

全国计算机等级考试教程(一、二级)

FOXBASE+

程序设计

曹德范 王文德 主编



地震出版社

73.967
CDF

全国计算机等级考试教程（一、二级）

FOXBASE⁺程序设计

主 编：曹德范 王文德

副主编：杜玉越 张庆德 许艳丽

高 峰 姚洪伟 张月岭

地 震 出 版 社

FOXBASE+程序设计

曹德范 王文德 主编

责任编辑:王 伟

*

地震出版社出版

北京民族学院南路9号(100081)

北京丰华印刷厂印刷

新华书店北京发行所发行

全国各地新华书店经售

*

787×1092 1/16 17.5印张 448千字

1998年3月第一版 1998年3月第一次印刷

印数 0001~4000

ISBN 7-5028-1540-6/TP·37

(1977) 定价:19.00元

前 言

自从电子计算机问世到现在短短几十年的时间,其应用范围日趋扩大,迅速渗透到工业、农业、国防、文教、医疗、交通、气象等以及日常生活的各个领域,并且还在继续扩大。因此,人们越来越重视学习、掌握计算机基础知识和操作技能,以适应我国现代化建设的需要。

在这种形势下,国家教委考试中心推出了计算机等级考试。它是一种客观、公正、科学的专门测试非计算机专业人员计算机应用知识与技能的全国范围的考试,通常称为全国计算机等级考试。该项考试推出以来受到社会各行各业的高度重视和积极响应,许多单位和部门把掌握计算机知识和应用技能作为考核、录用、选拔工作人员的重要条件之一。因此,全国计算机等级考试为用人单位在聘用计算机应用人员方面提供了一个科学而公平的考试标准,同时又为非计算机专业人员在择业、人才流动等方面开辟了一条道路。

全国计算机等级考试分四个等级。一级考试涉及计算机基本知识和使用微机系统的初步能力,主要是为从事文字、表格处理和常规信息检索、办公信息处理的人员设立的。二级考试涉及软、硬件基础知识和使用一种高级语言(BASIC,FORTRAN,PASCAL,C,FOXBASE⁺)编制程序、上机调试的能力,主要是为从事计算机或数据库管理人员而设立的。三级分A、B类考试:三级A类考核计算机应用基础知识和计算机硬件系统开发的初步能力;三级B类考核计算机应用基础知识和计算机软件系统开发的初步能力,它主要是为从事计算机应用工程中软、硬件工程人员而设立的。四级考核计算机项目或应用系统的分析和设计的必备能力。四个级别的考试均有笔试和上机考试两部分。

为了进一步推动计算机知识的普及与应用,促进等级考试的顺利开展和有效进行,根据考试大纲要求及有关参考材料,我们编写了《FOXBASE⁺程序设计》一书,以帮助非计算机专业人员尽快学习和掌握计算机的基础知识和操作技能。本书共分五章,第一章为数据库概论,第二章为数据库文件的基本操作,第三章为FOXBASE⁺程序设计,第四章为输入输出格式设计,第五章为等级考试模拟试题。每章后面都附有适量的习题,供读者进一步巩固和掌握所学的知识。

根据我们多年从事计算机教学的经验和实践,在本书编写过程中,既考虑到知识的系统性和完整性,又注意到内容的通俗性和实用性。

本书不仅适合于参加全国计算机等级考试(一、二级)的广大读者,而且也可以作为大专院校非计算机专业及中等专业学校在校生的计算机教学用书和参考书。

参加本书编写工作的有曹德范、王文德、杜玉越、张庆德、许艳丽、高峰、姚洪伟、张月岭,最后由曹德范、王文德组织审校和定稿。

由于时间仓促,加上编写者知识水平有限,书中难免存在错误和不当之处,敬请广大读者批评指正。

编者

1998.1

目 录

第一章 数据库概论	(1)
1.1 数据库基本概念	(1)
1.2 FOXBASE+ 简介	(7)
1.3 常量、变量、函数和表达式	(14)
1.4 FOXBASE+ 的主要技术指标	(31)
实验 1.1 FOXBASE+ 的进入、退出、变量赋值和输出	(32)
实验 1.2 函数和表达式	(33)
习题一	(34)
第二章 数据库文件的基本操作	(39)
2.1 数据库文件的建立	(39)
2.2 数据库文件的打开和关闭	(45)
2.3 数据库记录的追加与插入	(46)
2.4 数据库记录指针的定位操作	(50)
2.5 数据库记录的显示与编辑	(51)
2.6 数据库文件的查询	(57)
2.7 数据库文件的统计与汇总	(64)
2.8 涉及多个数据库文件的操作	(68)
2.9 FOXBASE+ 辅助操作命令	(75)
实验 2.1 数据库的建立、记录的输入、记录的显示与数据库结构的显示	(81)
实验 2.2 记录指针的移动、数据库记录与数据库结构的修改	(83)
实验 2.3 数据库文件的排序、索引和查询	(85)
实验 2.4 数据库文件记录数据的统计与汇总	(86)
实验 2.5 多工作区数据的操作	(88)
实验 2.6 FOXBASE+ 环境中文件的操作命令	(90)
习题二	(91)
第三章 FOXBASE+ 程序设计	(99)
3.1 FOXBASE+ 程序及其特点	(99)
3.2 命令文件的建立与执行	(101)
3.3 顺序结构程序设计与交互式命令	(102)
3.4 分支结构程序设计	(113)
3.5 循环结构程序设计	(121)

3.6 内存变量与数组	(137)
3.7 过程及其调用	(144)
3.8 菜单程序设计	(155)
3.9 综合程序设计	(165)
实验 3.1 简单程序设计	(171)
实验 3.2 分支程序设计	(172)
实验 3.3 循环程序设计	(172)
实验 3.4 数组和记录的数据交换	(175)
实验 3.5 过程及过程调用	(177)
实验 3.6 菜单程序设计	(177)
第四章 输入输出格式设计	(181)
4.1 格式控制语句	(181)
4.2 屏幕画矩形框格式语句	(185)
4.3 屏幕格式文件	(186)
4.4 打印报表格式设计	(188)
实验 4.1 综合程序设计	(191)
第五章 等级考试模拟试题	(199)
5.1 等级考试简介	(199)
5.2 模拟试题	(201)
附录一 全屏幕编辑控制键功能一览表	(255)
附录二 FOXBASE⁺命令一览表	(256)
附录三 FOXBASE⁺函数一览表	(262)
附录四 一级考试大纲	(266)
附录五 二级考试大纲	(268)
附录六 数据库语言程序设计考试要求	(270)
参考文献	(271)

第一章 数据库概论

在数据处理领域中,数据库技术变得越来越重要,已成为计算机应用中最为活跃的课题之一。随着信息处理技术的飞速发展,计算机管理数据的方式也不断改进。50年代末出现了文件管理系统,它把数据组织在一个个独立的数据文件中,采用按文件名访问,按记录进行存取的管理技术,大大减轻了程序员的数据管理劳动。70年代初发展起来的数据库管理技术不断完善和加强,已成为数据处理的核心问题,数据库技术的最新成果反映了数据处理的最新水平。

本章作为绪论,说明什么是数据库,介绍数据库技术的基本思想方法、基本概念及目前较流行的关系型数据库 FOXBASE⁺的基础知识。

1.1 数据库基本概念

1.1.1 数据管理技术发展的三个阶段

数据管理是指对数据的组织、存储、检索和维护等工作,是数据处理系统的核心,数据库技术是数据管理中的一种新方法。数据管理技术随着计算机软、硬件技术的发展,经历了人工管理、文件系统和数据库系统三个阶段。

1. 人工管理阶段

这是计算机用于数据管理的初级阶段。早期的计算机除了硬件之外,没有必要的软件支持,根据应用要求,程序员必须自行考虑数据的物理存放位置,因而用户程序不仅要设计数据处理的方法,还得记住数据在内存的地址,对数据的访问是按物理地址进行的。一般地,数据的逻辑结构与物理组织大体相同,基本上没有文件的概念,并且一组数据通常只对应于某一应用程序,数据依赖于程序,数据稍有变动,便导致程序全部作废,很难实现多个应用程序共享数据资源。

2. 文件系统阶段

60年代后,文件系统迅速发展。将数据组织成文件的形式是数据管理的重大发展。在文件系统中,数据以文件形式存储在外存上,用户利用文件系统对数据进行各种操作,如插入、删除、修改和查询等。数据和管理它的应用程序相互独立,数据的物理组织的改变可以不影响逻辑组织,但数据文件往往是根据某一应用程序的要求而建立的,数据逻辑组织服从于应用程序的要求,一旦数据结构作了修改,则应用程序也需作相应的修改,反之亦然。因此,在文件系统阶段,难以实现以文件为单位的数据共享。

文件管理系统是应用程序和数据之间的一个接口。应用程序通过文件管理系统建立和存储文件时,仅需注意数据的逻辑意义和物理特征,较少考虑数据的物理存储问题。

3. 数据库系统阶段

60年代末,提出了数据库的概念。数据库技术的主要目的是为了实现数据在最大范围内的共享,并使应用程序尽可能地不依赖于全局数据的逻辑结构和物理结构。

数据库系统管理方式的基本特征是采用了逻辑数据模型,具有完整的数据结构,数据之间存在着内在联系,可以最大限度地避免数据的重复;数据组织面向整个系统,用整体观念组织数据,形成一个数据中心,如同一个数据仓库,库中数据能满足所有用户的不同需求。这时应用程序不再与一个孤立的数据文件相对应,而是取自全局数据集的某个子集作为逻辑文件与应用程序相对应,通过一个软件——数据库管理系统 DBMS(Database Management System)实现逻辑文件与物理数据之间的映射。图 1.1 表示数据库管理方式的思想方法。

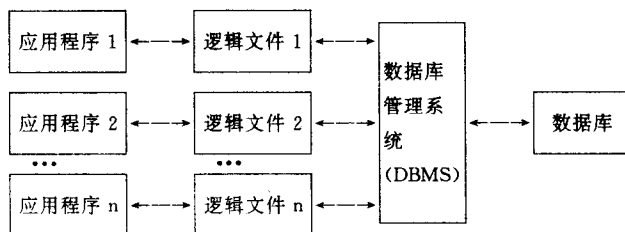


图 1.1 数据库系统管理方式示意图

1.1.2 数据、信息和数据处理

1. 数据

数据的含义不局限于传统的数值数据,而且还包括声音、语言、文字、图像等多种类型。在一般意义上,人们认为数据是客观实体的属性值,是一串物理符号的序列,是用来反映客观世界而记录下来的可以被鉴别的符号。

由于计算机只能存储、处理数据,表达客观实体的符号当然必须遵从一定的约定,根据约定,声音、文字、图像等都可以表示成数据并被计算机存贮,这就意味着数据不仅指那些具有量值大小的纯数值,更为一般的是对实体特征进行描述的数据。在数据库中,由于表示实体属性值的类型不同,决定了数据类型也不同。如职工数据库中的基本工资、补贴、扣款等为数值型数据,而姓名、职称等为字符型数据。因此,进行数据库设计时,必须定义数据类型,不同类型数据表达实体的不同属性,不可混淆。

2. 信息

信息是事物及其运动状态的表现形式,是人对客观事物的一种抽象理解。在许多场合下,信息与数据不加区分。通常人们认为,信息是经过加工(处理)后并对人类社会实践和生产活动产生决策影响的数据。

实质上数据和信息是两个互相联系、互相依存又互相区别的概念。数据和信息的关系是原料和结果的关系,也就是说,不是所有数据都能成为信息,经加工处理对决策有作用的数据才能成为信息。因此,数据反映信息,信息依靠数据表达,信息来源于数据。

3. 数据处理

数据处理(或信息处理)泛指数据的收集、存贮、分类、检索、维护、传输、计算、决策等一系列操作。其基本目的是对大量的、杂乱无章的原始数据进行整理、归并、加工计算,抽取并分析出对人们有价值的成分,作为决策的依据。

数据处理业务量大面广,各行各业各个领域每时每刻都在使用不同工具,采用不同技术进行着。自从用计算机实现数据处理后,不仅加快了处理速度,而且也扩大了数据处理的范围和规模。例如办公室自动化、银行账务、交通调度、人口统计、情报检索、图像处理、人工智能等等

都是数据处理的典型实例。

1.1.3 数据库、数据库管理系统与数据库系统

1. 数据库

数据库(Database)是数据处理的最新技术,是一种先进的软件工程。数据库这个词形象但并不准确,其实数据是个复杂的系统,难以用简练语言概括其全部特征。直观地说,数据库是存放数据的“仓库”,在计算机上使用的这个“仓库”就是指磁盘(硬盘或软盘)等外存设备。

尽管给出数据库的定义相当困难,人们还是从不同角度突出数据库的不同特点,给出不同的描述。下面给出我们认为提法比较全面的 Martin James 数据库定义,他既指出数据库的数据组织方法,又强调了数据库系统的数据管理方式。

数据库是一种相关数据的集合,这些数据避免了不必要的数据冗余,能为不同的应用程序所共享。数据的存储独立于程序,对数据的插入、更新、检索等操作均能以一种通用的和可控制的方式进行。

2. 数据库管理系统

数据库管理系统 DBMS 是一个软件系统。它是数据库系统的核心部分。它起着用户或应用程序与总体数据库之间的接口作用和数据统一管理作用。用户可以用查询语言或终端命令操作数据库,也可以用程序方式操作数据库。数据库管理系统为应用程序和数据库数据间提供了数据的逻辑独立性和物理独立性,并且数据库系统还提供数据保护和并发控制功能。因此,有了数据库管理系统的支持,用户不必注重于数据的逻辑或物理表达细节,只注重数据的内容就可以了。

一般来说,数据库管理系统由如下三部分构成:

- (1)数据库描述语言 DDL(Data Description Language)及翻译程序;
- (2)数据库操纵语言 DML(Data Manipulation Language)或查询语言及其翻译程序;
- (3)数据库管理子程序,包括系统运行控制程序、语言翻译程序、DBMS 公用程序。

3. 数据库系统

数据库系统是由计算机系统、数据库及其描述机构、数据库管理系统和用户组成,是由这几个方面组成的具有高度组织性的总体。因此,数据库系统不是指数据库本身,也不是指数据库管理系统,而是指计算机系统中引入数据库之后的系统组成。

一般来说,数据库系统包括计算机系统、数据和用户三个部分。

(1)计算机系统:

计算机系统是指用于数据管理的硬件和软件资源。硬件资源包括构成计算机系统的各种物理设备,如中央处理器、内外存储器及外部设备。软件资源包括操作系统、数据库管理系统(DBMS)和应用程序。

(2)数据:

存储在系统中的数据被分成一个或多个数据库文件。各数据库之间没有不必要的数据冗余。存储在数据库中的数据是从整体观点来组织的,不是面向个别应用,而是为多个应用共享的。

(3)用户:

用户有三大类,第一类是数据库管理员,他们负责对整个数据库系统进行总体控制和维

护,以保证数据库系统的正常运行;第二类用户是应用程序员,负责编写和维护应用程序,这些应用程序能对数据库进行通常的操作,如检索、建立、删除或改变信息等操作;第三类是最终用户,他们通过终端访问数据库,最终用户可以通过查询命令或应用程序对数据进行检索、插入、删除或更新等操作。

综上所述,我们可对本节讨论的三个概念作如下总结:数据库是以一定组织方式存储且具有相关性的数据集合。数据库管理系统是一个数据库管理软件,它的职能是维护数据库,接受和完成用户程序或命令提出的访问数据的各种请求。而数据库系统是指计算机系统引进数据库后组成的系统。一般数据库系统由数据库、数据库管理系统和用户构成。用户使用数据库是目的,数据库管理系统是帮助达到这一目的的工具和手段。

1.1.4 数据模型

数据模型给出了一种看待和描述数据及数据之间联系的抽象方法。数据模型设计是数据库系统的核心问题,它不仅用于明确表达构成数据库的数据的逻辑结构,还应表示对具有这些结构的数据的约束和允许的操作。数据模型的设计方法又决定了数据库的设计方法。当前较流行的设计方法有层次方法、网状方法和关系方法,与其相对应的数据模型是层次模型、网状模型和关系模型。下面简要介绍这三种模型。

1. 层次模型

层次模型是数据库中最早出现的数据模型,它是按照层次结构的方式组织数据的数据模型。它把数据间的联系表示成有向树或森林。这种数据结构就像一棵倒置的树。树中每个节点是记录型,规定一个父记录型可对应于多个子记录型,而一个子记录型只对应于一个父记录型。层次模型只能处理两个实体集之间的二元的 1:N 的层次型的联系。对于非层次型的其他联系,有些可转化为层次模型可接受的树结构。但如果两层实体集间具有 N:M 联系,一般情况下层次模型将无能为力。

2. 网状模型

层次模型可以有效地描述现实世界层次型联系的事物,它结构简单、层次清楚、便于理解,但对于大量存在的非层次型数据间的联系,用层次模型难以表达,这时可采用网状模型。

任意一个连通的相关结点的集合就是一个网状模型。网状模型反映数据间的错综复杂的联系。各数据间不仅表现为层次型联系,也发生横向联系。网状数据库中任一数据都可与其他任意结点上的数据实现联系,而层次型结构仅限于不同层次之间的纵向联系,这是二者的本质区别。

3. 关系模型

关系模型是用二维表形式表示信息以及信息之间的联系。在关系模型中,每一个二维表称为一个关系,表示某个实体集,表中每一行代表实体集中的一个实体,每一栏目称为属性,表名为实体集的名称。二维表既表达实体,又表达联系,这种统一的格式是关系数据模型的特点。

例如有三个实体:“学生”、“教师”、“课程”,相应的三个表格框架如图 1.2 所示,“学生”和“课程”是两个不同的关系,但学生总是要学习课程的,它们之间有内在联系。同理,教师总是要教学生的,因此“教师”与“学生”也有内在联系。为了建立“学生”与“课程”之间的联系,需引入另一个实体“学习”,它包含:学号、课程号、分数三个属性,可通过学号和课程号建立两关系间的联系。

编号	姓名	职称	年龄	教师关系框架
学号	姓名	年龄	性别	学生关系框架
课程号	课程名	学时数	编号	课程关系框架

图 1.2 关系框架

因此,使用二维表可实现实体间的联系。如果两个不同的关系确实存在客观间的联系,那么就一定能在各关系中找到具有相同性质的属性,通过两个关系的属性取值相等条件建立彼此间的联系。

在层次、网状和关系三种数据模型中,关系模型是唯一一种建立在数学概念关系理论基础上的数据模型,它成为数据库领域中最活跃的一个分支。目前,较流行的数据库系统绝大多数是关系数据库系统。

1.1.5 关系数据库基本理论

关系模型是建立在集合代数理论之上的数据模型,关系数据库是以关系模型为基础的数据库结构。近年来,已研制出若干采用关系模型的效率较高的实用的关系数据库管理系统,如 FOXBASE⁺、ORACLE 等。

1. 关系的数学定义

关系理论以集合论为基础,关系也可以在集合代数中定义。为了从集合论的角度给出关系,下面先引进笛卡尔积的概念。

【定义 1】设 $D_1, D_2, \dots, D_n$ 为一组集合,它们的笛卡尔积定义为:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) | d_i \in D_i, i=1, 2, \dots, n\}$$

其中每个 $(d_1, d_2, \dots, d_n)$ 叫做一个 n 元组,元组中的每个 d_i 叫做一个分量,是 D_i 中的一个值。 D_i 中元素的个数称为它的基数,记作 $|D_i|$ 。

【定义 2】笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的任意一个子集称为 $D_1, D_2, \dots, D_n$ 上的一个 n 元关系。

因此,关系也是集合,它的元素是元组,关系也可用二维表表示。由于现实世界中事物和事物间联系客观上存在某些约束,这就限定了某一关系模式的所有关系的内容。事实上,在笛卡尔积中,去掉那些不符合客观事实或没有意义的元组,这就是关系。

例如,设省名集合 $X = \{\text{山东, 江苏, 浙江}\}$,城市名集合 $Y = \{\text{济南, 南京, 杭州}\}$,则笛卡尔积 $X \times Y = \{(\text{山东, 济南}), (\text{山东, 南京}), (\text{山东, 杭州}), (\text{江苏, 济南}), (\text{江苏, 南京}), (\text{江苏, 杭州}), (\text{浙江, 济南}), (\text{浙江, 杭州}), (\text{浙江, 南京})\}$ 。 $X \times Y$ 的二维表表示,见表 1.1。

设省-城市关系为 R ,则 $R = \{(\text{山东, 济南}), (\text{江苏, 南京}), (\text{浙江, 杭州})\}$, R 的二维表表示,见表 1.2,显然关系 R 是笛卡尔积 $X \times Y$ 的一个子集。

同理,若再引入学校集合 $Z = \{\text{山东大学, 南京大学, 浙江大学}\}$,由笛卡尔积 $X \times Y \times Z$,可得到省-城市-学校关系 R_1 ,且 $R_1 = \{(\text{山东, 济南, 山东大学}), (\text{江苏, 南京, 南京大学}), (\text{浙江, 杭州, 浙江大学})\}$ 这是一个三元关系,它的二维表形式,见表 1.3。

FOXBASE⁺是一种关系数据库,在关系数据库中,把关系看成一个文件,关系中每个元组

称为一个记录,即表中一行。表中每一列具有一个属性,称为一个字段或一个数据项。

表 1.1 X×Y 的二维表

省名	城市名
山东	济南
山东	南京
山东	杭州
江苏	济南
江苏	南京
江苏	杭州
浙江	济南
浙江	南京
浙江	杭州

表 1.2 省-城市关系 R

省名	城市名
山东	济南
江苏	南京
浙江	杭州

表 1.3 省-城市-学校关系 R1

省名	城市名	学校名
山东	济南	山东大学
江苏	南京	南京大学
浙江	杭州	浙江大学

2. 关系的性质

由于关系的集合特性,关系具有如下性质:

(1) 关系中,每个属性值是不可再分的元素,它取自同一个域,各个属性必须赋以不同名称,属性的个数称为关系的“元”或“度”。

(2) 列的次序是不重要的。

(3) 关系中,各元组各不相同,不允许有重复的元组。

(4) 元组的次序也是不重要的。

由此可见,判断两个关系是否相等,与属性次序无关,对关系如何命名也无关;若两个关系的差别只是关系名不同,属性次序不同或元组次序不同,则认为这两个关系是相等的。

3. 关系数据库中的数据组成

下面以目前最流行的关系数据库系统 FOXBASE⁺ 为例,简单介绍关系数据库的数据组成和关系运算。

FOXBASE⁺ 的处理对象是数据,按照关系模型,数据的组成可分为四个层次:字段、记录、文件和数据库,其中每一层次均是后一层次的组成部分。

(1) 字段(FIELD):

字段是关系数据库文件中不可分割的数据单位。一个字段由一个或多个字符构成。字符的个数称为字段的宽度。用字段描述某个实体对象的属性,相当于二维表格中的一列,一般用属性名作为字段名。

(2) 记录(RECORD):

记录是描述某一个体的数据集合。记录由若干字段组成,组成记录的所有字段长度之和称为记录长度。记录相当于二维表格中的一行。

(3) 文件(FILE):

文件也称数据库文件,它是具有相同性质的记录的集合。文件中的记录是同格式、等长度的。一个文件相当于一张二维表。

(4) 数据库(DATABASE):

可以看成一组相关文件的集合。

4. 关系数据库中的关系运算

关系运算是关系数据语言的基础。关系数据语言的核心部分是查询操作,它们往往用一个关系运算表达式来表示。传统的集合运算(并、交、差等)不能灵活地实现多样的数据查询操作。因此,关系模型定义了自己的关系运算,包括筛选(SELECT)、投影(PROJECT)和联接(JOIN)等运算。下面仍以 FOXBASE⁺ 为例,介绍关系数据库中的关系运算。

(1) 筛选:

筛选是对数据库文件中的记录进行选择,使操作只对选中的记录有效。FOXBASE⁺ 对记录的选择有如下三种方法:

在命令中给出范围短语,使命令只对指定范围内的记录进行操作。

在命令中给出条件短语,使命令只对满足条件的记录进行操作。

用筛选命令 SET FILTER TO <条件> 把满足条件的记录筛选出来,使命令只对被筛选出来的记录进行操作。

(2) 投影:

投影是对数据库的字段进行选择,即根据用户的要求选择数据库中的部分字段作为操作对象。用命令 SET FIELDS TO <字段名表> 或在命令中用 FIELDS<字段名表> 短语使命令只对<字段名表> 中所指定的字段进行操作。

(3) 联接:

联接是对两个数据库文件进行横向拼接。根据用户的指定,将两个数据库文件中的某些或全部字段,按照关键字段联接生成一个新的数据库文件。联接用系统提供的专用命令 JOIN 来实现。

此外,用命令 SET RELATION TO <文件名> 把当前打开的数据库文件和<文件名> 指定的文件建立关联,使命令的操作对象为这两个数据库文件。

上述关系操作可以单独进行,也可以组合在一起。对数据库的三种基本操作,我们可以作一个通俗的类比:假定你手头有一批相关二维表格,手中持有一把剪刀和一瓶胶水,你可根据需要,对表中的数据进行加工处理,然后进行剪裁、拼接、粘连,从而形成一张满足要求的新表格。只不过这里的处理速度是手工处理速度所望尘莫及的。

1.2 FOXBASE⁺ 简介

1.2.1 FOXBASE⁺ 的历史

FOXBASE⁺ 是一个关系数据库管理系统,它和 DBASE 一样,是近几年来应用最广的微型计算机软件。在关系数据库的发展过程中,国内外流行最早最广的是 DBASE, DBASE 经历了 DBASE I, DBASE II, DBASE III PLUS, DBASE IV 几个发展阶段。其中影响最大的是 1986 年推出的 DBASE III PLUS,被评为 1986 年美国最佳软件。DBASE III 具有很强的数据管理功能和灵活的程序设计功能,编程非常方便。与其他高级语言相比,不但省时省力,而且程序和数据独立,易于维护和扩充,因而广泛应用于各行各业的事务管理。DBASE III 进行了成功的汉化,易于为广大中国用户所接受,对我国的计算机应用的普及起到极大的推动作用。

然而,DBASE III 也有其不足,那就是计算能力较弱,处理速度慢,没有数组和自定义函数

等等。这些弱点不但限制了 DBASE III 在更多领域的应用,而且导致了在 DBASE III 上开发的应用项目往往只停留在单项程序或比较低的水平上,难以达到综合信息管理更高的要求。因此,人们很自然地期望有一个更新更好的数据库管理系统,它既能保持 DBASE III 众多特点和优点,又在功能、速度方面比 DBASE III 更强。正是在这种形势下,近年来不断有新的与 DBASE III 兼容的数据库管理系统软件推出。其中被公认为最成功、性能最优良的是美国 Fox Software 公司推出的 FOXBASE⁺。

FOXBASE⁺是美国 FOX 软件公司的产品,它是用 C 语言开发的,与 DBASE 兼容,速度是 DBASE III 的 6.9 倍,配上协处理器速度还可以进一步提高。自 1986 年 6 月 FOXBASE⁺1.0 问世以来,又相继推出了 FOXBASE⁺2.00 和 FOXBASE⁺2.10 等版本,功能更加完善。FOXBASE⁺是一个多用户数据库管理系统,可以在网络软件的支持下运行于多用户环境,也可在单用户环境下运行。另外有单用户版本,而且单用户版本和多用户版本语法完全兼容。

FOXBASE⁺的成功汉化,极大地开发了该软件在国内的市场,受到越来越多用户欢迎,成为国内最流行的软件之一。本书介绍 FOXBASE⁺2.10 汉化后的单用户版本。

1.2.2 FOXBASE⁺系统文件组成

在汉字操作系统下,FOXBASE⁺2.10 由一系列文件组成。

1. 基本文件

FOXPLUS.EXE(MFOXPLUS.EXE)	250kB	执行程序
FOXPLUS.OVL(MFOXPLUS.OVL)	140kB	覆盖程序

括号中的文件为多用户系统文件。执行程序 FOXPLUS.EXE 启动 FOXBASE⁺,此后常驻内存。覆盖程序 FOXPLUS.OVL 在 FOXBASE⁺启动后只有一部分装入内存,其余部分在需要时以覆盖的方式调入内存,使用后又被新的内容所覆盖。

有了这两个基本文件,FOXBASE⁺就可以运行。具有这两个基本文件的磁盘被称为 FOXBASE⁺系统盘。

2. 辅助文件

FOXHELP.EXE	160kB	帮助文件
FOXBIND.EXE	40kB	过程组合程序
FOXPCOMP.EXE	83kB	准编译程序

帮助文件 FOXHELP.EXE 用于向用户提供帮助,文件中有关于 FOXBASE⁺的语法及使用说明信息。过程组合程序 FOXBIND.EXE 用于把多个过程文件组合成一个过程。准编译程序 FOXPCOMP.EXE 用于对 FOXBASE⁺命令文件进行编译,以提高程序运行速度,增加程序的保密性。

1.2.3 FOXBASE⁺的运行环境、安装、启动与退出

1. FOXBASE⁺的运行环境

FOXBASE⁺在单用户环境使用时,要求 DOS2.0 以上版本;在多用户环境下使用时,要求 DOS3.1 以上版本。目前 MSDOS6 是最流行的磁盘操作系统,MSDOS5 以下的版本已经很少使用。中文操作系统 UC DOS3.1,UCDOS5.0 是目前应用最广的汉字平台。所以运行 FOXBASE⁺的软件环境不存在任何问题。

同样,FOXBASE⁺要求的硬件环境也失去实际意义。它要求一台 IBM PC/XT 标准配置以上的微机,因为这种微机已被淘汰,而 PC486,PC586 已是主流机型。

总之,目前流行的微型机运行 FOXBASE⁺不存在环境受限问题。

2. FOXBASE⁺的安装、启动与退出

(1)FOXBASE⁺系统安装:

这里所说的系统安装是指将存储在软盘上的 FOXBASE⁺系统文件拷贝到硬盘上。过程如下:

——进入中文操作系统,出现 C>

——在硬盘根目录之下建立 FOX 子目录

C>MD FOX

——进入子目录 FOX,使 FOX 子目录成为当前目录

C>CD FOX

——将 A 盘中的文件拷贝到子目录 FOX

C>COPY A:*. *

(2)FOXBASE⁺的启动:

——进入中文操作系统,出现 C>

——进入 FOX 子目录

C>CD FOX

——运行 FOXBASE⁺执行文件

C>FOXPLUS (多用户键入 MFOXPLUS)

待出现 FOXBASE 系统提示符“·”,表示系统启动成功。这时,用户可以键入各种命令,完成各种各样的操作。

(3)FOXBASE⁺的退出:

当数据库文件操作结束时,必须正常地退出 FOXBASE⁺系统,否则将有可能使数据库文件的数据遭到破坏或丢失。退出 FOXBASE⁺的命令是 QUIT,退出 FOXBASE⁺,返回到操作系统提示符:

· QUIT

C>

1.2.4 FOXBASE⁺的数据类型和文件类型

1. 数据库文件的基本概念

为了便于说明 FOXBASE⁺的数据类型,首先以表格的形式(见表 1.4)给出一个数据库文件,顺便介绍与数据库文件有关的概念。

(1)字段:

字段有时又称作域(FIELD)或数据项,它是数据库中可以处理的最小单位。表中每一列就是一个字段,每个字段有四个参数:字段名、字段类型、字段宽度和小数位数。

(2)记录:

表中每一行被称为一个记录,它由若干相互关联的字段组成。

表 1.4

人事档案表(RS.DBF)

Record # 开始标志		编号	姓名	性别	婚否	年龄	工作日期	职称	基本工资	简历
记录指针 → 当前记录	1	1002	刘小明	男	.T.	42	08/02/76	副教授	305.00	Memo
	2	1001	李 林	男	.T.	38	07/25/81	讲师	245.00	Memo
	3	3004	张晓霞	女	.F.	21	07/28/96	助教	175.00	Memo
	4	3001	郑文斌	男	.T.	55	10/23/66	教授	435.00	Memo
	5	3002	王 强	男	.F.	26	08/20/92	副教授	270.00	Memo
	6	3003	高文明	男	.T.	32	09/30/85	教授	390.00	Memo
	7	2001	吴 华	女	.T.	45	08/08/76	副教授	335.00	Memo
	8	2002	赵方平	女	.F.	25	07/30/93	讲师	245.00	Memo
	9	2003	林中升	男	.F.	21	08/23/96	助教	175.00	Memo
结束标志										

(3)数据库文件:

具有相同属性的记录的集合称为一个数据库文件,简称库文件。库文件名的命名规则和DOS规定一样,由文件基本名和扩展名两部分组成,FOXBASE+默认的库文件扩展名是.DBF。

(4)记录指针:

每个正在使用的库文件都有一个记录指针,它指向某一个记录。记录指针是FOXBASE+系统在内部为被打开的数据库文件配置的,用户可以根据需要通过命令来查看和改变指针的位置。随着命令的执行,指针的位置可能发生变化。

(5)当前记录:

记录指针指向某一记录,则该记录被称为当前记录。在对数据库文件进行操作或编写应用程序时,用户必须清楚地知道指针的位置,因为某些命令只对当前记录有效。

(6)记录号和记录号函数:

库文件中每一个记录都有一个序号,称为记录号。系统通过记录号计算出每个记录的数据在磁盘上的存放位置,从而实现快速存取。每个库文件都设置一个记录号函数RECNO(), RECNO()里存放着当前记录的记录号,若5号记录为当前记录,则RECNO()的值等于5。

(7)文件开始标志和文件开始测试函数:

每个库文件都设置一个文件开始标志和文件开始测试函数BOF()。只有当记录指针指向文件开始标志时,BOF()函数的值为逻辑值.T. (TRUE,真),否则为逻辑值.F. (FALSE,假)。

(8)文件结束标志和文件结束测试函数:

每个库文件都设置一个文件结束标志和文件结束测试函数EOF()。只有当记录指针指向结束标志时,EOF()函数的值为逻辑值.T. (真),否则为逻辑值.F. (假)。EOF()是一个

非常有用的函数, 当我们对数据库文件进行处理时, 往往是对每一条记录逐一处理, 通常用 EOF() 的值是 .T. 还是 .F. 来判断处理过程是否应该结束。

2. FOXBASE+ 的数据类型

数据类型表示数据的基本属性。它是一个十分重要的概念, 因为对数据操作必须遵循一条原则: 只有相同类型的数据才能进行操作, 否则就会出现语法错误。FOXBASE+ 定义了六种数据类型: 字符型、数值型、逻辑型、日期型、备注型和屏幕型。

(1) 字符型数据(Character, 用 C 表示):

字符型数据是指一切可打印的字符(如英文字母、数字、各种符号)。汉字也属于字符型数据, 每个汉字字符宽度为 2 个字节, 英文字符为一个字节, 在打印或显示时一个汉字字符占两个字符位置。一个字符型数据最长不得超过 254 个字节。表 1.4 的姓名、性别字段中的数据就是字符型数据。

(2) 数值型数据(Numeric, 用 N 表示):

数值型数据只能由数字、小数点和正负号组成, 表 1.4 的基本工资字段中的数据就是数值型数据。

数值型数据的最大宽度为 19 个字节(包括小数点和正负号位)。

(3) 逻辑型数据(Logical, 用 L 表示):

逻辑型数据是为进行各种逻辑判断而设置的, 它只可能有两个值: 一个是逻辑值 TRUE(真), 用 .T., .t., .Y., .y. 之一表示, 另一个是逻辑值 FALSE(假), 用 .F., .f., .N., .n. 之一表示。

如表 1.4 中的婚否字段中的数据就是逻辑型数据, 第一个记录的婚否字段值为 .T., 表示刘小明已结婚。

逻辑型数据占一个字节。

(4) 日期型数据(Date, 用 D 表示):

日期型数据用来存放一个日期, 系统默认的数据格式为“月/日/年”(MM/DD/YY), 长度固定为 8 个字节。如表 1.4 中工作日期字段就是日期型数据。

(5) 备注型数据(Memo, 用 M 表示):

数据库文件的字符型字段最大宽度为 254, 为了适应大块文本信息的存放, 在数据库文件中设置了 Memo 字段。但该字段只存放文本信息的地址指针, 其内容存放在数据库的辅助文件, 即专门用于存放 Memo 字段的备注文件中。文本信息最多可占用 8 块, 每块 512 字节, 共 4096 字节。表 1.4 中简历字段就是备注型字段。备注型数据仅用于字段名变量。

备注型字段的宽度固定为 10 个字节。

(6) 屏幕型数据(Screen, 用 S 表示):

屏幕型数据用来存放整个屏幕(25×80 个字符)的信息。屏幕型数据仅用于内存变量。

若存储当前屏幕映象, 需用如下命令实现:

SAVE SCREEN TO <内存变量>

若重现屏幕型内存变量存储的屏幕映象, 需用如下命令实现:

RESTORE SCREEN FROM <内存变量>

3. FOXBASE+ 的文件类型

FOXBASE+ 定义的九种类型的文件如表 1.5。我们仅对数据库文件、备注文件和索引文件