语言开发,因此便于移植和推广。它可运行于不同的操作系统(如 MS-DOS、WINDOWS、UNIX 等)和不同的计算机(如 IBM/PC、VAX/780 等)。FoxBASE+的多用户版本与单用户版本也完全兼容。

3. 运行速度快。FoxBASE+运行速度平均比 dBASE III PLUS 快 5.9 倍,如果配置 80287/80387/80487 协处理器,运行速度可得到进一步提高。

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,在程序方式下既可以解释执行,也可以通过编译产生目标文件而执行。

由于 FoxBASE+具有上述特点,又有汉化 FoxBASE+版本,因而得到广泛的应用。

二、FoxBASE+系统组成

FoxBASE+本身是由一系列文件组成的,称为 FoxBASE+系统文件。

单用户版本 FoxBASE+主要系统文件为:

FOXPLUS·EXE	执行程序
FOXPLUS·OVL	覆盖程序
FOXPHelp·HLP	帮助程序
FOXBIND·EXE	过程文件编辑器
FOXPCOMP·EXE	伪编译程序

多用户版本 FoxBASE+主要系统文件为:

MFOXPLUS·EXE	执行程序
MFOXPLUS·OVL	覆盖程序
FOXPHelp·HLP	帮助文件
FOXBIND·EXE	过程文件编辑器
FOXPCOMP·EXE	伪编译程序

其中执行程序和覆盖程序是运行 FoxBASE+的必备文件。

三、主要性能指标

每个数据库文件的最大记录数:	10 亿个
每个记录的最多字段数:	128 个字节
每个记录的最大长度:	4000 个字节
每个字符字段的最多字符个数:	254 个

数值运算中有效位数: 16 位
 字符串的最多字符个数: 254 个
 每条命令的最多字符个数: 254 个
 索引关键字的最多字符个数: 100 个
 内存变量缺省数: 256 个
 内存变量最多个数: 3600 个
 数组的最多个数: 3600 个
 每个数组的最多元素个数: 3600 个
 可同时打开各类文件的最多个数: 48 个
 可同时打开数据库文件的最多个数: 10 个

本书主要介绍单用户、经汉化的 FoxBASE+2.10 版数据库管理系统, 以下简称为 FoxBASE+ 系统。

1.3 FoxBASE+数据库文件

数据库管理系统处理的对象是数据, 它是以数据库文件形式存放在外存(如磁盘)上的。下面以表 1-1 的二维表为例, 将它设计为一个 FoxBASE+关系数据库文件(以下简称为库文件或文件), 该文件如表 1-2 所示。

表 1-2

Record#	商品码	商品名称	规格	产地	日期	进口否	单价	备注
1	0101	西湖彩电	54CD8A	杭州	02/01/93	.F.	2900.50	Memo
2	0102	熊猫彩电	54P10	南京	01/02/94	.F.	2898.60	Memo
3	0202	松下空调	1匹	日本	01/10/94	.T.	6888.80	Memo
4	0203	科龙空调	1匹	顺德	02/20/93	.F.	2980.50	Memo
5	0301	索尼音响	A11K	日本	04/01/94	.T.	4555.50	Memo
6	0401	乘风电扇	400mm	杭州	06/06/94	.F.	265.50	Memo
7	0402	华生电扇	400mm	上海	02/05/94	.F.	448.40	Memo
8	0601	容声冰箱	203L	顺德	05/12/93	.F.	2511.10	Memo
9	0602	上菱冰箱	203L	上海	06/24/94	.F.	2699.80	Memo
10	0603	西冷冰箱	190L	杭州	02/17/94	.F.	2108.80	Memo

在 FoxBASE+系统中, 文件名的格式为(文件名*.DBF), 其中扩展名“.DBF”是库文件的标志。设该库文件取名为 SP*.DBF。

由表 1-2 可见, 每行由 9 列组成, 在库文件中每列称为一个字段, 每个字段有它的字段名和字段值, 所谓字段名即为该字段取名, 它位于该列第一行, 也就是表中第一行为各字段的名称, 如商品码、商品名称、规格等均为 SP*.DBF 库文件的字段名。字段值则位于该列的其它行, 如“西湖彩电”、“松下空调”等为商品名称字段的值。每行有这 8 个字段值构成一个记录, 每个记录前有一个记录号(由系统自动给定), 在 SP*.DBF 库文件中一条记录表示某个商品的编码、名称、产地、单价等信息。表中 10 条记录构成一个有关家电商品情况的数据库文件。

综上所述, 数据的组成层次如下:

1. 字段: 在数据库文件中最基本的单元为字段, 在设计库文件时不仅要为每个字段取字段

名,而且要定义它的类型和长度,字段的类型只有五种:数值型、字符型、日期型、逻辑型和备注型。

2. 记录:由若干相关联的字段组成,记录是对某一个具体实体的逻辑描述。

3. 文件:由若干条同类记录组成的信息集合。

4. 数据库:所有文件的集合。

FoxBASE+数据库是建立在关系模型基础上的,因而它们之间存在着对应关系。在关系模型中一张二维表即为一个关系,在数据库系统中即为一个文件,关系中的元组、属性和属性名分别对应于文件中的记录、字段和字段名。

1.4 FoxBASE+系统的运行

一、运行环境

1. 软件运行环境

在DOS环境下运行FoxBASE+,应使用汉字DOS2.0及以上版本,如在网络上运行则须使用汉字DOS3.1以上版本。

2. 硬件运行环境

(1)在单用户环境下,整机为IBM-PC/XT或IBM-PC/AT及其兼容机。至少有一个硬盘和一个软盘驱动器。有硬盘库机器的内存至少为512K字节,无硬盘库机器的内存至少为640K字节。

(2)在多用户环境下,整机可为386、486等及其兼容机。内存空间至少为1.5M字节。

二、FoxBASE+系统安装

前面所叙述的FoxBASE+系统文件是存放在多张软盘上,称为FoxBASE+系统盘。在安装FoxBASE+系统时,首先要在硬盘上建立一个子目录,然后将FoxBASE+系统盘复制到该子目录中。

三、FoxBASE+的启动与退出

1. FoxBASE+系统的引导

(1)首先引导汉字操作系统,默认盘为C盘,屏幕出现操作系统提示符C)。

(2)转到具有FoxBASE+系统的子目录下,例如该子目录为FOX,进入该子目录的方法是在提示符C)后输入“CD\FOX”,如下所示:

```
C)CD\FOX↵
```

```
C)
```

(3)在操作系统提示符下输入FOXPLUS(在多用户状态下则键入MFOXPLUS)按回车键后即进入FoxBASE+系统状态,并出现圆点“.”提示符。

```
C)FOXPLUS↵
```

现在可以使用FoxBASE+系统的各种命令和函数进行数据处理,如建立数据库、查询、显示和修改等处理。

2. 退出FoxBASE+系统

当完成各类数据处理后,需要退出FoxBASE+系统,只需要输入命令QUIT即可,如下

所示：

```
.QUIT ↵  
C)
```

这时屏幕上又出现操作系统提示符，回到操作系统状态。

系统执行 QUIT 命令时将把内存中的文件和数据送到磁盘上予以保存，任何非法的退出都可能造成数据的丢失。请务必注意要用 QUIT 命令正常退出，返回操作系统。

1.5 系统配置文件

系统配置文件是用于系统启动时配置资源和设定初始状态的文件。主要有 CONFIG · SYS(DOS 系统配置文件)和 CONFIG · FX/DB(FoxBASE+系统配置文件)。

一、DOS 系统配置文件

CONFIG · SYS 文件在启动 DOS 操作系统进行初始化设置时，是系统环境设置的基本依据，若该文件存在，则按该文件中所设置的各个参数对系统进行初始化，若该文件不存在，则按缺省值进行初始化。

CONFIG · SYS 文件的参数设置直接影响 FoxBASE+系统设置的有以下两个参数：

1. 缓冲区设置

格式： BUFFERS=<n>

该命令指出可设置的缓冲区个数为 n，n 取值为 1-99，若缺省此项设置，其默认值为 2。

缓冲区可加快 FoxBASE+的运行速度，但要占据内存空间，每个缓冲区占用 528 个字节内存。

2. 打开文件数设置

格式： FILES=<n>

该命令指出可同时打开的文件个数为 n，n 取值为 8-99，缺省此项设置，系统默认值为 8。可同时打开的文件个数中，其中 5 个文件为系统占用(标准输入、标准输出、标准错误输出、辅助设备和标准打印机)。因而用户只有使用 n-5 个文件。

在使用 FoxBASE+应用程序时，若屏幕出现“文件打开太多”的错误，这意味着 CONFIG · SYS 文件中 BUFFERS 和 FILES 参数设置不当，重新设置这两项参数有助于错误的排除。

二、FoxBASE+系统配置

CONFIG · FX/DB 文件是 FoxBASE+系统初始化时为系统设置参数的依据。CONFIG · FX/DB 文件所包含的内容较多，如设置缓冲区数、打开文件个数、允许内存变量个数、查定义功能键等，详见第十一章。它直接影响 FoxBASE+的运行性能。

当引导 FoxBASE+系统时，系统首先检查 CONFIG · FX 文件是否存在，若存在则按该文件的参数设置来设置初始状态，若不存在则检查 CONFIG · DB 文件是否存在，若存在则按该文件的参数设置来设置初始状态，若这两个文件都不存在，则按缺省值设置初始状态。

习 题

1. 简述数据管理技术发展的三个阶段。
2. 数据库系统通常有哪几部分组成？

3. 什么叫数据库？什么叫数据库管理系统？
4. 简述数据库的三种模型。
5. 简述关系模型的主要特点及其三种主要的关系运算。
6. 汉字 FoxBASE+ 的软件系统环境是什么？
7. 试分别说明字段、记录、库文件的含义。
8. 如何进入和退出 FoxBASE+ 系统。
9. 简述 CONFIG.SYS 文件的作用。

例如: " ' "、'She said; "I am a student. " ' 是合法的字符型常量; 'ABC'DEF'GHI' 是非法的字符型常量。

3. 逻辑型常量

逻辑型常量是用来作为各种逻辑判断的数据。它只有两个值: 逻辑真(TRUE)和逻辑假(FAULSE)。

逻辑真常量: 用 .T. , .t. , .Y. , .y. 表示。

逻辑假常量: 用 .F. , .f. , .N. , .n. 表示。

逻辑型数据用 L(Logical) 表示, 它有一个字节的固定长度。

4. 日期型常量

日期型常量是用来表示日期的常量。它有多种表示格式, 常用的有两种格式: 美国格式和欧洲格式。美国格式是 MM/DD/YY, 即月/日/年, 其中月、日、年都是两位数字, 之间用 "/" 分隔, 只写一位数字是错误的。欧洲(英国)格式是 DD/MM/YY 即日/月/年。通常采用美国格式, 可以通过 SET DATE 和 SET CENTURY 命令来改用其它格式。

日期型数据用 D(Date) 表示, 它有 8 个字节的固定长度。

二、变量

变量是指在程序执行过程中可以改变其值的数据项。每个变量都有一个变量名, 作为对其的唯一标识。FoxBASE+ 的变量可分为两大类: 一类是字段变量, 就是数据库中的字段; 另一类是内存变量, 它存在于内存中, 独立于数据库文件。

1. 字段变量

数据库文件中的字段是一个变量, 故称它为字段变量或字段名变量。FoxBASE+ 的字段变量有 5 种数据类型: 数值型、字符型、逻辑型、日期型、备注型, 字段变量的数据类型由建立库结构时决定。字段变量是多值变量, 数据库中有多少条记录, 字段变量就有多少个值(详见第三章)。

2. 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。它用来存储数据处理过程中所需要的常量、中间结果和最终结果, 实现与数据库文件的信息交换。因此, 内存变量是一种临时工作单元, 需要时可以临时定义, 不用时又可以随时释放。每个内存变量都是独立的, 都有一个唯一的内存变量名为其标识。

(1) 内存变量名

变量代表在程序执行过程中其值可以改变的存储单元, 这个存储单元的名字就是变量名, 它是变量的标识。变量名具有变量的值和变量的类型两种属性。

FoxBASE+ 规定变量名最多为 10 个字符, 它们可以由字母、汉字、数字、嵌在中间的下划线组成, 但必须以字母或汉字开头, 中间不能有空格。内存变量名的拼写没有限制。但若内存变量名与关键字(包括命令、函数)重名时, 会产生混乱。建议内存变量名不要与关键字重名, 例如:

X、SP01、NAME、XS __ RQ、商品名称等都是合法的内存变量; 而 1A、X-Y、"GZ"、姓名等都是非法的内存变量名。

(2) 内存变量的分类、类型及对它的简单操作

按独立程度来分, FoxBASE+ 的内存变量可分为简单内存变量和内存数组变量, 简单内

第二章 FoxBASE+ 应用基础

数据库管理系统处理的对象是数据。数据在 FoxBASE+ 系统中是以常量、变量、函数、表达式、文件等各种形式存在的。它们是应用 FoxBASE+ 的基础。在学习数据时,特别要注意它们的数据类型。因为,对数据操作的基本要求就是对相同类型的数据进行操作。

2.1 几个常用符号的约定

本文使用的几个常用符号,它们的含义约定如下:

(): 必选参数表示符。由这对符号括起来的部分用户必须要提供参数。

[]: 任选参数表示符。由这对符号括起来的部分由用户决定是否选择,可以选,也可以不选,所以又称为选择项或任选项。

/: 二选一表示符。表示用户可从本符左右两项中任选一项。

...: 表示同类项目的重复出现或省略了在当时叙述中的不涉及部分。

注意,这些专用符号和其中的提示,不是命令行和函数的组成部分。在输入具体命令时,上面的符号不能作为命令的一部分输入计算机,它们只是命令格式的书面表示。

本书中以一个圆点“·”开头的命令,表示在交互状态下执行;否则就是在程序(命令文件)中执行。

2.2 常量和变量

一、常量

常量是在程序执行过程中不变化的数据。FoxBASE+ 定义了四种类型的常量:数值型常量、字符型常量、逻辑型常量、日期型常量。

1. 数值型常量。数值型常量是可以进行数学运算的常数。它可以是整数和实数。实数有两种表示方法:一种是带小数点的实数;另一种是用科学记数法表示的实数,如 1.23E5,它一般用于数据位数较多的实数情况。数值型数据的数值精度为 16 位有效数字。数值型常数的长度包括整数位数、小数位数和小数点。数值型数据用 N(Numeric)表示。

2. 字符型常量。字符型常量是用单引号、双引号、方括号等定界符括起来的数据,又称为字符串(Character String)。它包括 52 个英文字母,10 个数码,各种符号和空格,即一切可打印的 ASCII 码字符,也可以是汉字。字符型常量的长度是定界符括起来的字符的个数,一个汉字相当于两个字符的长度,字符型数据的最大长度是 254。字符型数据用 C(Character)表示。例如 'A', "HANGZHOU", "123", [计算机] 是字符型常量; " " 表示三个空格的字符型常量; "" 表示长度为零的空字符串,请注意空串与空格串的区别。

使用字符型常量必须加定界符,而且定界符必须成对匹配。当字符串本身包含某种定界符时,则要选择另两种定界符作为该字符串的定界符,以示区别。

存变量简称为内存变量。

内存变量共有 5 种数据类型:数值型、字符型、逻辑型、日期型、屏幕型。

内存变量可通过各种赋值命令建立,这些命令同时完成变量的建立和赋值,并自动把该变量的类型定义为相应的数据类型。

内存数组变量只能是一维或二维的,一般通过数组定义命令 DIMENSION 建立,通过各种赋值命令可以对整个数组同时赋值,也可以对各单元独立赋值并把它们定义成相应的数据类型,同一数组中各单元的数据类型可以不同。

内存变量建立在内存中,退出 FoxBASE+ 系统,内存变量自动清除。可以用内存变量文件保存内存变量,也可以从该文件恢复到内存中。

内存变量的个数一般最多为 256 个,共占用 6K 字节。用户可以通过系统配置文件 CONFIG.FX 中的参数设置,使变量的个数增加到 3600 个,内存工作区域增加到 64K。

我们可以用“=”命令给内存变量赋值,用“?”命令输出内存变量的值。

例 2.1 简单的内存变量赋值和输出命令示例

. 姓名="李大名"

. 工资=290.24

. ? 姓名

李大名

. ? 姓名,工资

李大名 290.24

变量有“取之不尽”和“以新冲旧”的特点。所谓“取之不尽”,就是当变量具有某个值后,可以无限次的将这个值赋给其它变量,进行各种处理,不会因为把值赋给了别人而改变它本身的值。所谓“以新冲旧”,就是当给一个已有值的变量赋以新值时,变量原有的旧值也就被新值取代了。于是,随着程序和命令的执行,变量的值也随之变化。

三、数据分类

FoxBASE+ 的数据有常量、变量、函数 3 种,变量分为内存变量和字段变量,内存变量又分为简单变量和数组变量,所以 FoxBASE+ 的数据共有 5 种。

FoxBASE+ 的各种数据类型见表 2-1 (带 V 者表示有该类型的数据,带 X 者表示没有)

表 2-1

数 据 种 类		数值型	字符型	逻辑型	日期型	备注型	屏幕型
变 量	常 量	V	V	V	V	X	X
	字 段 变 量	V	V	V	V	V	X
	内 变 简单变量	V	V	V	V	X	V
	存 量 数组变量	V	V	V	V	X	V
函 数		V	V	V	V	X	X

2.3 表达式

表达式是用运算符将常量、变量和函数连接起来的有意义的式子。常量、变量和函数是

表达式的组成部分,作为一种特例,一个常量,一个函数,一个内存变量,一个字段变量都是简单的表达式。

根据表达式运算结果的数据类型,表达式分为 5 种类型:数值型、字符型、关系型、逻辑型、日期型。

一、数值型表达式

数值型表达式由数学运算符、数值型常数、内存变量、字段变量和函数组成,其运算结果为常数。

数学运算符有 6 种:

+: 加法(单目运算时为正)

-: 减法(单目运算时为负)

*: 乘法

/: 除法

** 或 ^: 乘方

(): 括号

其优先顺序从高到低为:

括 号 — 单目运算 — 乘 方 — 乘 除 — 加 法

对于同级运算,按自左向右的规则进行运算。在书写 FoxBASE+ 表达式时,应注意以下几点:

在 FoxBASE+ 表达式中,每个符号占一格,所有符号必须写在同一行上,没有大小、上下之分。例如:3 的平方写成 3^2。

原来在数学表达式中省略的或改写成“.”的乘号,必须用“*”重新写上。例如:3X 必须写成 3 * X。

所有括号都用小括号(),括号必须配对。例如:4CX+5(X+1)) 必须写成 4 * (X+5 * (X+1))。

二、字符型表达式

字符型表达式由字符运算符、字符型常数、内存变量、字段变量和函数组成,其运算结果为字符串。

字符运算符只有 2 种:

+: 字符串连接符,其功能是将运算符两边的字符串连接起来,形成一个新的字符串。

-: 串尾空格移位连接符,在将两个字符串连接时,把前面字符串的尾部空格移到后面字符串的尾部。

字符型运算只有一种优先级,即按自左向右的规则进行运算。

例 2.2 字符型表达式运算示例

(1).?“ABC”+?“DEF”

ABC DEF

(2).?“ABC”-?“DEF”

ABCDEF

(3).X=“03”

.“PAY”+X

PAY03

参与字符型表达式运算的常量、变量、函数必须是字符型，不是字符型的数据可先通过转换函数将其转换为字符型，否则会产生“数据类型不一致”的错误。例如：

.X=3

.? X+"月"

Operator/operand type mismatch &&. 数据类型不一致，产生出错信息。

三、关系型表达式

关系型表达式由关系运算符、数值表达式、字符表达式，或表达式的特例，常量、变量和函数组成。它们可以是数值型、字符型、日期型、逻辑型数据，但关系运算符两侧的数据类型必须一致。

关系型表达式的格式为：

〈表达式 1〉〈关系运算符〉〈表达式 2〉

关系运算符有 8 种：

<： 小于

>： 大于

<=： 小于等于

>=： 大于等于

=： 等于

<>或#： 不等于

\$： 字符串包含

=，： 字符串精确比较

关系型运算符只有一种优先级，即按自左向右的规则进行运算。具体运算时，先计算关系运算符两侧表达式的值，然后再进行关系运算。

关系运算符用于对两个相同类型的数据的值进行大小比较，数值型数据按其值的大小进行比较；字符型数据按它的 ASCII 码值的大小进行比较；汉字按它的内码进行比较。关系运算的结果为逻辑型数据，其结果只能为逻辑真（T.）或逻辑假（F.）。

例如：

关系表达式 $3 > 2$ ，“A”<“a”， $6 \geq 6$ ， $100 \neq 101$ 的值为 T.；

关系表达式“ABC”>“ABD”，“计算机”=“计算器”， $2 > 3 + 4$ 的值均为 F.。

“=”为字符串精确比较运算符，在 SET EXACT OFF 状态下，其功能相当于关系运算符“=”，在 SET EXACT ON 状态下，尾部空格参加比较。系统默认为 SET EXACT OFF。

例 2.3 关系运算符“=”运算示例。

在 SET EXACT OFF 状态下，

关系表达式“DEF”=“DEF”与“DEF”=“DEF”的值均为 F.；

在 SET EXACT ON 状态下。

关系表达式“DEF”=“DEF”的值为 T.；

关系表达式“DEF”=“DEF”的值为 F.。

“\$”为字符串包含运算符，其格式为：

“〈字符串 1〉\$”〈字符串 2〉”

其功能为:如字符串 1 包含在字符串 2 中,表达式的值为真(.T.),否则为假(.F.)。

例 2.4 关系运算符“\$”运算示例。

.?“彩电”\$“西湖彩电”

.T.

.?“杭州”\$“浙江”

.F.

.?“ABCD”\$“AB”

.F.

从上述例子中可以看到,关系运算符“\$”处理的对象是字符型数据,而运算的结果却是逻辑型数据。应用这个原理,可用来进行子串寻找,当关系表达式的值为“.T.”时,子串存在;当关系表达式的值为“.F.”时,子串不存在。这在信息的模糊查询时是非常有用的。

四、逻辑型表达式

逻辑表达式由关系表达式、逻辑运算符、逻辑常量、内存变量、字段变量和函数组成。其一般格式为:

〈关系表达式〉〈逻辑运算符〉〈关系表达式〉

其功能为:通过逻辑运算符对逻辑型数据进行运算,其运算的结果仍为逻辑型常量(.T.、.F.)。

逻辑运算符有 3 种:

.NOT. 或 !:逻辑非

.AND.:逻辑与

.OR.:逻辑或

注意: 逻辑运算符两边的圆点不能省去。

逻辑运算的优先级为:

逻辑非——**逻辑与**——**逻辑或**

逻辑表达式运算时,先运算关系表达式,再运算逻辑表达式。

例 2.5 设逻辑型变量 LOG1,LOG2 的值分别为 .F.、.T.,且 I,J,K 分别取值 -4,6,8,则下列逻辑表达式运算结果如下:

.?“LOG1.OR.LOG2”

.T.

.?“K>J.AND.J<8”

.T.

.?“NOT.(I<8.OR.K=8)”

.F.

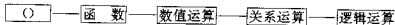
逻辑型表达式和关系表达式的值均为逻辑值,可以把它们作为“条件”,用在命令的“FOR”或“WHILE”的后面,作为限制特殊处理的范围和定义数据筛选的条件;也可以作为条件分支语句和循环语句的条件,控制程序的流程。

五、日期型表达式

日期型表达式由算术运算符(+或-)、算术表达式、日期型常量、内存变量、字段变量和函数组成。日期型数据是一种特殊的数据,它们之间只能进行减(-)运算、运算的结果是算术值。

日期型数据与表示天数的数值数据可以进行(+)、减(-)运算,运算的结果仍为日期型数据。两个日期型数据不能相加。

在表达式中,各种运算的优先级是:



2.4 基本函数

函数是除常量、变量以外的一种本身具有运算功能的特殊的数据。每取一次函数的值,就是进行一次特定的操作。FoxBASE+有100余种函数,因此,它有100余个不同于命令的特殊功能。

函数基本形式为:函数名(<操作数表达式>)

由于函数是一种操作功能,它也就有操作数。函数的操作数有三种情况:有一部分函数的操作数由系统规定,用户不用输入,此时函数的形式为:函数名();还有一部分函数的操作数必须由用户输入;另有部分函数的操作数可以由用户指定,如用户不指定,系统就按有关规定来给出。对于后两种情况,输入的操作数就相当于一般函数的自变量,FoxBASE+对每个函数自变量的个数、类型、值域都有明确的规定,在调用函数时必须遵守这些规定。由于函数可以用表达式做自变量,如该表达式中又含有函数,便可实现函数的嵌套使用。

函数是一种数据,它输出的值有四种数据类型:数值型、字符型、日期型、逻辑型。在学习函数时,要特别注意它输入输出的数据类型。

FoxBASE+标准函数按功能和用途可分为6大类:数学运算函数、字符运算函数、日期时间函数、类型转换函数、数据库函数、测试函数。FoxBASE+还允许用户自定义函数(UDF)。这一节只介绍常用的基本函数,其它函数请参见附录。

一、数学运算函数

数学运算函数的输入输出数据类型均为数值型。

1. ABS——绝对值函数

格式: ABS(<数值表达式>)

该函数用来计算数值表达式的绝对值。例如:

.? ABS(30-70)

40

2. SQRT——平方根函数

格式: SQRT(<数值表达式>)

该函数用来计算数值表达式的平方根。(<数值表达式>的值必须大于等于0。例如:

.? SQRT(ABS(-225))

15

3. EXP——自然指数函数

格式: EXP(<数值表达式>)

该函数将数值表达式的值作为指数X,求出 E^X 的值。例如:

.? EXP(1.0000)

2.7183