高等财经专科学校试用教材

FOXBASE⁺关系数据库系统

张洪瀚　主编

东北财经大学出版社

需要全本请访问 www.ertongbook.com

(辽)新登字 10 号

图书在版编目(CIP)数据

FOXBASE⁺ 关系数据库系统/ 张洪瀚主编 . - 大连:东北财经大学出版社,1996 .12
高等财经专科学校试用教材
ISBN 7-81044-174-4

Ⅰ .F… Ⅱ .张… Ⅲ 数据库管理系统,FOXBASE⁺-关系数据库-高等学校:专业学校-教材 Ⅳ .TP392BA

中国版本图书馆 CIP 数据核字(96)第 21015 号

高等财经专科学校试用教材

FOXBASE⁺ 关系数据库系统

FOXBASE⁺ Guanxi Shujuku Xitong

1996 年 12 月第 1 版
1996 年 12 月第 1 次印刷

张洪瀚 主编

责任编辑:朱 艳
责任校对:孙 萍

出 版 者:东北财经大学出版社
地 址:大连·黑石礁
邮政编码:116025

制 版:大连斯达电脑开发公司
印 刷:大连印刷工业总厂

开 本:850×1168 1/ 32
印 张:10 3/ 4
字 数:269 000
印 数:1—5 000

ISBN 7-81044-174-4/ T·35 定价:15 .00 元

内 容 简 介

FOXBASE$^+$ (2 .10)是美国 FOX 软件公司 1988 年 7 月推出的数据库管理系统。它具有与 dBASE - Ⅲ/ PLUS 完全兼容、运行速度快、多用户版本与单用户版本兼容、易于移植、适应于教学等诸多优点。目前,我国各行各业的管理信息系统,特别是经济类各种管理信息系统大多数都是用 FOXBASE$^+$ 开发的。

本书系统地、全面地讲解了 FOXBASE$^+$ 的各种函数、命令及其应用,并配备了大量的例题及习题。本书所有函数及例题均在机上调试过,以便于读者学习和理解,并对部分例题给出了程序流程图。特别是本书第九章,以实用管理系统中几个常见模块程序为例,分别进行讲解并给出调试过的程序,以使读者学习后,在进行简单的管理系统开发工作时能走捷径。

全本共分十章,除利用大部分章节讲解 FOXBASE$^+$ 的基础知识以及程序设计外,还在第一章介绍了数据结构的基本知识及几种典型的数据结构;在第八章中介绍了网络环境下的操作;在第十章介绍了 FOXPRO 的基础知识和特点。这为读者在不同环境下使用数据库管理系统打下了良好的基础。

本书内容深入浅出,语言通俗易懂,选题注重实用,可作为大中专学校数据库管理系统教材,也可作为从事计算机应用和开发工作的技术人员的参考书或自学教材,同时,也可以作为计算机过级考试的 FOXBASE$^+$ 部分的参考教材。

编 者

1996 年 4 月 28 日

编　审　说　明

本书是全国财经类通用教材。经审阅，我们同意作为高等财经专科学校试用教材出版。书中不足之处，请读者批评指正。

财政部教材编审委员会

一九九六年九月十三日

前　言

为适应高等财经专科学校会计电算化专业的教学需要，根据财政部颁布的《财政(经)普通专科会计电算化专业教学方案》，财政部培训中心组织编写了会计电算化专业系列教材。本系列教材包括：《计算机基础》、《微型机原理与汇编语言》、《实用软件与信息安全维护》、《FOXBASE⁺关系数据库系统》、《C 语言程序设计》、《管理信息系统》、《会计电算化应用》等七本。该系列教材不仅适合于三年制大专会计电算化专业教学使用，也适合于成人教育及从事会计电算化工作人员学习参考。

《FOXBASE⁺关系数据库系统》由黑龙江财政专科学校张洪瀚主编，并负责对全书的修订、总纂和定稿。全书写作分工如下：第一、六、九、十章由黑龙江财政专科学校张洪瀚编写；第二、三章由广东财税高等专科学校谢虹编写；第七、八章由江西财院九江分院潘杰编写；第四、五章由黑龙江财政专科学校郑德权编写。本书编写时得到了财政部培训中心和兄弟院校的大力支持，在此表示衷心的谢意！

本书及配套的教学大纲，已由财政部教材编审委员会组织的专家评审通过。

本书由东北财经大学张不同教授主审。张不同教授对全书提出了不少有益的建议，在此深表谢意！

编　者

1996 年 6 月

需要全本请访问 www.ertongbook.com

目　录

第一章　数据组织与数据库…… 1

§1.1　数据处理基础知识 …… 1
§1.2　数据的逻辑组织方式 …… 3
§1.3　数据库基础知识…… 10

第二章　FOXBASE⁺ 概述 …… 15

§2.1　FOXBASE⁺ 的主要特点及性能指标 …… 15
§2.2　FOXBASE⁺ 系统版本及运行环境 …… 17
§2.3　FOXBASE⁺ 系统文件类型 …… 20
§2.4　FOXBASE⁺ 系统的安装、启动与退出…… 23

第三章　FOXBASE⁺ 的数据、函数及表达式 …… 26

§3.1　FOXBASE⁺ 的常量及变量 …… 26
§3.2　FOXBASE⁺ 的函数 …… 34
§3.3　FOXBASE⁺ 的运算符及表达式 …… 62
§3.4　FOXBASE⁺ 的命令的使用 …… 66

第四章　数据库基本操作 …… 72

§4.1　数据库文件结构的操作…… 72
§4.2　数据库文件数据的操作…… 83
§4.3　数据库文件的排序、索引及查找…… 101

§4.4 数据库文件数据的统计 …… 111

第五章 数组、多重数据库操作及系统管理 …… 116

§5.1 数组操作及相关命令 …… 116
§5.2 数据库文件之间的操作 …… 124
§5.3 系统管理 …… 135

第六章 FOXBASE⁺ 程序设计 …… 150

§6.1 结构化程序设计特点及程序流程图 …… 150
§6.2 程序的建立与运行 …… 153
§6.3 输入输出及交互式命令 …… 156
§6.4 屏幕格式设计及菜单命令 …… 169
§6.5 顺序结构程序设计 …… 180
§6.6 分支结构程序设计 …… 182
§6.7 循环结构程序设计 …… 198
§6.8 子程序及过程文件 …… 213

第七章 FOXBASE⁺ 其他文件的产生 …… 233

§7.1 文本文件的产生及应用 …… 233
§7.2 报表格式文件的产生 …… 236
§7.3 屏幕格式文件的设计 …… 239
§7.4 标签格式文件的建立 …… 240
§7.5 目标文件的产生 …… 243

第八章 FOXBASE⁺ 的多用户操作 …… 245

§8.1 数据的保护 …… 245
§8.2 死锁处理 …… 249
§8.3 多用户命令和函数 …… 252

第九章 实用程序典型模块设计实例…… 258

§9.1 菜单程序设计 …… 258
§9.2 录入程序设计 …… 267
§9.3 编辑程序设计 …… 271
§9.4 统计程序设计 …… 275
§9.5 查询程序设计 …… 277
§9.6 打印程序设计 …… 282

第十章 FOXPRO 2.5 简介 …… 285

§10.1 FOXPRO 的特点及工作环境 …… 285
§10.2 FOXPRO 新颖功能 …… 287

附录一 FOXBASE⁺ 系统函数索引 …… 291
附录二 FOXBASE⁺ 的命令索引 …… 298
附录三 FOXBASE⁺ 系统参数设置索引 …… 309
附录四 错误信息及其说明…… 314
附录五 INKEY()键值表 …… 324
附录六 READKEY()的键值表…… 326
附录七 ASCⅡ字符表…… 328
附录八 国标区位码表(制表符)…… 331

第一章　数据组织与数据库

本章作为全书的开篇,将概括介绍在数据处理范畴内所涉及到的常用基本概念及数据组织方式。本章从数据结构的角度,简单介绍典型的数据组织方式;从数据库理论的角度,简单介绍数据模型,为后续内容奠定数据处理理论基础。

§1.1　数据处理基础知识

一、数据与信息

(一)数据

数据是指客观实体的属性。它是反映客观实体的性质、形态、结构和特征的一组符号。这组符号可以是数值型和非数值型的。如"这台计算机的 CPU 是 486 的",这里所描述的客观实体是这台计算机,所描述的属性是 CPU,486 就是该属性的值,属性的值就是一个数值型数据。人们就是通过各种属性来认识事物的,同一类事物具有相同的属性,每个具体的事物则通过不同的属性值来与其他的同类事物相区别。又如"这张软盘的密度为高密",这是一个非数值型数据,同时还可以有其他类型的非数值型数据。

(二)信息

信息一词目前还没有一个统一的定义,人们出于不同的研究目的,从不同的角度出发,对信息有不同的解释。但最常用的解释为:信息是对数据加工后得到对使用它的人来说是有用的数据。

(三)数据与信息的关系

数据与信息二者是不可分离又有一定区别的概念。信息是依附于数据而存在的,而处理数据的目的又是为获得蕴含于其中的信息。信息必然是数据,并非任何数据都是信息,有用的数据才是信息。信息是抽象的认识或知识,它不依赖于具体的载体、介质及技术手段。而数据则依赖于具体技术手段,在计算机化的信息系统中往往和计算机系统有关。尽管如此,人们常常不加区别地使用数据和信息这两个词,它们往往也确实不需要加以区别,因为信息与数据是相对而言的。例如,一个系统(S1)的输入,我们称之为数据(DATA1),一个系统的输出,我们称之为信息(INFO1)。而另一个系统(S2)的输入数据要求为 INFO1,输出信息为 INFO2。这样对 S1 来说 INFO1 为信息,对 S2 来说 INFO1 又是数据。因此,对数据和信息两个词在某些情况下不需要严格区分,也可以不加区分地使用,如数据处理也可以说是信息处理。

二、数据处理系统

数据处理是指对数据进行收集、存贮、加工与传递等一系列活动的总和,是把数据从一种形式转换成另一种形式的处理过程。其每一具体的处理步骤都有特定目的,通过数据处理获得反映事物本质的信息。

数据处理系统是集对于数据进行收集、组织、存贮、加工、传递等功能为一体,以处理数据提供信息为目的而建立起来的一个整体。

数据处理系统从处理方式上可分为:批处理系统、联机集中处理系统和分布处理系统。

(一)批处理系统

批处理是将收集的数据成批送到有关部门,经过编码、誊写、平衡再输入计算机处理,最后输出结果。它的优点是可以集中控制、集中处理。它的缺点是处理不及时、书面重复工作现象太多。

(二)联机集中系统

联机的方法是每当输入一个事务时就立即处理。数据直接从源数据通过终端进入计算机进行处理,可立即得到处理后的结果。同时可以通过终端进行数据查询及数据修改等工作。联机处理与批处理相比节省许多书面工作,并可以立即修改错误,缩短数据处理周期,可以立即得到反馈信息等优点。一般有一个数据库系统用以进行数据管理。

(三)联机分布系统

联机分布系统是利用计算机网络通讯方式,在本地工作站对数据进行处理。对数据管理采用分布式数据库管理系统,即除中央处理系统有数据库以外,各工作站都有数据库,处理时不一定要通过中央系统,可由网上各点工作站进行处理。但在处理过程中不一定能把每一个事务全部处理完,可以处理本地能处理的工作,处理不了的交给其他工作站或中央处理系统完成。这种方式可以最大限度地实现计算机系统资源共享,减少通讯的负担。

§1.2　数据的逻辑组织方式

本节所介绍的内容,属于数据结构的知识。学习 FOX-BASE$^+$ 的目的是使读者能具有一定的程序设计能力。如果要想设计出好的程序,就必须了解一些数据结构的知识。

数据结构主要研究三大方面内容:

一是研究数据元素之间抽象化的相互关系,即数据的逻辑结构。常用的逻辑结构分为线性和非线性结构。线性结构主要研究的是表结构;非线性结构研究的是树型结构和图型结构。

二是研究数据在计算机中的实际存贮方式,即数据的物理结构。常用的物理结构有如下四种:

顺序方式:这种物理结构主要用于线性的逻辑结构,它把逻辑上相邻的结点存贮在物理上相邻的存贮单元中,结点之间的关系

需要全本请访问 www.ertongbook.com

由存贮单元的邻接关系来体现。

链接方式:这种方法是将每个结点附加上指针字段,除存贮数据本身以外,还用指针字段存贮逻辑上相邻结点的实际存贮单元的地址。因此,这种物理结构不要求连续的存贮单元。

索引方式:对逻辑结构中所有结点按一定方式排成一个序列,每个结点在序列中都对应一个位置号,这个位置号就是结点的索引。索引的存贮结构就是用结点的位置号来确定结点的存贮结构地址。

散列方式:这种方法是根据结点本身的值和选用的函数来确定结点实际存贮的地址。

对于相同的逻辑结构选择了不同的物理结构,则为不同的数据结构。

三是研究数据结构所要研究的问题。它们是:对于每种结构定义各自的运算,设计出相应的算法,并对算法从正确性、时间、空间和简单性进行评价分析。各种结构常见的运算主要包括:建立、消除、插入、删除、访问、修改、排序和查找等运算。

本节只对数据的逻辑结构进行介绍,关于其他两部分内容读者可以查阅数据结构方面的书籍。下面分别介绍几种常用的数据逻辑结构。

一、表结构

(一)线性表

线性表是 $n(n\geq 0)$ 个具有相同特性的数据元素($a_1, a_2, a_3, \ldots a_i, a_{i+1}, \ldots a_n$)的有限序列。

其中,n 为线性表的长度,当 $n=0$ 时线性表为一个不含任何元素的空表。序列中第 i 个元素为 a_i $(1\leq i\leq n)$,元素 a_{i-1} 为元素 a_i 的前驱元素,元素 a_{i+1} 为元素 a_i 的后继元素。

a_1 为表的第一个元素,称作表头元素,a_n 为表的最后一个元素,称作表尾元素。

在线性表中除表头元素和表尾元素以外,每个元素都有且仅有一个直接前驱元素和一个直接后继元素。而对于表头元素,它没有直接前驱元素,只有一个直接后继元素。表尾元素没有直接后继元素,只有一个直接前驱元素。

线性表常用运算如下:

(1)置一个线性表为空表;

(2)求线性表的长度;

(3)读取线性表中第 i 个元素;

(4)修改或重写线性表中第 i 个元素;

(5)查找线性表中首先满足给定条件的元素;

(6)向线性表中第 i 个元素或满足给定条件的第一个元素所在位置插入一个新元素;

(7)删除线性表中第 i 个或满足条件的第一个元素;

(8)对线性表中的所有元素按某个数据项的值升序或降序重新排序。

线性表的物理结构通常采用顺序存贮和链接存贮。用顺序方式存贮的线性表,我们称作顺序表;用链接方式存贮的线性表,我们称作链表。对于顺序表,它的特点是逻辑上相邻的两个元素在物理位置上也相邻,因此,可以随机访问表中任一个元素,它的存贮位置可用一个简单、直观的公式来表示,并节省存贮空间。但对它做插入和删除操作时,需要移动大量的元素之后才能进行。而链表则能较方便地进行插入和删除操作,但由于它的物理结构所决定,它访问表中任一元素时,不能随机访问,必须从头开始,另外增加存贮指针的空间。因此,在选择结构时,要根据对其要做的重点操作来确定它的物理结构。

(二)栈

栈是限定仅在表尾进行插入或删除操作的线性表。表尾端称为栈顶,表头端称为栈底。不含元素的空表称为空栈,向一个栈插

入新元素称为进栈,它是把新元素放到栈顶元素的上面,使之成为新的栈顶元素;从一个栈删除元素称为出栈,它是把栈顶元素删除掉,使其相邻的元素成为新的栈顶元素。

设栈中元素为:(a_1,a_2,a_3,...,a_i,a_{i+1},...,a_n),其中 a_1 为栈底元素,a_n 为栈顶元素,栈中元素按 a_1 ...a_n 的次序进栈,第一个出栈的元素为栈顶元素。因此,我们又称栈为后进先出的线性表。

常用的栈的运算有:

1 .进栈:向栈顶插入一个新元素;

2 .出栈:删除栈顶元素;

3 .读取栈顶元素;

4 .置空栈;

5 .判断一个栈是否为空;

6 .求当前的栈的长度。

(三)队列

队列是一种运算受限制的线性表。它只允许在表的一端进行插入,这端称作队尾;只允许在表的另一端进行删除元素,这端称作队首。插入元素称作进队;删除元素称作出队。新元素进队后就成为新的队尾元素;从队首删除元素后,其后继元素成为新的队首元素。

设队列中元素为:(a_1,a_2,a_3,...,a_i,a_{i+1},...,a_n),其中 a_1 为队首,a_n 为队尾;队列中的元素是按照从队首 a_1 到队尾 a_n 的顺序进队;出队也是按照这个顺序。由于队列的插入和删除分别在表的两端进行,与栈相反,它是个先进先出的线性表。

常用的队列的运算有:

(1)置空队;

(2)进队;

(3)出队;

(4)取队首;

(5)求已知队列当前所含元素的个数。

以上介绍了线性结构中几种常用的基本结构,当然有时为了某种运算的方便或改善结构的使用方便,可以对这几种基本结构进行扩充,如为改善队列的假溢出现象,可以构造循环队列。另外,串、数组等也属于线性结构。

二、树和二叉树

(一)树型结构

树型结构是一种常用的非线性结构。它是 n(n≥0)个结点的有限集合。当 n=0 时,它是棵空树,在一棵非空树中有且仅有一个结点没有前驱结点,这个特殊的结点被称作根结点;当 n=1 时,这棵树只有根结点;当 n>1 时,其余的结点可分为 M 棵(M>0)互不相交的子树(称作根的子树),每个根子树又同样是一棵树。显然,树的定义是递归的,树是一种递归的数据结构。下面给出一棵非空树的递归定义:

1.树的定义

定义:非空树是包括 n 个结点的有限集合 T(n≥1),使得:

(1)有且仅有一个无前驱结点,这个结点称作根结点;其余每个结点都有且仅有一个前驱结点;包括根结点在内,每个结点可以有任意个后继结点。

(2)除根结点以外的其他结点被分成 M 个(M≥0)不相交的集合 T_1,T_2,...,T_M,而且这些集合的每一个又都是树。树 T_1,T_2,...T_M 被称作这个根的子树。

从这个递归定义来看,可以用子树来定义树。只包含一个结点的树必然仅由根组成,包含 n>1 个结点的树可借助少于 n 个结点的树来定义。

2.树型结构中的术语

(1)结点的度:结点拥有子树的个数为该结点的度。

(2)树的度:树内所有结点的度的最大值为该树的度。

(3)叶结点:度为 0 的结点为叶结点。

(4)分支结点:度大于 0 的结点为分支结点。

(5)孩子结点及双亲结点:每个结点的子树的根(每个结点的后继结点)称作该结点的孩子结点;相应地,该结点称作孩子结点的双亲结点。

(6)兄弟结点:具有同一双亲结点的孩子结点互称兄弟结点。

(7)结点的层数:从树根开始,根结点为第一层,它的孩子结点为第二层,以此类推。

(8)树的深度:树中结点的最大层数称为树的深度。

(9)有序树:若树中结点的各子树是按照一定次序从左向右有次序(不能互换)安排的,则称该树为有序树;反之,则称为无序树。

(二)二叉树

二叉树是一种最简单、最重要的树型结构之一。它的特点是树中每个结点至多只有二棵子树(二叉树中不存在度大于 2 的结点),并且二叉树的子树有左右子树之分,其次序不能互换。因此,二叉树为一棵树的度最大为 2 的有序树。

满二叉树是在一棵二叉树中,当第 i 层的结点数为 2^{i-1} 个时,则称此层的结点数是满的,当树中的每一层都满时,称此二叉树为满二叉树。

完全二叉树是在一棵二叉树中,除最后一层外,若其余各层都是满的,并且最后一层或是满的,或是从右边缺少连续若干个结点,则称此二叉树为完全二叉树。

由于二叉树有些较重要的特性,因此它是一种常用的树型结构。关于二叉树的性质在这不做介绍了。

(三)树的基本操作

(1)初始化操作,置空树;

(2)求结点所在的树的根结点;

(3)求某结点的双亲结点;

(4)求树中某结点的第 i 个孩子结点;

(5)插入子树;

(6)删除子树;

(7)遍历:按某个次序依次访问树中各个结点,并使每个结点只被访问一次。

三、图型结构

(一)图的定义

图是一种复杂的非线性数据结构。一个图 G 是由两个集合 V 和 E 组成,其中 V 是有限的非空顶点(图中的结点称作顶点)集合;E 是用顶点偶对表示的边集合。图 G 的顶点集合和边集合分别记为 V(G)和 E(G),而将图 G 表示为 G = (V, E)。

(二)图中的基本术语

1 .若在图 G 中表示边的顶点偶对是无序的,则称 G 是无向图。在无向图中,顶点偶对 (V_i, V_j) 和 (V_j, V_i) 代表同一条边,表示连接顶点 V_i 和顶点 V_j。

2 .若在图 G 中表示边的顶点偶对是有序的,则称 G 是有向图。在有向图中,用顶点偶对 $< V_i, V_j >$ 表示从顶点 V_i 到顶点 V_j 的边,而用顶点偶对 $< V_j, V_i >$ 表示从顶点 V_j 到顶点 V_i 的边。对于 $< V_i, V_j >$ 称顶点 V_i 为边的始点,而顶点 V_j 为边的终点。

3 .完全图。在任何一个具有 n 个顶点的无向图中,其边数恰好为 n * (n - 1)/ 2 的无向图称为完全图。

4 .有向完全图。在有 n 个顶点的有向图中,其边数为 n * (n - 1)条的为有向完全图。

5 .顶点的度。对于无向图来说,顶点 V 的度为以该顶点为一个端点的边的数目。对于有向图来说,顶点 V 的度为入度和出度之和。所谓入度是以该顶点为终点的边的数目;出度是以该顶点为始点的边的数目。

6 .路径。它是从一个顶点到另一个顶点边的序列,路径的长

需要全本请访问 www.ertongbook.com