

会计电算化试用教程（二）

关系数据库实用语言 与程序设计

陈宏明主编

湖南省财政厅审定

湖南科学技术出版社

序 言

会计电算化又称计算机辅助财务管理,是会计工作现代化的一个重要标志,也是会计改革的方向。随着改革的不断深化,会计电算化已经在会计领域中广泛推行,它越来越为人们的认识所接受,并取得了良好的社会效益。因此学习会计电算化知识是会计人员知识更新的需要,也是会计工作适应时代发展的需要。可以说,今后随着计算机的普遍推广应用,不懂得和不熟悉会计电算化的会计人员将是不合格、不称职的会计人员。为此,我们特组织编写了《会计电算化试用教程》,以此来提高会计人员的素质,推动会计电算化事业的发展。

本套教程共四本。之一《微型计算机实用技术》讲述计算机的基本组成,DOS3.3的使用方法,汉字系统2.13的操作方法,是会计电算化的入门教材。之二《关系型数据库实用语言与程序设计》讲述会计电算化的工作语言FoxBASE+的基本原理和一般会计电算化程序的编制方法,是会计电算化的基本教材。之三《会计电算化实用范例》解剖会计电算化系统中最常用的凭证汇总、帐务管理系统、报表管理系统,分析会计电算化实用系统的编制方法,使会计人员能够学以致用,是提高会计人员编写会计电算化系统程序水平的深入教材。之四《上机操作指南》配合前三本教材,讲述如何上机实现会计电算化的全过程,指导电算会计岗位人员进行系统管理,维护及各种基本技能训练。本教程以广大会计人员为读者,深入浅出,通俗易读,内容连贯,自成体系,适合于不同层次的会计人员学习会计电算化知识,是学习会计电算化的系列教材。为了读者能够更好的理解教程内容,我们将教程之一和之二录制了教学录像带,并且将教程之二和之三的程序和数据库录入软盘配合发行。湖南省财政厅指定本教程为全省各种层次和形式的会计电算化岗位培训的统一教材。

全套教程由陈宏明负责总纂并审定。在编写过程中得到了湖南省财政厅、湖南广播电视大学、长沙水利电力师范学院、湖南省计算机专科学校等单位 and 领导的大力支持,在此一并表示诚挚的感谢。

《会计电算化试用教程》编写组

1992年3月于长沙

前 言

会计电算化最常用的实用语言,现已转向关系型数据库 FoxBASE+。它与 dBASE III PLUS 完全兼容,但速度快 5.9 倍,比编译 dBASE III 快 2.3 倍,甚至比最新推出的 dBASE IV 还要快,是当今市场上最快的 dBASE 兼容系统。它具有极高的灵活性,适合于在变化很宽的操作系统环境(从 MS DOS 到 UNIX、XENIX)和硬件(从 IBM PC 到 VAX11/780)下运行,而且 FoxBASE+ 的单用户与多用户版本完全兼容。它增加了数组功能,可以使一个数据库同时与多个数据库建立关系,可以同时打开 48 个文件,等等。由于 FoxBASE+ 的优越性和与 dBASE 的兼容性, FoxBASE+ 已经作为会计电算化系统开发应用的工作语言。

我们根据多年来在会计专业、财务计算机培训班和会计电算化培训班讲授“会计电算化关系数据库”和“会计电算化程序设计”课程的经验和研制会计电算化系统软件成果(其中有的通过能源部部级鉴定,被认为达到国内先进水平,并处于国内同类研究领先水平)经验的基础上,编著了本书。目的是使会计人员学会运用 FoxBASE+ 编写会计电算化应用程序。书中提供了学习编写程序所必须了解的基本知识及编程方法。全书紧密结合会计电算化的实例,内容深入浅出,循序渐进,分析详细,从理论到实际,自成体系,特别适合于自学。

本书第一、二、三、四章及附录由柳见成编著;第五、六、七、八章由陈宏明编著。陈宏明主编并最后修改定稿。

全书所举的实例均在 IBM PC/XT、286、386 及其兼容机上,在 25 行 FoxBASE+2.00 下运行通过,为了读者更好地使用,已将本书中的所有程序及数据库录入软盘,配合发行。

在本书的编写过程中,得到了湖南省财政厅、长沙水利电力师范学院领导、公共课部和计算中心的大力支持,在此表示诚挚的感谢。

由于作者水平有限,错误之处,在所难免,欢迎批评指正。

作者

1992 年 1 月于长沙

目 录

第一章 汉字 FoxBASE+基础

§ 1.1 关系型数据库管理系统概论	(1)
§ 1.2 FoxBASE+的启动与退出	(8)
§ 1.3 数据类型与文件类型	(9)
§ 1.4 命令的语法规则	(13)
§ 1.5 表达式	(14)
§ 1.6 函数	(19)

第二章 数据库基本操作

§ 2.1 定义数据库结构	(29)
§ 2.2 建立数据库结构	(31)
§ 2.3 输入记录	(32)
§ 2.4 打开与关闭数据库文件	(35)
§ 2.5 记录定位与插入	(36)
§ 2.6 显示记录	(39)
§ 2.7 修改记录	(42)
§ 2.8 修改数据库结构	(46)
§ 2.9 删除记录	(48)
§ 2.10 记录求和与统计	(51)

第三章 数据库管理操作

§ 3.1 选择工作区	(54)
§ 3.2 复制数据库文件	(55)
§ 3.3 排序与索引	(62)
§ 3.4 检索记录与分类求和	(68)
§ 3.5 关联数据库	(73)
§ 3.6 更新数据库	(78)
§ 3.7 合并数据库	(79)
§ 3.8 过滤数据库	(80)
§ 3.9 内存变量数组与内存变量文件	(85)

第四章 FoxBASE+的运行环境

§ 4.1 FoxBASE+的参数设置命令	(90)
§ 4.2 FoxBASE+的配置文件	(100)
§ 4.3 FoxBASE+下的操作系统管理功能	(103)

第五章 程序设计基础

§ 5.1 程序概述	(105)
§ 5.2 人机交互命令和简单程序	(108)
§ 5.3 分支结构程序设计	(113)
§ 5.4 循环结构设计方法	(120)
§ 5.5 过程和自定义函数设计方法	(132)
第六章 格式程序设计方法	
§ 6.1 屏幕格式设计方法	(141)
§ 6.2 设计格式文件控制屏幕的方法	(148)
§ 6.3 报表格式文件	(149)
§ 6.4 打印输出程序的设计方法	(154)
§ 6.5 分页打印表格的设计方法	(158)
第七章 常用模块设计和调试	
§ 7.1 菜单模块	(162)
§ 7.2 口令模块	(165)
§ 7.3 程序的测试与调试方法	(166)
§ 7.4 FoxBASE+调试技术	(171)
§ 7.5 过程的连接与程序的编译	(185)
第八章 会计电算化程序综合设计方法	
§ 8.1 系统分析	(187)
§ 8.2 系统设计	(190)
§ 8.3 FoxBASE+设计方法	(204)
§ 8.4 数据库的设计方法	(208)
§ 8.5 模块的分析设计方法	(211)
附录 A FoxBASE+命令一览表	(234)
附录 B FoxBASE+函数一览表	(244)

第一章 汉字 FoxBASE+基础

§ 1.1 关系型数据库管理系统概论

一、数据库与数据库系统

数据库系统是现代社会信息量急剧增加提出的要求与计算机软硬件技术高速发展两方面相结合的结果,数据库技术是六十年代初提出来的,从七十年代以来,数据库系统的理论、技术与方法已日趋成熟。虽然建立一个理想的数据库系统十分复杂和困难,但由于社会的需要和计算机性能价格比的迅速提高,数据库系统在实用上已经相当广泛,正日益渗透到社会各行各业的领域中去。

信息一般被定义为向人们(或机器)提供的关于现实世界中各种事实的知识。现实世界中充满了各种信息,计算机在处理信息时,需要将外界的信息转换为计算机能识别的符号,也就是转换为数据。

数据是泛指一切用于描述事物并且可以为计算机所识别、存储和处理的符号。数据可分为两大类。一类为数值型数据,如价格、工资、产量等数字,它们具有可计算性。另一类为非数值型数据,如文字、图形、声音和各种特殊符号等。因此,信息和数据是不可分离而又有一定区别的概念。例如,你从气象预报中得知明天将下雨,这是你所获得的一个信息,但只有当你用一定的数据符号形式把它具体表示出来,才能由计算机进行处理。时至今日,几乎除暗示以外,所有的信息均可表示成计算机能识别的数据形式,予以存储、传递和各种处理。

数据处理是对数据进行收集、储存、加工、检索和传递等一系列过程的总和。数据处理的目的从大量的、没有条理的数据中抽取并推导出对某些特定的目的有价值的数,借以作为人们决策的依据。

数据处理技术发展至今,已经经历了三个阶段。即人工管理阶段、文件管理阶段和数据库管理阶段。

1. 人工管理阶段

五十年代中期以前,计算机问世还不久,当时外存储器的容量很小,软件上也还没有产生成熟的操作系统。计算机主要用于数值计算,一般不需要将数据长期保存。数据的存储结构要由程序员在程序中设计。因此,一方面程序与数据相互依存,数据没有独立性,修改数据势必修改程序,同时不同程序中的重复数据很多,造成存储空间的浪费。另一方面程序员必须在设计程序时仔细考虑数据存储的物理结构和逻辑结构,操作很不方便。图 1-1 反映了上述特点。

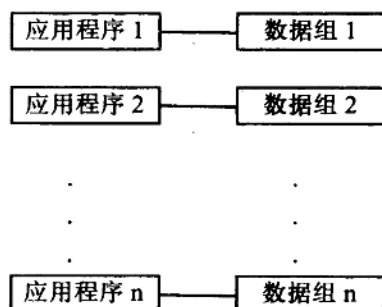


图 1-1 人工管理系统

2. 文件管理阶段

从五十年代后期开始, 外存技术中出现了磁盘等可以直接存取的存储设备, 系统软件中也出现了操作系统和文件管理系统。在操作系统支持下, 利用文件管理系统可以将数据以文件形式长期保存在外存储器上, 并可以反复对文件进行查询、修改、增删等操作。例如, 要建立一个单位的人事管理系统, 可分别建立如下三个数据文件。

档案文件							
编号	姓名	性别	年龄	政治面貌	工资		

业务文件							
编号	姓名	性别	年龄	所学专业	学历		

健康状况文件							
编号	姓名	性别	年龄	健康状况	病史		

图 1-2 人事管理文件结构

在文件系统管理下, 程序和数据之间具有一定的独立性。程序员设计时只须考虑数据的逻辑结构, 而数据在存储器上的存储位置 (即数据的物理结构) 和其它细节则都由文件系统处理, 从而大大节省了设计程序的工作量。文件管理系统的特点如图 1-3 所示。

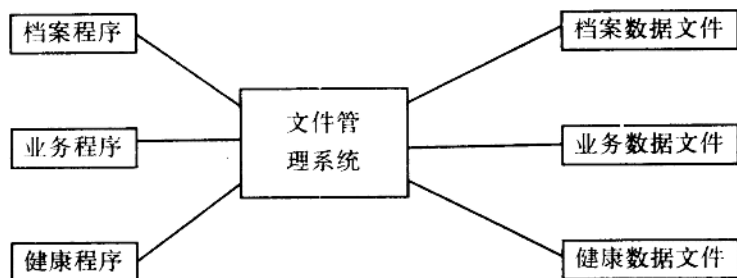


图 1-3 文件管理系统特点

文件管理系统仍然有许多不足。各个数据文件为对应的程序所“专有”, 是根据某一应用要求组织起来的, 其它程序用不上。各数据文件处于松散的状态, 数据与程序的独立性不强, 数据文件之间的数据冗余仍然较大, 既浪费了存储空间, 又为数据管理带来了困难。例如, 当有人调出时, 必须修改所有的数据文件, 稍有疏漏, 就会造成数据不一致性问题。

3. 数据库管理阶段

六十年代后期以来, 由于计算机大量应用于数据处理、情报检索、办公室自动化和财务管理等领域, 所处理的数据大多为非数值型数据。非数值型数据在结构上比数值型数据复杂得多, 不仅要表示各部分数据的内容, 而且要反映出各部分数据之间的联系。另一方面, 随着计算机软硬件技术的迅速发展, 出现了大容量的磁盘存储器, 为集中存储和管理大量数据提供了物质基础, 以计算机为核心的信息处理系统开始逐渐形成。数据库技术就是在这种背景下产生和发展起来的。

要给数据库 (Data Base) 下一个确切和完整的定义很困难, 而且有很多种。《英汉计算机词典》对数据库的定义是: 数据库是计算机存储设备上合理存放的相互关联的数据的集合。这些数据具有如下特点:

- (1) 最小冗余, 即数据以最大可能不重复。
- (2) 资源共享, 即以最优方式服务于一个或多个应用程序。

(3)数据独立，即数据的存放独立于使用它的应用程序。

(4)统一管理，用一个软件统一进行各种基本操作。

例如图 1-3 所示的人事管理系统可用数据库组织如下：

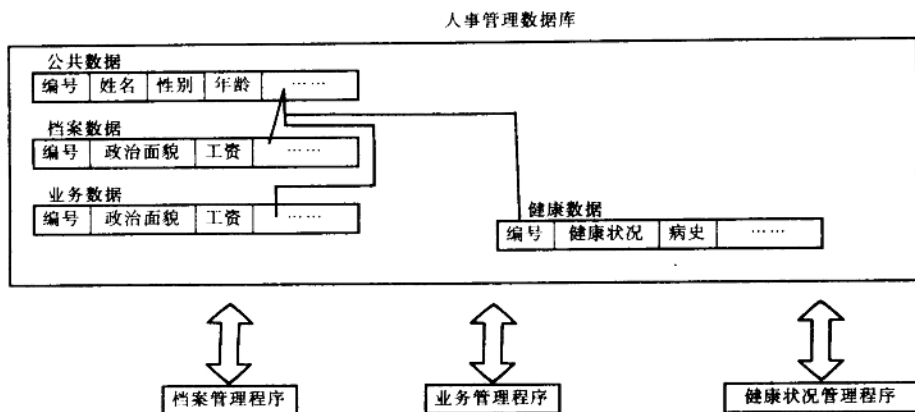


图 1-4

根据图 1-4，我们可稍详细地讨论数据库的如上几个特点。首先，在数据库中排除了数据的冗余性，能将重复的公共数据去掉。当有人调出或调入时，只要修改一次公共数据即可，但需要对数据进行统一管理。其次，数据具有独立性。在数据库中，数据由管理软件统一管理，应用程序不再负责。因此，当修改数据的结构时，不会影响程序，而程序的修改也可单独进行。这样不仅易于修改应用程序，也大量节省了程序员原来花在设计数据结构、建立和维护文件上的时间和精力。第三，数据库实现了数据的共享。不仅现行数据库系统的文件可综合使用，新建的系统也可使用已经存在的数据库。

由于计算机的共享一般是并发的，即可能有许多用户同时使用，因此数据库的管理必须具有三种控制功能。①安全性。即采取必要的措施（如设置口令等）防止不合理地使用数据库，或恶意破坏和非法存取数据。②完整性。完整性是指数据的正确性、有效性、相容性。数据库应具有检验的功能，以控制数据在一定范围内方为有效，或这一部分数据与另一部分数据之间必须满足一定的关系等。③并发控制。即避免并发程序之间的相互干扰，协调用户使用数据上的冲突。

综上所述，数据库是通用化的综合性的数据集合，具有多种功能。必须有一个软件系统也就是数据库管理系统（Data Base Manage System）在建立、运行和维护时对数据库进行统一管理。数据库系统的关系可如图 1-5 所示。

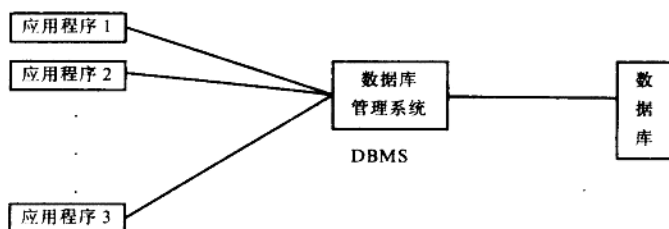


图 1-5 数据库系统组成

二、数据模型

在建立数据库系统时，首先必须对客观事物及其联系进行分析。通常用“实体”这一概念来描述客观事物。实体可以是实际的东西，如一个人、一个零件、一笔金额，也可以指抽象的事物，如一次订货、一次借款等，还可以指事物与事物之间的联系，如订货单位与供货厂家之间的订货关系。

每个实体都具有若干属性，实体是通过它的属性来描述的，因此实体就是相关属性的组合。例如：编号 / 101、姓名 / 潘易萍 /、性别 / 女、年龄 / 23、政治面貌 / 团员、学历 / 高中等属性的组合就表示了“潘易萍的档案”这样一个实体。

数据模型是对客观事物的实体及其联系的数据描述。它反映出数据在数据库中排列和组织所遵循的规律。数据模型把数据逻辑地组织成数据库，使用户能有效地存取和处理数据。当前流行的数据模型有层次模型、网络模型和关系模型三种。

1. 层次模型

层次模型具有树型结构的特点。层次模型由结点和连线组成。结点表示实体，连线表示实体之间的联系。在现实世界中，许多实体间的联系本身就是一个自然的层次关系。例如，图 1-6 反映了一个学校行政组织的层次模型。

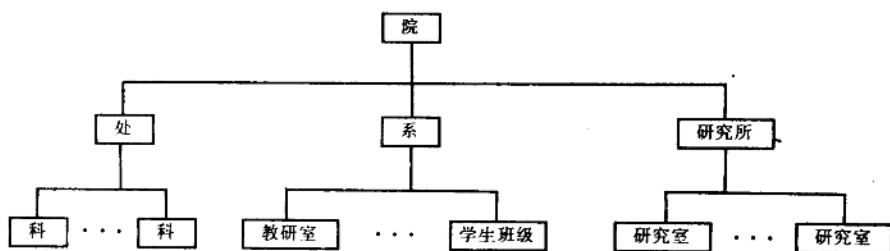


图 1-6 层次模型实例

层次模型中最上一层结点称为根，最下一层结点称为叶，其余结点称为中间结点。如果要查询某一结点，必须按照从根开始的某一条路径进行。除根以外每个结点向上只能有一个父结点，除叶以外每个结点向下可有多个子结点。

2. 网络模型

网络模型和层次模型本质上是一样的，只是取消了层次模型中父子结点的限制。在网络模型中任何两个实体之间都可以建立基本的联系，因此结构更为复杂，不像层次模型那样清晰、简单。图 1-7 是一个反映购货与领料关系的网络模型。

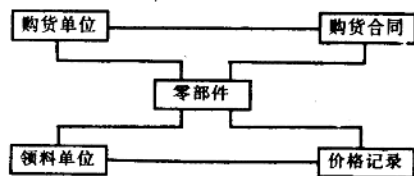


图 1-7 网络模型实例

3. 关系模型

领料单位层次模型与网络模型属于构造性的格式化模型，使用了图论的概念，用结点和连线表示实体及其相互之间的联系。而关系模型则是以关系代数理论为基础，把数据模型看

成是关系的集合。

在关系模型中所有数据都被组织成一种二维表的形式，一个二维表在数学上称为一个关系。在二维表中存放两类数据：实体本身的数据及其实体之间的联系。这种不是人为地设置而是由数据本身自然地建立起它们之间的联系，就是关系模型的本质。

现以三个数据库文件为例说明关系模型的特点。这三个数据库由供应厂表 G、零件表 L 和仓库表 GL 三个关系组成。见图 1-8。

供应厂表 G			零件表 L				仓库表 GL		
厂名	厂址	电话	零件号	零件名	价格	存放地点	零件号	供应厂名	储存量
AA	长沙	21212	L1	轴承	120	一号仓库	L1	AA	100
BB	株洲	31313	L2	齿轮	250	二号仓库	L2	AA	100
CC	零陵	41414	L3	螺栓	8	一号仓库	L1	BB	300
							L2	CC	100
							L3	CC	500

图 1-8 关系模型实例

在描述关系的二维表中，每一行称为一个记录，对应于一个实体；每个记录又由若干个字段（对应表上的列）组成，每个字段描述了该实体的一个属性。例如零件表 L 这一关系具有四个字段，分别描述各个零件的零件号、零件名、价格、存放地点四个属性。从图 1-8 不难看出，各个关系之间的联系都可以通过它们共同的属性（称为关键字）表示出来。例如：

供应厂表 G 和仓库表 GL 之间有一共同属性——供应厂名；

零件表 L 和仓库表 GL 之间有一共同属性——零件号。

图 1-8 中“仓库表 GL”关系着“供应厂表 G”和“零件表 L”这两个关系。所有关系的集合称为一个关系数据库（一个关系也可组成一个最简单的关系数据库）。关系定义的集合就是关系模型。在关系数据库中可以对记录进行查询、增加、删除等各种操作。

例如要找出提供轴承的是哪些厂家，可从零件表 L 中查到轴承的零件号，再从仓库表 GL 中查找对应的各个厂名。如需要还可根据厂名从供应厂表 G 中查出各厂家的地址等。

表格是日常工作中最常见的、直观的记录形式，然而只有满足下列要求的表才能称为二维表。

- (1)表中每一列的属性值是相同类型的数据。
- (2)列的顺序可以任意。
- (3)行的顺序可以任意。
- (4)表中每一列的属性是不可再分割的数据项。
- (5)表中任意两个记录不能完全相同。

符合上述要求的关系称为规范化的关系，它可消除表中数据重复引起的冗余，使关系中的属性含义清楚、单一，解决操作的可行性和提高用户操作的方便性。

由此可见，不是任何表格都可看作二维表。例如图 1-9 为一张财务收款凭证表，显然不符合上述要求。

收款凭证

19__年__月__日

借方科目: 银行存款

总字第4号

银收字第1号

摘 要	贷方科目	帐 页	金 额	
国家拨入 流动资金	流动基金		一级科目	二级或明细科目
			50,000.00	
合 计			50,000.00	

会计主管 记帐 出纳 审核 填制

图 1-9

对于非规范化二维表,应将其转化为符合要求的二维表,才可由关系型数据库所接受。本书建立一个反映财务记帐凭证关系的数据库,该数据库对应的二维表如图 1-10 所示。每笔凭证对应于凭证库中的一条记录,每条记录包含 7 个属性(即字段):月份、日期、凭证编号、摘要、科目代号、记帐方向和金额。

月份	日期	凭证编号	摘 要	科目代号	记帐方向	金额
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

图 1-10

4. 关系型数据库的几个基本概念

(1) 字段: 字段是数据库中可处理的最小数据单位,它可由一个或多个字节组成。字段有字段名和字段值两个方面。例如记帐凭证关系中科目代号是一个字段名,它在每笔不同的凭证记录中可能取不同的值。

(2) 记录: 记录是由反映各种不同属性的字段组成的一组数据。记录是数据库系统组织和存取数据的基本单位。例如记帐凭证库中一笔凭证的完整描述就构成一条记录。

(3) 文件: 文件是由若干同类记录组成的一个集合。文件是磁盘存取操作的基本单位,每个关系(或者说每张二维表)都以一个数据库文件的形式存于磁盘。关系型数据库管理系统提供的操作可以在上述三个层次上进行,即可以对文件、对记录或对字段实施操作。

(4) 关键字: 关键字是用来识别一条记录的一个字段或几个字段的组合。关键字应尽可能具有唯一标识性和无冗余性。例如当日记帐凭证库中的月份、日期字段就不能作为关键字,而科目代号、凭证编号可以作为关键字。在关键字有可能重复的情况下,还可以指定辅助关键字。例如可以科目代号为关键字,再加上金额字段组成复合关键字。

(5) 工作区: 关系型数据库往往需要同时使用几个数据库文件,为了使这些同时在用的

文件不至于互相干扰，通常采用不同的数据工作区把它们分割开来。FoxBASE+的不同工作区可以用编号 1~10 或字符 A~J 来区分。当前操作的数据库所在的工作区称为当前工作区，进入 FoxBASE+时当前工作区的编号默认为 1。

三、从 dBASE 到 FoxBASE+的发展

支持上节所述的三种数据模型的数据库管理系统分别有层次型数据库管理系统、网络型数据库管理系统和关系型数据库管理系统。其中关系模型由于以直观易懂、便于接受的二维表形式描述数据，且具有数据独立性强和数学理论严格等特点，应用最为广泛。据统计世界上正在运行的数据库系统中 80% 以上为关系型数据库管理系统。

微型计算机上使用的数据库系统并不追求十分完备，而以易学、实用和操作方便为主要目标。dBASE 是国内外流行最广、发展最快的微机关系型数据库管理系统。dBASE 的第一代产品是 1981 年由美国 ASHTON-TATE 公司推出的 dBASE II，受到社会的普遍欢迎，被称为“大众数据库”。我国于 1982 年对 dBASE II 进行了汉化，至今仍有少量的 dBASE II 在运行。1984 年 ASHTON 公司在对 dBASE II 作了相当大的改进和扩充后推出了 dBASE III 系统，后又于 1986 年推出了 dBASE III 的网络版本 dBASE III PLUS。dBASE III 能在 IBM PC 系列十六位微机及其兼容机上运行，与 dBASE II 相比，在处理速度、数据库容量、计算精度等技术指标上大为增强，具有丰富的函数和数据类型，以及较完善的程序控制结构命令。dBASE III PLUS 不仅与 dBASE III 完全兼容，而且增加了在网络环境下运行的功能，并对原有的许多命令和函数作了增强，以及增加了不少新的命令与函数。dBASE III 和 dBASE III PLUS 的汉化版本在我国开发出来后，在各个领域中得到迅速的普及和应用。目前在国内微机运行的数据库大多数是由这两种数据库管理系统支持的。

FoxBASE+是由美国的 Fox 公司于 1987 年推出的多用户关系型数据库管理系统。FoxBASE+与 dBASE III 和 dBASE III PLUS 完全兼容，也就是说，在 dBASE III 或 dBASE III PLUS 支持下开发的应用软件与 dBASE III PLUS 的所有命令都可以原封不动地在 FoxBASE+环境下运行。除此以外，FoxBASE+比 dBASE III PLUS 的工作速度快 5~6 倍，并且对 dBASE III PLUS 的许多命令作了功能上的增强，扩充了不少新的命令，从而使系统更为完善和实用。

FoxBASE+在软件上能适应从单用户操作系统 MS-DOS2.0 到多用户操作系统 MS-DOS4.0、XENIX、UNIX 等多种环境，在硬件上能运行于从 XT、286 及兼容的十六位微机到 80386、80486 等 32 位微机，既能在单机上运行也能在网络多用户环境中使用。FoxBASE+目前在我国已有多种汉化版本，这是一个很有应用前途的微机关系型数据库管理系统。

本书介绍汉字 FoxBASE+的主要功能与程序设计方法。由于 FoxBASE+与 dBASE III PLUS 100% 兼容，因此本书介绍的命令绝大多数都能在 dBASE III 和 dBASE III PLUS 环境下直接运行。对于 FoxBASE+的新增命令、函数和主要的扩充功能，书中将给予适当的说明。

现将 dBASE II、dBASE III、dBASE III PLUS 和 FoxBASE+的主要技术性能指标列表如下，供读者参考。

表 1-1

性能指标	dbase II	dbase III	dbase III+	FOXBASE+
每个数据库最大记录数	65535	10 亿个	10 亿个	10 亿个
每个记录最大字段数	32	128	128	128
每个记录最多字符数	1000	4000	4000	4000
每个字段最大长度	254	254	254	254
可同时运行的内存变量数	64	256	3600	3600
内存变量可占用的字节数	1563	6000	6000	64K
数值精度	10 位	15 位	15~16 位	16 位
可同时打开的主加文件数 *	2	10	10	10
可同时打开的各类文件数	15	15	15	48
可同时使用的工作区数	2	10	10	10
字段类型数	3	5	5	5
内存变量数组	无	无	无	3600 个
用户自定义函数	无	无	无	有
网络功能	无	无	有	有

* 如果使用了 MEMO 字段, 则一个数据库文件按两个文件计算。

§ 1.2 FoxBASE+的启动与退出

汉字 FoxBASE+的软件包括两张 5.25"软磁盘, 主要内容如下:

- 1#盘: MFoxPLUS.EXE 执行文件 247808 字节
 FoxPCOMP.EXE 编译文件 72480 字节
 FoxBIND.EXE 过程连接文件 24496 字节
 2#盘: MFoxPLUS.OVL 覆盖文件 138032 字节
 FoxPHELP.HLP 帮助文件 149909 字节

运行 FoxBASE+之前应先将上述文件拷贝到硬盘上的子目录 Fox 下, 子目录可由用户自己建立。汉字 FoxBASE+有 11 行屏幕显示和 25 行屏幕显示两种版本, 本书采用 25 行版本, 但除屏幕格式设计命令和极少数编辑键操作以外, 其余命令对 11 行版本同样适用。

一、启动 FoxBASE+

在 DOS 操作系统提示符下键入 FoxBASE+的执行文件名。

C>MFOXPLUS

屏幕上出现圆点提示符, 即表示进入了 FoxBASE+状态, 可以接受用户输入的命令。在圆点提示符下, 系统对用户键入的符合 FoxBASE+语法规则的命令进行解释和执行, 并在屏幕上给出执行的结果。

例: .?120-19

101

在本书以后的举例中, 圆点提示符后面的内容为需要键入的命令, 下面则为屏幕显示的运行结果。

二、退出 FoxBASE+

在圆点提示符下，用户可以建立新的数据库或打开已存在的数据库。数据库在内存打开后，里面的数据有可能因意外断电或误操作而遭到破坏。因此在数据库操作完成后，应先退出（也称为关闭）FoxBASE+状态，回到操作系统提示符下再关电源。

退出命令为 QUIT

.QUIT

FoxBASE+ normal shutdown

C>

§1.3 数据类型与文件类型

一、数据类型

数据库中的数据按其构造、处理方式和用途等基本属性，分为若干不同的类型。数据库文件操作的基本原则之一就是只有相同类型的数据之间才可以进行各种操作。因此，数据类型 (Data Type) 是一个非常重要的概念。

FoxBASE+的数据类型有五种：数值型、字符型、日期型、逻辑型和备注型。这些类型的构成和使用规则说明如下。

1. 数值型 (Numeric)：类型代号为 N。数值型数据只能由阿拉伯数字、小数点和正负号构成。数值型数据之间可进行数学运算。数值的最大精度为 16 位有效数字。

2. 字符型 (Character)：类型代号为 C。字符型数据可由一切可打印的文字和符号组成。字符型数据之间不可进行数学运算，但可以用“+”和“-”号把几个字符型数据连接起来，还可以按字符型数据的 ASCII 码进行大小比较。一个字符型数据包含的字符个数称为它的长度（或宽度），一个字符型数据的最大长度为 254。要注意的是一个汉字相当于两个字符的长度。

3. 日期型 (Date)：类型代号为 D。日期型数据用于存放表示日期的数据。系统默认的日期格式是美国格式“月/日/年” (mm/dd/yy)，日期型数据长度固定为 8 个字符。两个日期型数据之间可以作减法运算，结果为数值型数据。日期型数据可以加或减一个数值型数据，结果仍为日期型数据。这些运算中的数值型数据，均表示两个日期间相差的天数。

4. 逻辑型 (Logic)：类型代号为 L。逻辑型数据只有两个值，即逻辑真与逻辑假。它的长度固定为 1。逻辑真可用 .T.、.t.、.Y.、.y. 表示，逻辑假可以用 .F.、.f.、.N.、.n. 表示。逻辑型数据之间只能进行各种逻辑运算。

5. 备注型 (Memo)：类型代号为 M。由于一个字符型数据最大长度为 254，但在实际工作中往往需要在一个字符型数据中保存更多的信息。因此，FoxBASE+采取在数据库文件之外附加一个备注文件的方法，来存放这些长度较大又不等长的字符信息块，该数据的类型即为备注型，长度固定为 10，仅用于存放字符信息块的地址指针。备注型数据可以编辑、显示和打印，但不能作任何运算。

除以上 5 种数据类型外，FoxBASE+还增加了一个屏幕型的数据类型，类型代号为 S。S 型数据可用于保存屏幕显示的内容，但只能在字符显示方式下使用，在图形工作方式下无效。

在 FoxBASE+中，数据类型是可以相互转换的。不同类型的数据之间，通过类型转换后便可进行运算或操作。FoxBASE+主要是通过一些类型转换函数来实现这种转换。在实

用中多为数值型与字符型之间、日期型与字符型之间的相互转换。

二、常量与变量

常量就是在操作过程中保持不变的数据。变量是其内容可能要变化的数据。

1. 常量

FoxBASE+常量的数据类型可以是数值型、字符型、日期型或逻辑型，但不能为备注型。

(1)数值型常量：数值型常量可以是整数或小数，不可用分数，但可用科学记数法表示。例如-9、654、123.456、1.23407E3等，其中1.23407E3表示1.23407乘以10的三次方，即用普通记数法表示的1234.07。

(2)字符型常量：字符型常量是用单引号、双引号或方括号等定界符括起来的数据，又称为字符串。字符串的定界符必须成对匹配，例如“数据库管理系统”、‘FoxBASE’、[123.45]等。如果某种定界符本身也是字符串的内容，则需要用另一种定界符括起，如“a[NAME]”等。应当指出的是，“”也是字符串，它表示长度为零的空字符串。

(3)日期型常量 日期型常量用于表示日期。在FoxBASE+中日期型常量无法直接表示，一般是用字符串通过转换函数CTOD()来表示的。如CTOD(“05/17/91”)表示1991年5月17日。

(4)逻辑型常量：逻辑型常量只有两种值，即逻辑真与逻辑假。逻辑真用.T.、.t.、.Y.、.y.表示，逻辑假用.F.、.f.、.N.、.n.表示。

2. 变量

FoxBASE+的变量分为字段名变量和内存变量两种。

(1) 字段名变量

字段名变量也就是二维表中的字段名。字段名变量的名称可以由字母、汉字、数字和下划线组成，必须以字母或汉字开头，名称的长度不能超过10个字符。

字段名变量的类型有数值型、字符型、日期型、逻辑型和备注型五种。

①数值型：整数或小数，总长度为1~19位。如为小数，小数点也占一位，小数位数为0~15位，而且至少应比字段长度少两位。数值型字段的初始值为0。

②字符型：长度为1~254个字符。字符型字段的初始值是长度等于该字段长度的空格字符串。

③日期型：长度固定为8个字符。日期型字段的初始值不表示任何日期，它的显示形式为“/ /”。

④逻辑型：长度固定为1。逻辑型字段的初始值是.F.。

⑤备注型：长度固定为10。备注型字段的实际内容存放在备注文件(.DBT)中，在字段中的只是该内容的地址，由于该地址对用户是透明的，因此显示时仅以“MEMO”表示。应注意的是，备注型数据的输入、修改不同于其它类型，并且只能进行读写操作，不可参与任何运算。

字段名变量只能存在于数据库中，是在生成数据库时建立的。一个数据库文件中的字段名变量最多为128个，而且它们的总长度不得超过4000个字节。

对于一个数据库文件中的某一个字段，该数据库文件有多少条记录，就有多少个值，因此字段名变量是一种多值变量，它的值随记录的不同而异。

(2)内存变量

内存变量是一种临时性地存储某些中间运算结果和信息的存储单元，它独立于数据库文件而存在，在使用时可随时建立。内存变量的数据类型共有4种：数值型、字符型、日期型、逻辑型，其规定与字段名变量相同。字段名变量的数据类型必须在建立数据库时预先进行定义，而内存变量的数据类型则是在对内存变量赋值时确定的。

①内存变量的赋值

对内存变量赋值主要有如下两种命令格式，在进行赋值的同时也就确定了该变量的内容和数据类型。

格式一： STORE <表达式> TO <内存变量表>

格式二： <内存变量> = <表达式>

赋值时首先计算表达式的值，然后将表达式的值赋给指定的内存变量。格式一与格式二的区别在于格式一可以将一个表达式的值同时赋给多个变量，而格式二每次只能给一个内存变量赋值。

例：. STORE "会计电算化" TO HJ

. STORE 123.456 TO S.T

. P="微型计算机应用"

. A=.T.

. B=3*4+6/2

除以上方法外，还可通过 ACCEPT、INPUT、WAIT、SUM、COUNT 等操作命令建立内存变量。内存变量一种临时存储单元，一旦断电或关机就会消失。如果要将内存变量长期保存，必须用 SAVE 命令将当前建立的内存变量以文件形式存到磁盘上。FoxBASE+在内存变量的基础上还扩充了内存变量数组功能，从而为程序设计带来了很大的灵活性。内存变量数组和内存变量文件的使用方法详见第三章 § 3.9 节。

②显示内存变量

格式： LIST MEMORY [TO PRINT]

或： DISPLAY MEMORY [TO PRINT]

例：. LIST MEMORY

HJ	Pub	C	"会计电算化"	
S	Pub	N	123.456 (	123.45600000)
T	Pub	N	123.456 (	123.45600000)
P	Pub	C	"微型计算机应用"	
A	Pub	L	.T.	
B	Pub	N	15.00 (	15.00000000)

6 variables defined, 46 bytes used

251 variables available, 5954 bytes available

此命令列出当前内存中建立的全部内存变量的变量名、变量内容、变量的数据类型以及所占的字节数，Pub 表示该变量为公共内存变量。

三、文件类型

FoxBASE+的各类数据和程序都是以文件的形式存储在磁盘上，通过操作系统的文件

管理功能和 FoxBASE+ 本身的文件管理功能进行存取和使用。FoxBASE+ 一共设置了 13 种类型的文件。按这些文件的组织形式、数据特点和用途,大致可分为结构式和非结构式两大类。结构式文件主要是数据库文件和索引文件,非结构式文件主要是命令文件、屏幕格式文件和文本文件等。所有这些文件中,数据库文件和命令文件是最重要的,也是应用最多的两种文件。FoxBASE+ 中大多数操作都与这两种文件有关。

FoxBASE+ 对各种类型文件的文件名规定,基本上与操作系统的规定相同,但要稍严格一些。FoxBASE+ 文件名由 1~8 个字母、数字或下划线组成,必须以字母开头,下划线只能嵌在文件名中间。扩展名由 1~3 个字符组成。文件名与扩展名中间都不得含有空格。FoxBASE+ 是通过扩展名来识别不同的文件类型的,因此对每种文件类型都默认一种扩展名,如果用户不加以指定,建立文件和操作时系统会自动加上或识别默认的扩展名。图 1-11 反映出 FoxBASE+ 的九种主要文件的关系。

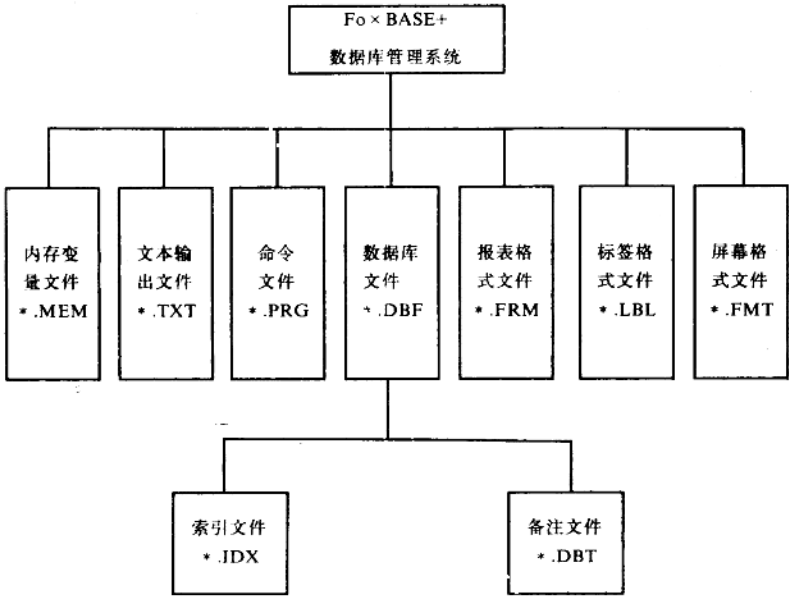


图 1-11 FoxBASE+ 文件组成

1. 数据库文件 (默认扩展名为.DBF)

一个数据库文件就是关系数据库中的一个关系,因此,它是 FoxBASE+ 中最核心的文件。数据库文件按记录存储数据,每个记录最多允许有 128 个字段,每个记录的总长度限制为 4000 个字符。一个 FoxBASE+ 数据库文件最多可存储 10 亿条记录。

2. 命令文件 (默认扩展名为.PRG)

命令文件是由用户编制的 FoxBASE+ 应用程序文件。命令文件是一种 ASCII 码文本文件,可以用 FoxBASE+ 的文本编辑命令进行编辑,也可以用任何一种文字处理软件 (如 wordstar) 进行编辑。FoxBASE+ 的命令文件分为程序、过程、用户自定义函数等几种形式。

3. 索引文件 (默认扩展名为.IDX)