

第一章 汉字 FoxBASE+ 基础知识

第一节 数据库系统的基本概念

一、数据处理及其发展

1. 数据与信息

数据是一串物理符号序列,用来记录或标记事物的物理状况。通常,数据可分为两类:表示工资、奖金和物价等多少的数据,称为数值型数据;表示人名、地名和物品名称的数据,称为字符型数据。

信息是事物状态及其运动状态的表现形式。通俗地讲,信息是经过加工处理并对人类社会实践和生产活动产生决策影响的数据表现形式。一种新思想、一个新主意、一种办法以及各种策略和计谋(信息),是依据客观情况(广义的数据),经过人们紧张思考和反复研究(加工处理)而产生的。只有经过去粗取精、去伪存真的加工处理,原始数据才发生了质的变化,才能给人们以智慧的火花和新的知识的启迪。

信息和数据在概念上既有联系,又有区别。不经过加工的数据只是一种原始材料,或是一堆死材料,不能给人以知识和智慧,影响不了人类的活动,它的价值只是在于记录了客观世界的事实。只有经过提炼和加工,被赋予新的知识的数据(已经发生了质的变化)才能成为信息。因此,信息来源于数据,是对数据进行加工处理的产物,它的价值在于对人类认识世界和改造世界活动的现实意义。然而,经过加工处理之后所得到的信息,仍以数据的形式出现,此时的数据是信息的载体,是人们认识和利用信息的一种媒介。

2. 数据处理

在人类的社会(政治的、经济的和文化的)活动中,信息总居于非常重要的地位。可以毫不夸张地说,人类的全部活动都是在信息的支配和指导下进行的。因而人类社会机体中有一类部门是专门从事数据处理的,如统计部门、政府部门、银行、邮局,……,它们要花费大量的人力、物力和时间去从事数据的收集、综合、加工,最后获得各种有益的信息,作为指导活动的依据,形成一个围绕信息的“数据-信息-决策-活动”的信息反馈周期(见图 1-1)。在社会信息化的今天,有意识地、准确地利用这种规律将有利于使自己立于不败之地。

在这个周期中,信息是重要的,而数据和对数据的处理是它的基础。所谓数据处理则是指对“原料”数据的综合加工以导出“产品”数据(信息)的过程。这一过程包括对数据的收集、存储、分类、计算、加工、检索和传输等若干处理阶段。

人类处理数据是社会活动的需要和社会信息化的必然。实际上,数据处理这一特殊的社会

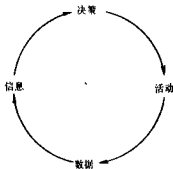


图 1-1 信息反馈周期

活动很早就开始,其历史可以追溯到远古时代。自有人类出现就开始了数据处理活动。原始人类的结绳记事,垒石计数便是最早的数据处理形式和借助的“工具”。随着生产力的发展和文明的进步,处理工具的发明和利用,数据处理方法也得到了相应的发展。就其整个发展过程来看,经历了手工数据处理、机械数据处理和电子数据处理等三个主要阶段。电子数据处理阶段的主要特征是电子计算机的发明和应用。它使数据处理实现了自动化。

3. 数据处理的发展阶段

在电子数据处理方式下,利用计算机系统作为处理工具,通常是把“原料”数据和对数据的处理方法或过程的描述(算法)输入系统,然后由系统自动地执行处理任务,最后输出处理结果。随着计算机技术的发展,数据处理技术经历了三个阶段。

(1) 程序管理方式

这是计算机用于数据处理的初期阶段,对数据的管理是由程序员个人考虑和安排的。他们把数据处理纳入程序设计的过程中,程序员除了编制他的课题程序外,还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。严格说来,这种管理只是一种技巧,这是数据自由管理的方式。其特点是:数据不长期保存,没有软件系统对数据进行管理,基本上没有文件概念,数据附属应用程序,本身不具有独立性而且大量的数据冗余。

(2) 文件系统方式

当计算机操作系统包含有文件系统后,把数据组织成文件的形式就使得计算机数据管理方法得到了极大的改善。我们知道,一大堆杂乱数据是毫无意义的,也没有实用价值。数据的文件组织方式就是把数据按一定的规则组织起来,成为一个有效的数据组合体。文件中的数据是以“记录”的形式存放,记录是由某些相关数据项组成,若干个具有相同性质的记录的集合构成文件。文件可以按照不同的组织方法分为顺序文件、索引文件、倒排文件等等。每一个用户都可以建立、维护和处理一个或某几个文件,每个文件都有特定的文件名或文件标识存储在外部存储介质上。数据被组织成文件之后,就可以离开处理它的程序而独立存在,以后用户就可以在程序中按这个文件标识引用其中的数据。所有文件由一个称作文件管理系统的专用软件对其进行管理和维护。

文件管理系统是应用程序和数据文件之间的一个接口。应用程序通过文件管理系统建立和存储文件。反之,应用程序要存取文件中的数据时也需要通过文件管理系统来实现。目前,文件管理系统仍然是一种较为广泛实用的数据管理方式。它使得用户在程序设计时,只考虑到数据的逻辑定义和物理特征,按规定的组织方式建立文件和按规定的存取方法使用文件,较少地考虑到数据物理存储方面的问题。

文件系统的方法比起程序管理方式虽然有了很大的改进,但这种方法仍然存在许多弱点,即文件本身还是基本上对应于一个或几个应用程序,尽管程序不必要直接与文件打交道而有文件管理系统作为接口,但它仍然是一个不具有弹性的无结构的信息集合,存在着冗余量大、空间浪费、文件不易扩充等缺点,还不能充分反映现实世界事物之间广泛的内在联系。

(3) 数据库管理方式

数据库方式是在文件系统基础上发展起来的最新技术。数据库的实质是一个所有存储在计算机存储介质上的相关数据所构成的集合。其基本思想就是要对所有的数据实行统一的、集中的、独立的管理,数据独立于程序而存在并且可以提供给各类不同用户共享使用。

数据库系统是一种有组织地、动态地存储有密切联系的数据集合,并对其进行统一管理的

计算机软件以及硬件所组成的系统。数据库系统由数据库管理系统、存储于存储介质上的数据和应用程序组成。

数据库管理系统是数据库系统的核心部分,它提供了对数据库中的数据资源进行统一管理和控制的功能,是用户程序和数据库中的数据间的接口,它由一系列软件组成。为微机配置的这种系统最为流行和通用的就是 FoxBASE+ 和 dBASE 数据库管理系统。

数据库系统的数据是存储在外部介质上的相关数据集合,是构成数据库系统的重要部分。通过设计数据存储结构,可以将数据按一定的规则和方式存放这些数据,以便迅速地而方便地读取这些数据。FoxBASE+ 数据库管理系统提供了数据存放和读取的有效手段。

应用程序由用户编写,用来调用数据库中所存储的数据的各种命令文件。FoxBASE+ 数据库管理系统自备有编写应用程序的语言,它提供了一系列与高级语言中各种语言相类似的命令,用户可以直接使用这些命令编写应用程序。

二、数据模型

数据库系统研究的对象是客观事物以及反映这些客观事物间相互联系的数据。各个数据对象以及它们之间存在的相互关系的集合,称为数据模型。根据数据所描述的实体对象的属性特征及人们所采用的描述方法,数据模型可分为三大类:数据的层次模型、数据的网状模型和数据的关系模型。

1. 层次模型

在这种模型中,每个数据元素可以与下而任何一层的多个数据元素相联系,但只能与它上而一层中的一个数据元素相联系。最高一层的数据元素称为根元素,根元素是唯一地而且只有它有下属元素。层次模型结构就象由根元素出发向下逐层辐射枝叶的一棵倒放的树。树的结点表示数据元素,枝表示联系。图 1-2 给出层次模型的一个例子。

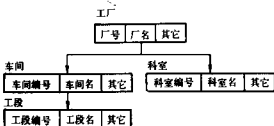


图 1-2 数据的层次模型

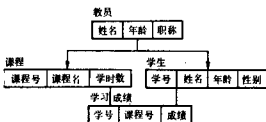


图 1-3 数据的网状模型

层次模型可以表示许多事实。如一个单位的行政组织关系,生产组织关系,一个家属的世代关系都呈层次模型结构。但是,由于层次模型结构的严格性,它不能直接地表示所有事实。更由于其固有的缺陷,对于数据存储和操作都不是理想的。

2. 网状模型

数据网状模型的基本结构是图。节点表示数据元素;边表示数据元素间的联系,在网状模型中,可以有一个以上的数据元素没有上层联系。因此,网状模型有更一般的表示事实的能力。如图 1-3 所示教师、学生和课程之间的关系是层次模型不能表示的。因为图中的每个节点都与其它两个节点有联系。

3. 关系模型

在关系模型中,不同数据元素之间的联系是用关系来表示的。其实质是将数据的逻辑结构归结为满足一定条件的二维表的形式,每一个二维表称为一个关系,如表 1-1 表示了每个职工本身的简单情况的关系模型结构。

表 1-1 职工关系

Record #	职工号	姓名	性别	出生时间	工作时间	职称	婚否	工资
1	1001	陈明明	男	05/12/27	10/01/49	高工	.T.	210.50
2	1002	吴梅	女	10/05/38	08/15/60	高工	.T.	186.00
3	1003	李小军	男	12/30/38	06/01/65	高工	.T.	182.00
...	...	...	...	...	...	...	...	...

关系模型的主要特点表现在它的数据描述的统一性,即描述的对象间的联系等均只能用关系来表示。由此可见,关系模型的结构十分简单,易被一般人所接受,且又有很强的表示能力。关系模型有其严格的数学基础,数据的各种处理是以集合代数为基础的。因此,它是目前比较流行的一种数据模型。

以数据的层次模型为基础设计的数据库系统称为层次数据库,以数据的网状模型为基础设计的数据库系统称为网状数据库。这两种数据库系统在七十年代出现较多,目前仍有不少中小型机使用这类结构的数据库。同样,以数据的关系模型为基础设计的数据库系统称为关系数据库,它是目前世界上比较流行的数据库管理系统。

三、关系型数据库系统

关系数据库系统把数据间的组织和联系表示成一张二维关系表格(如表 1-1),用这种关系模型设计关系数据库系统。显然,有了这样一个关系数据库,属于以下种类的用户询问就可以得到回答。

- (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:如某职工的职称是什么?月工资是多少?
- (3) 指出有某种属性的对象:如所有女职工都是谁?有工程师职称的人都是谁等等。
- (4) 满足某种条件的对象的属性情况:如职称是工程师的姓名、工作时间、工资等。
- (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)某数额的人数有多少等。

在 FoxBASE+ 关系数据库管理系统中,按照关系模型,数据的组成可分为四个层次。

(1) 字段 (FIELD)

字段是关系数据库文件中最基本的不可分割的数据单位。它用来描述某个实体对象的属性,相当于二维表格中的一列,且都有相同的数据类型。一般,我们就用属性名作为字段名。

(2) 记录 (RECORD)

记录是描述一个个体的数据集合。记录由若干个字段值组成,相当于二维表中的一行。组成记录的所有字段的总长度就是记录的长度。

(3) 文件 (FILE)

文件也称库文件,是具有相同性质的记录的集合。它相当于一张二维关系表。文件中的记录必定是同格式、等长度的,使用中不应有无法区别的两个数据完全相同的记录存在。一般,我

们用关系名作为文件名。

(4) 数据库(DATA BASE)

描述实体集的所有文件的集合称为数据库,它由若干个二维表的数据文件组成。数据库不是某一方面数据文件的简单集合,它按照一定法则对文件进行重新组织,以便使数据具有最大的独立性和最小冗余度,并实现对数据的共享。

第二节 汉字 FoxBASE+ 的主要特点与性能指标

FoxBASE+ 系统是美国 FOX 公司 1986 年中期开发成功的多用户关系数据库管理系统,1987 年 6 月由美国 SCO 公司发行。FoxBASE+ 系统是 dBASE 系统的继承和发展。在我国,各种汉字 FoxBASE+ 系统已广为流行。

一、FoxBASE+ 的主要特点

(1) FoxBASE+ 和 dBASE III plus 完全兼容,通过系统提供的命令,可将 dBASE 环境中的程序和数据很容易地转到 FoxBASE+ 环境中运行。

(2) 具有很快的运行速度,是现有各种 dBASE 及其兼容软件中速度最快的一种,平均比 dBASE III、dBASE III plus 快 6~7 倍,比编译 dBASE III 快 2 倍左右。

(3) FoxBASE+ 用 C 语言开发,易于移植和推广。FoxBASE+ 可运行于多个操作系统(如 MS-DOS、UNIX)及多种机器环境(IBM-PC/XT,VAX11/7800),多用户版本与单用户版本完全兼容,用户软件随着硬件和系统软件的发展很容易向高档机环境移植。

(4) 提供多种运行方式,即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编译执行。

(5) FoxBASE+ 可以定义数组,实现数组与文件记录的数据交换;可以自定义函数;可以保存和恢复屏幕映象。

(6) FoxBASE+ 在运行中自动地动态调整缓冲区、程序存储区及其它资源在内存中的位置,从而能充分使用当前机器所配置的内存资源。

二、FoxBASE+ 的主要技术指标

每个数据文件记录个数:	最多 10 亿个
每个记录字节个数:	最多 4000 个字符
每个记录字段个数:	最多 128 个
数值字段:	最多 19 个字节
字符字段:	最多 256 个字节
日期字段:	8 个字节
逻辑字段:	1 个字节
备份字段:	最多 64K 字节
内存变量个数:	最多 3600 个,默认 256 个
字符串变量可用字节数:	最多 64K 字节,默认 6K 字节
其它变量可用字节数:	受剩余内存大小限制
数组个数:	最多 3600 个

每个数组元素个数:	最多 3600 个
数值精度:	16 位有效数字
每个命令行长度:	最多 254 个字节
每个索引码长度:	最多 100 个字节
可同时打开的文件数:	最多 48 个, 默认 16 个
可同时打开的数据库文件:	最多 10 个
可同时打开的索引文件:	最多 21 个
每个数据文件可同时打开索引文件:	最多 7 个
DO 命令嵌套层数:	最多 24 层
程序结构嵌套层数:	IF、WHILE、CASE 等最多 64 层
过程文件中子过程个数:	最多 128 个

第三节 汉字 FoxBASE+ 的环境与启动

一、硬件和软件环境

1. 硬件环境

汉字 FoxBASE+ 适用于 IBM-PC/XT、AT、286、386 等微型计算机以及它们的兼容机种, 例如长城 0520CH、浪潮 0520D、浪潮 286、长城 286 等。一般来说, 其最少硬件要求是:

- (1) 至少 375K 的内存空间,
- (2) 至少两个磁盘驱动器(一个硬盘、一个软盘)。

2. 软件环境

汉字 FoxBASE+ 需在 MS/PC-DOS、OS-2 等操作系统和 CCDOS、ACBIOS、ECBIOS、长城汉卡、东海汉卡、联想汉卡等多种中西文环境中运行。通常在 CCDOS、XSDOS 等中西文操作系统支持下运行。

二、汉字 FoxBASE+ 的基本组成

汉字 FoxBASE+ 主要组成部分是执行程序(MFoxPLUS. EXE)和覆盖程序(MFoxPLUS. OVL)。执行程序启动 FoxBASE+ 后驻留内存, 覆盖程序在 FoxBASE+ 启动时只有一部分装入内存, 其余的在需要时以覆盖方式调入内存, 使用后又被新的内容所覆盖。

汉字 FoxBASE+ 的辅助部分有帮助文件(FoxPHELP. HLP)、过程组合文件(FoxBIND. EXE)、准编译程序(FoxPCOMP. EXE)、欧洲文字排序基准文件(EUROPEAN. MEM)等。帮助文件用于向用户提供有关 FoxBASE+ 的语法和用法等说明信息, 过程组合程序用于把若干个命令文件组合成过程文件, 准编译程序用于编译 FoxBASE+ 的命令文件, 欧洲文字排序基准文件用于解决法、德、意等欧洲各国字母排序问题。

三、汉字 FoxBASE+ 的启动

汉字 FoxBASE+ 是由西文版 FoxBASE+ 的多用户版本 MFoxBASE+ 汉化开发而成的, 全部软件包括 2 张 5.25" 软磁盘, 内容分别如下:

- 1# 软盘 MFoxPLUS. EXE

FoxPCOMP. EXE

FoxBIND. EXE

2# 软盘 MFoxPLUS. OVL

FoxPHELP. HLP

EUROPEAN. MEM

汉字 FoxBASE+ 的启动步骤如下:

(1) 启动中西文操作系统(如 CCDOS);

(2) 将装有汉字 FoxBASE+ 的 1#、2# 软盘的文件复制到硬盘上(若已装入硬盘,此步骤可省略);

(3) 在当前 FoxBASE+ 目录下,执行如下命令:

C>MFoxPLUS

进入汉字 FoxBASE+ 交互执行方式,出现汉字 FoxBASE+ 提示符“·”,等待用户运行汉字 FoxBASE+ 的数据库管理系统。

数据库操作完成后,在交互方式下返回到“·”提示符。为退出汉字 FoxBASE+ 系统而返回操作系统,须执行如下命令:

·QUIT

第四节 汉字 FoxBASE+ 的基本语言元素

一、数据类型

数据是信息的表示形式。数据库中存储着各种各样用户感兴趣的数据。

数据有许多重要的属性,数据类型则是数据的基本属性。数据类型是数据库系统中十分重要的概念,因为只有相同类型的数据之间才能进行运算。这是数据库操作首先必须遵守的原则,违反了这条原则,就会发生错误。

汉字 FoxBASE+ 定义了六种数据类型。

(1) 字符型:用 C (CHARACTER) 表示,由 26 个英文字母、数字、空格和各种符号组成。

(2) 数值型:用 N (NUMERIC) 表示,只能由数字、小数点和正负号组成。N 型数据是可以进行算术运算的数据。

(3) 逻辑型:用 L (LOGICAL) 表示,它只允许两个逻辑值“真”(TURE)和“假”(FALSE),或“是”(YES)和“非”(NO),L 型数据是用来进行各种逻辑判断用的。

(4) 日期型:用 D (DATE) 表示,它是用来表示日期的。D 型数据具有固定的长度,它只占 8 位存储空间。美国格式为 mm/dd/yy,而西欧格式为 dd/mm/yy。

(5) 备注型:用 M (MEMORY) 表示,它只能用来定义数据的类型。它具有 10 个字符的固定长度。

(6) 屏幕型:用 S (SCREEN) 表示,它只用于保存屏幕显示信息。

二、常数

常数是指在程序运行过程中不变化的数据。在汉字 FoxBASE+ 中常数为四种类型。

(1) 字符型常数:用定界符括起来的字符数字串。定界符规定为三种:单引号“'”,双引号

“n”和方括号“[]”。如果某一种定界符本身是字符型常数的组成部分,就应选择另一种定界符。

(2)数值型常数:可以是整数或实数。

(3)逻辑型常数:逻辑值“真”用 .T.、t.、Y.、y. 表示,逻辑值“假”用 .F.、f.、N.、n. 表示。

(4)日期型常数:用月/日/年来表示。这里的月、日、年分别要求用两位数字表示,如:1989年12月29日表示为12/29/89。

三、变量

变量是指在程序运行过程中其值是可以变化的量。汉字 FoxBASE+ 中有两种形式的变量,一种是字段名变量,另一种是内存变量。

1. 字段名变量

字段名变量是用户所定义的数据库结构中的任一数据项,是一种多值变量。如果一个数据库有25条记录,那么该数据库的每一字段就有25个值,当用这个字段名作变量时,它的当前值就是数据库的指针所指的那条记录的对应字段的值。数据库文件有一个记录指针。记录指针是随着数据库的操作不断移动的,因此字段名的取值也是随着记录指针的移动而不断变化的,所以它是变量。

字段名变量的数据类型共分为五类,它们分别是C、N、D、L和M型数据。字段名变量的数据类型是在定义数据库结构时确定的,变量名是以不超过10个字母(5个汉字)、数字或下划线组成的字符串。

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数、中间结果和最终结果。因此,内存变量是一种临时工作单元,需要时临时定义,不需要时可以随时释放。内存变量可以参与计算处理,也可以作为控制变量控制程序的运行。

内存变量共分为五种类型,即C、N、D、L、S。

内存变量中存放什么类型数据(C、N、D、L、S),该变量就具有相应的属性。内存变量名是以字母、数字、汉字或下划线组成的字符串,串长最多允许有10个字符(5个汉字),变量名必须以字母或汉字开头,下划线必须嵌在变量名中间,但不能嵌入空格。汉字 FoxBASE+ 在系统中规定内存变量不超过3600个,一个变量最多只能存放254个字符,所有内存变量占用的存储空间约为64K字节。

给内存变量赋值命令有“=”、STORE、INPUT、WAIT、ACCEPT等,详见第四章。

四、函数

除了常数、变量之外,另一种数据就是函数。汉字 FoxBASE+ 的函数概念和数学中的函数概念是一致的。汉字 FoxBASE+ 除了向用户提供了78个函数外,还提供了自定义函数。在使用函数时应注意以下几个问题:

- (1) 每个函数必然有一个返回值;
- (2) 返回值有确定的数据类型;
- (3) 函数可以和其它数据进行操作运算;
- (4) 函数要求的参数也有一定的数据类型,参数类型不匹配时,将会出现语法错误。

汉字 FoxBASE+ 系统中的函数格式及功能详见第五章。

五、运算符

汉字 FoxBASE+ 定义了算术型、关系型、逻辑型和字符型四种运算符。

(1) 算术运算符：可以对数值型数据进行常规的算术运算并产生结果。

算术运算符分别为：

+	加法运算	-	减法运算
*	乘法运算	/	除法运算
**	或		乘方运算
()			优先运算符

(2) 关系运算符：用来比较并产生逻辑结果(真或假)。关系运算符用于字符、数值或数值表达式(两个表达式的类型要相同)的比较运算。

关系运算符分别为：

<	小于	<>	不等于
>	大于	<=	小于等于
=	等于	>=	大于等于

(3) 逻辑运算符：用来对逻辑型数据进行各种逻辑运算，形成各种简单的逻辑结果。

逻辑运算符分别为：

()	分组括号	.NOT.	逻辑非
.AND.	逻辑与	.OR.	逻辑或

(4) 字符串运算符：用于字符串的连接或者比较。

字符串运算符分别为：

\$：用于表示两个字符串之间的包含与被包含的关系，运算的数据只能是字符型的。
格式为：(子字符串) \$ (字符串)

如果(子字符串)被包含在(字符串)中时其结果为“真”，否则为“假”。

十：字符串连接符，用于把两个或两个以上字符串连接成一个字符串。

一：空格压缩连接符，它可以把第一个字符串尾部的空格去掉，然后与第二个字符串连接形成一个新的字符串。

各种运算符的优先关系是：

每一类型的运算符都有一组控制运算执行顺序的规则。在这些规则中规定了各种运算符的优先级别。关系运算符和字符串运算符具有相同的优先级别，其执行的顺序为从左到右。

算术运算符的优先级别和一般算术运算规则完全相同，即先乘方后乘除，最后加減。圆括号的优先级别最高，遇到同样级别的运算时，从左至右依次运算。

逻辑运算符的优先级别顺序是：逻辑非(.NOT.)、逻辑与(.AND.)、逻辑或(.OR.)，逻辑运算符和算术运算符一样也可以利用优先级运算符括号来改变它们之间操作运算的先后顺序。

当同一表达式中使用了几种运算符时，类型的优先级别顺序依次是算术型、字符型、关系型、逻辑型，同一优先级别中的全部运算为从左到右顺序执行。只有在使用了圆括号的情况下才能改变运算的顺序。

六、表达式

表达式是由同类型的各种数据,如常数、变量、函数通过各种运算符联结起来,具有一定意义的式子。表达式经过各种运算后总能得到一定的运算结果,称为表达式的值。

根据表达式运算结果的数据类型的不同,表达式分为以下四种。

(1)数值表达式:由算术运算符、数值型常数、变量和函数组成,如: $55/89 * (194-82)$ 。其运算结果是数值型数据。

(2)字符表达式:由字符运算符、字符型常数、变量和函数组成,如: "I am" + "a" + "student"。其运算结果是字符型数据。

(3)关系表达式:由关系运算符与字符表达式或数值表达式组成。其运算结果是一个逻辑值,关系表达式两边的数据类型要一致。字符的比较是根据 ASCII 码值的大小进行的,汉字是按机内码值进行比较的。如果含有表达式,则先计算表达式的值,再进行关系的比较。关系成立值为 .T.,否则值为 .F.。例如:

• ? "ABC" > "DEF"

.F.

• ? SQRT(3²-4*1*2) >= 0

.T.

(4)逻辑表达式:由逻辑运算符、逻辑型常数、变量和函数组成,例如:

性别 = "男".AND. 职称 = "高工"

"ABC" <= "XYZ".OR. 123 >= 456

逻辑表达式实际是一种判断条件,表达式值为 .T. 则表示条件成立,表达式值为 .F. 则表示条件不成立。

七、文件类型

汉字 FoxBASE+ 系统以不同的格式将各种信息存储在磁盘文件中,以满足不同处理的需要。文件格式和处理方式的不同,形成了不同的文件类型。

在汉字 FoxBASE+ 系统中文件的类型是以扩展名来区分的,共定义了 11 种类型的文件。其中,命令文件(.PRG)、内存文件(.MEM)、文本文件(.TXT)是独立的,而其它各类文件是由库文件(.DBF)派生出来的。

各类文件按其文件的结构特征、操作特点,又可分为两大类:

结构文件

库文件	.DBF
内存文件	.MEM
索引文件	.IDX
备注文件	.DBT
报表格式文件	.FRM
标签文件	.LBL

非结构文件

命令文件(未编译)	.PRG
命令文件(已编译)	.FOX

文本文件	.TXT
格式文件(未编译)	.FMT
格式文件(已编译)	.FMX

结构文件和非结构文件二者的建立、修改、操作和显示方式是不一样的。例如：我们可以在 CCDOS 下用 TYPE 命令显示命令文件的内容，但不能显示库文件的内容。因为命令文件是非结构式文件，即 ASCII 码文件，可以用 TYPE 命令显示；而库文件是结构文件，只能用专门的命令对它进行操作。所以，在对文件进行操作时，一定要注意这两种情况的差别。

其次，汉字 FoxBASE+ 对命令文件的编译是一种伪编译（或称准编译），编译后产生的特殊代码文件不能独立运行，还得由 FoxBASE+ 解释执行（即 Runtime 版本）。

第五节 汉字 FoxBASE+ 的命令结构和运行方式

一、命令的结构和书写规则

命令的结构称之为语句。汉字 FoxBASE+ 的命令又被称为语句，它是充分吸取多种高级语言的优点逐步发展形成的。汉字 FoxBASE+ 共有一百多个命令。从功能上看，这些命令可以分为：

- 数据库库文件操作命令
- 数据库数据操作命令
- 输入和输出操作命令
- 内存变量操作命令
- 命令文件的建立、控制和运行命令
- 系统参数设置命令

等几大类。

1. 命令结构

汉字 FoxBASE+ 命令通常是由两部分组成的。第一部分是命令动词，它的词义指明了该命令的功能。第二部分包含有几个跟随在命令动词后面的选择项，这些选择项通常用来对所执行的命令进行某些限制性的说明。

在汉字 FoxBASE+ 命令中，有许多是对库文件记录进行操作的命令。其基本结构格式如下：

命令动词 [(<范围>)] [FIELDS (<字段名表>)] [FOR (<条件>)] [WHILE (<条件>)]

2. 命令格式中符号的约定

命令动词：是一个英文词，它表示这个命令所要完成的操作。为了方便，当命令动词超过四个字母时，可以采用简写，只写前面四个字母，例如：

DISP 与 DISPALY

是一样的。

[]：可选参数表示符。在键入命令时，表示括号内的参数可以键入，也可以不键入，但都不影响命令本身的功能。不键入时则默认为系统的缺省值。

()：必选参数表示符。由用户根据问题的需要提供具体参数。如果缺少必选参数，命令则发生语法错误。但是，在书写具体命令或键入时，不能有方括号和尖括号。

〔范围〕:表示本命令对库文件进行操作的记录范围,一般有四种选择:

(1) ALL:表示对库文件的全部记录进行操作。

(2) NEXT n:表示只对包括当前记录在内的以下 n 条记录进行操作。

(3) RECORD n:表示仅对第 n 个记录进行操作。

(4) REST:表示记录范围是从当前记录到文件的最后一个记录。

如果省略〔范围〕这一项,一般情况下意义与选择 ALL 相同。

FIELDS (字段名表),这个选择项是选取参与操作的数据库字段。FIELDS 是系统命令字, (字段名表)列出若干个字段名,字段名之间用逗号分隔开,表示命令仅对所列字段进行操作。省略此选择项,表示对所有的字段进行操作。

FOR (条件) 和 WHILE (条件):FOR 和 WHILE 是系统命令字,〔条件〕是一个逻辑表达式,它的值是“真”或“假”。这个选择项是选择记录的,表示命令仅对作为(条件)的逻辑表达式值是“真”的记录进行操作。省略此选择项时,表示命令对所有记录进行操作。当 FOR (条件) 和 WHILE (条件) 在同一个命令语句中使用时,系统规定 WHILE (条件) 优先。

3. 命令书写的规则

(1) 每条命令必须以命令动词开头,动词后面的选择项先后顺序可以是任意的,但必须符合命令格式的规定。

(2) 用空格来分隔每条命令中的各个选择项,但应注意, T, 或 F, 两个逻辑值中的小圆点和字母之间不容许有空格。

(3) 一个命令的最大长度是 254 个字符。在计算字符总数时空格应包含在内。如果一个命令太长,一行写不下时,可以使用续行符“,”在行尾进行分行,并在下行连续书写。

(4) 命令中的字符大小写可以混合使用。

(5) 命令动词和命令字可以用其前四个字母表示。

(6) FoxBASE+ 没有保留字,但在给标识符命名时应尽可能不使用 FoxBASE+ 系统中命令动词和命令字,以免程序在运行时发生混乱。

(7) 一行只能写一条命令,每一条命令的结束标志是按回车键。

二、运行方式

汉字 FoxBASE+ 有命令方式(交互方式)和程序方式两种运行方式。

命令方式是在汉字 FoxBASE+ 的圆点提示符后键入命令行,按回车键后立即执行之。

程序方式是通过汉字 FoxBASE+ 命令:

MODIFY COMMAND (命令文件名)

或行编辑程序先建立命令文件。建立时一行行键入命令行,然后存入磁盘,文件名是用户指定的(命令文件名),扩展名是 .PRG。然后由 DO 命令,即:

DO (命令文件名)

执行。

程序方式是汉字 FoxBASE+ 的主要执行方式。

习 题

1. 试说明数据与信息区别和联系。

2. 什么是数据库、数据库管理系统？
3. 数据库管理系统在数据库系统中的地位和作用是什么？
4. 什么是字段、字段值、记录？
5. FoxBASE+ 有几种数据类型？各种数据类型的意义和表达式如何？
6. FoxBASE+ 的变量有几种，它们之间有什么异同点？
7. 说明下列数据哪些是变量？哪些是常数？是什么类型的常数？
姓名、"年龄"、Y、y、"01/12/91"、成绩、"97.88"、"汉字 FoxBASE+"103.22、职称、L、D、
.N。
8. 请列出下列逻辑表达式：
 - (1) 年龄大于 40 岁和小于 50 岁的所有高工名单；
 - (2) 年龄大于 20 岁和小于 35 岁，且工资等于 82 元的所有助工的名单；
 - (3) 工龄大于 30 年，工资大于 113 元和小于 140 元的工程师以上所有人员名单。
9. FoxBASE+ 有几类文件？起什么作用？各自的默认扩展名是什么？
10. 如何安装你的 FoxBASE+ 系统？

第二章 汉字 FoxBASE+ 库文件操作命令

库文件操作命令包括数据库的建立,数据库的整理和数据库的操作使用三个方面。

第一节 数据库的建立

汉字 FoxBASE+ 可以通过多种方法建立数据库,归纳起来可分为四类:直接用 CREATE 命令建立新数据库;将已建立的旧数据库改造成新的数据库;通过两个或两个以上的旧数据库联接产生一个新的数据库;利用汉字 FoxBASE+ 以外的程序产生的数据文件变换为数据库文件。本章介绍前三类方法,第四类方法在第四章第六节的“汉字 FoxBASE+ 与其它软件的接口”中介绍。

一、首次建立数据库

汉字 FoxBASE+ 建立数据库要分两步进行,第一步建立数据库结构,第二步输入数据。第一次建立数据库可用 CREATE 命令实现。

格式:CREATE [〈文件名〉]

功能:建立一个新的数据库文件的结构,并把文件加到磁盘目录中去。

〈文件名〉指的是一个新的要建立的数据库文件名,由用户自己确定,由不超过 8 个字符的字母、数字和下划线组成。扩展名 .DBF 若没填写,系统会自动加上。如果〈文件名〉缺省,则系统将显示

Enter the name of the new file;

要求用户输入新数据库的文件名。如果文件名已存在,则系统将提示

C: emp. dbf already exists, Overwrite it? (Y/N)

其意思为 C 盘上 emp. dbf 已存在,询问用户是否将其覆盖。若用户认为可以覆盖,键入 Y 键,系统则将 C 盘上原来的 emp. dbf 文件的数据全部覆盖,并建立新的 emp. dbf 文件。

每条记录是由若干个字段组成。建立新的数据库文件的结构就是定义该数据库由哪些字段组成。汉字 FoxBASE+ 允许每条记录最多可有 128 字段。定义数据库文件结构的各字段时必须给出每个字段的字段名(Fieldname)、类型(Type)、宽度(Width)、小数位数(Dec)。

字段名是某个字段的名称,在一个文件中必须唯一。字段名必须以汉字或字母开头的字母、汉字、数字及下划线组成,最多可达 10 个字符(5 个汉字)。

类型表示该字段中存放的数据类型,用其第一个英文字母(C、N、L、D、M)确定,它们分别是字符型(C)、数值型(N)、日期型(D)、逻辑型(L)和备注型(M)。

宽度说明该字段允许的最大字符数。字符型的宽度为 1~254,数值型为 1~19,逻辑型为

1,日期型为 8,备注型为 10。

小数位数仅对数值型有用,允许取值 0~15。统计宽度时,小数点本身也算一个字符。

下面以一个人事档案数据库文件 emp.dbf 为例,来介绍一下建立数据库结构的具体步骤。

emp.dbf 由 8 个字段组成:职工号、姓名、性别、出生时间、工作时间、职称、婚否、工资。

职工号字段为数值型,宽度为 4,因为职工号为职工的编号,因此没有小数。

姓名字段为字符型。考虑到有复性的可能,故定义该字段宽度为 8。

性别字段为字符型,宽度为 2。

出生时间、工作时间均为日期型,系统规定宽度为 8。

职称字段为字符型,宽度根据实际情况而定,所以,这里定义为 6。

婚否字段可定义为一个逻辑字段,用逻辑值(Y 或 N)来表示“已婚”和“未婚”,系统规定宽度为 1。

工资字段为数值字段,宽度定义为 6,小数定义为 2 位(精确到分)。

假设汉字 FoxBASE+ 系统已在 C 盘中,现在要在 A 盘上建立 emp.dbf 数据库文件,则可在“.”提示符下,键入操作命令

• SET DEFAULT TO A

这条命令的功能即指定 A 驱动器为工作盘。接下来便可输入建立数据库的命令

• CREATE emp

扩展名 .DBF 系统将会自动加上。屏幕上显示如下:

CURSOR ← →		INSERT		DELETE		Up/Down Field, ← E X	
Char, ← S ← L		Char, ← V		Char, ← G		Cursor Menu, ←]	
Word, ← A ← F		Field, ← N		Word, ← Y		Exit/Save, ← W	
Pan, ← ← → →		Help, ← \		Field, ← U		Abort, Esc	
field name	type	width	dec	field name	type	width	dec

1 Character

Enter the name field

Field names begin with a letter and may contain letters, digits and underscores

屏幕上部是一个用于全屏编辑的光标控制键一览表。底部为系统提示信息,告诉用户此时要输入字段名,并提示用户字段名是以字母开头,由字母、数字及底划线组成。在汉字 FoxBASE+ 中允许使用汉字。字段名最多允许使用 10 个字符,其长度由第一个方框限制住了。

这时用户可在第一个光标处输入“职工号”及回车键,光标则自动移至第二个方框处。注意:输入字段名时,字段名中间不允许出现空格,否则,回车后光标不能移到第二个方框处,同

时系统将鸣铃警告。第一个方框输入完毕, 屏幕显示如下:

CURSOR ← →	INSERT	DELETE	Up/Down Field: ^ E ^ X
Char: ^ S ^ L	Char: ^ V	Char: ^ G	Cursor Menu: ^]
Word: ^ A ^ F	Field: ^ N	Word: ^ Y	Exit/Save: ^ W
Par: ^ ← ^ →	Help: ^ \	Field: ^ U	Abort: Esc
field name type width dec	field name type width dec		

1

Press SPACE to change the field type

CHARACTER fields contain character information of a specified length.

光标此时在类型上, 要求用户输入字段类型, 屏幕上显示系统自动给出字符型, 若符合要求, 则按回车键即可; 若不符合, 则按空格键进行选择; 也可直接输入各字段类型的第一个英文字母 C、N、D、L、M。现在职工号为数值型, 可输入“N”然后按回车键, 则屏幕显示

CURSOR ← →	INSERT	DELETE	Up/Down Field: ^ E ^ X
Char: ^ S ^ L	Char: ^ V	Char: ^ G	Cursor Menu: ^]
Word: ^ A ^ F	Field: ^ N	Word: ^ Y	Exit/Save: ^ W
Par: ^ ← ^ →	Help: ^ \	Field: ^ U	Abort: Esc
field name type width dec	field name type width dec		

1

Enter the field width

Numeric fields are 1 to 19 digits wide, including the decimal point and sign.

光标此时停在宽度上, 输入数字“4”及回车键, 则屏幕显示

CURSOR ← →	INSERT	DELETE	Up/Down Field: ^ E ^ X
Char: ^ S ^ L	Char: ^ V	Char: ^ G	Cursor Menu: ^]
Word: ^ A ^ F	Field: ^ N	Word: ^ Y	Exit/Save: ^ W
Par: ^ ← ^ →	Help: ^ \	Field: ^ U	Abort: Esc
field name type width dec	field name type width dec		

1

Enter the number of decimal places.

Decimal widths are 1 to 15 and must be at least 2 less than the field width.

光标此时移至第四个方框上,要求用户输入小数位数。职工号字段没有小数,则不需输入,直接按回车键,光标跳到下一个字段上,屏幕显示如下:

CURSOR ← →	INSERT	DELETE	Up/Down Field, ^ E ^ X
Char, ^ S ^ L	Char, ^ V	Char, ^ G	Cursor Menu, ^]
Word, ^ A ^ F	Field, ^ N	Word, ^ Y	Exit/Save, ^ W
Pan, ^ ← ^ →	Help, ^ \	Field, ^ U	Abort, Esc

field name	type	width	dec	field name	type	width	dec

1 职工号	Numeric	4					
2		Character					

1 职工号 Numeric 4

2 Character

Enter the name field

Field names begin with a letter and may contain letters, digits and underscores

现在便可输入第二个字段了。同第一个字段的输入步骤一样,依次输入字段名“姓名”,类型“C”,宽度“8”。对于字符型字段,只有字段宽度,没有小数,故只需输入字段宽度“8”,然后按回车键,系统就跳到第三个字段上,如下图所示。

CURSOR ← →	INSERT	DELETE	Up/Down Field, ^ E ^ X
Char, ^ S ^ L	Char, ^ V	Char, ^ G	Cursor Menu, ^]
Word, ^ A ^ F	Field, ^ N	Word, ^ Y	Exit/Save, ^ W
Pan, ^ ← ^ →	Help, ^ \	Field, ^ U	Abort, Esc

field name	type	width	dec	field name	type	width	dec

1 职工号	Numeric	4					
2 姓名	Character	8					
3		Character					

1 职工号 Numeric 4

2 姓名 Character 8

3 Character

Enter the name field

Field names begin with a letter and may contain letters, digits and underscores

第三个字段是性别字段,也是字符型,同第二个字段一样输入。第四、第五个字段都是日期型,其宽度固定为 8,且无小数位数,故只需输入字段名、类型“D”。后面的字段依次同样输入,直至 8 个字段输入完毕为止。此时,屏幕显示如下:

CURSOR ← →	INSERT	DELETE	Up/Down Field, ^ E ^ X
Char, ^ S ^ L	Char, ^ V	Char, ^ G	Cursor Menu, ^]
Word, ^ A ^ F	Field, ^ N	Word, ^ Y	Exit/Save, ^ W
Pan, ^ ← ^ →	Help, ^ \	Field, ^ U	Abort, Esc