第一章 概 述

在科学技术飞速发展的今天，电子计算机已成为极其重要的信息处理和通讯的工具，以各种应用系统的运行取代人们繁重复杂的手工操作。实践表明，生产效率和服务水平的提高，效益的增长，都与计算机的使用有直接的关系。计算机不仅仅用于数据计算，更多的是帮助人们进行决策分析。如何对大量的数据进行有效的组织和管理，这是每一位开发计算机应用系统的技术人员都应考虑的问题。

目前，用于开发应用系统的高级语言和数据库种类繁多，本书所介绍的 FoxBASE＋是目前最常用的数据库之一，它在国内已经广泛的推广。

本章将简要阐述数据库的分类和用途，以及 FoxBASE＋ 数据库的特点，随后再介绍 FoxBASE＋ 数据库使用的软硬件环境。

第一节 数据库的分类及用途

一、数据及数据管理

在介绍具体数据库之前，有必要先解释"数据"一词的含义。

按照国际标准化组织（International Standard Organization 简称 ISO）的定义，数据是对事实、概念或指令的一种特殊表达形式，它可以用人工或自动化装置进行通信、翻译或处理。因此，按照 ISO 的定义，数据不仅是指数字，还可以是文字（如大、小写英文字母和汉字）、图形、图形符、语音以及其它各种符号等。现代化的计算机是可以接收各种类型数据的，然而，把这些数据全部输入到计算机中，并不是每个用户的目的，按照原样取出也非期望的结果，归根到底，最终的目标就是要充分利用计算机对已输入的数据，按照一定要求进行统计、集中、独立的处理，再以某种形式将所需的数据提供出来。上述工作，可称之为"数据管理"。

二、数据库的分类

大量杂乱的数据输入到计算机里以后，如何才能更好地完成"数据管理"呢？70 年代初，诞生了一种数据管理方法——数据库管理系统（DataBase Management System，简称 DBMS），它强调数据是一种资源，要对这种资源加以组织，使之尽可能地为多个用户，即需要使用数据的人服务，使他们对数据具有一定的权利，从而实现数据的共享。

DBMS 采取的方法是建立若干个数据库。这就是一个具有通用性和综合性的数据的集合，是按照自然或常规的联系而构成的，就像一个仓库，保存着大量的数据，包含对各数据的描述及相互间的制约关系。

有了数据库以后，对它进行维护，同时接收和完成用户提出的各种访问数据的请求，这是数据库管理系统（DBMS）的职能。目前，国际上较为流行的数据库分类方法，是将数据库

分为二种类型：层次型数据库、网状型数据库和关系型数据库。

层次型数据库是一个定向的且具有一定顺序的树型结构，其特点是它有而且只有一个根结点和若干个从属结点。结点有时也叫片段，是构成数据库的基本单位。所有节点都位于一定的层次上，根节点与各从属结点之间由一条带箭头的连线联系起来。图 1-1 为一层次型数据库示意图，表示某大学的教学体系结构。学校为第一层，下设若干个系，如计算机系、自动化系、和外语系等，这些系属于第二层。每个系下设几个专业，这些专业属第三层…，所有这些均受该校领导的统一管理。因此，学校即为根节点，系、专业和年级等均是从属节点，它们相互之间的联接方式是学校对系、系对专业、专业对班。从图 1-1 可以看出，比例关系都是 1：N(N 为自然数)。

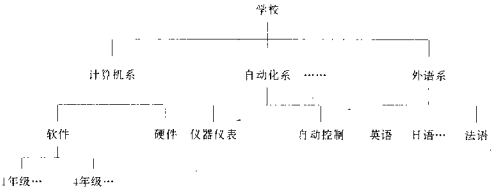

图 1-1 层次型数据库示意图

网状型数据库与层次型数据库相比，结构略复杂些，它不再是那种 1：N 的简单的树型结构，而是 M：N 的联结关系。图 1-2，有利于理解网状型数据库结构的特点。此图表示出学校内部教师、学生和所开设的课程之间相互关系，学校有多名教师，每位教师教多个班，每班有多名学生，每个学生需要学多门课程，同一门课可能由几个教师讲授，某位教师课后辅导多名学生，某个学生课后需要多位教师辅导，某些班级可能开设相同课程，某课程可能在不同班级开设。

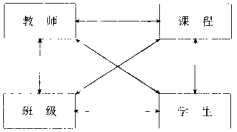

图 1-2 网状型数据库示意图

以上二种类型的数据库，共同存在一个问题，就是用户在处理数据时都要求对数据之间的联系非常清楚，而联系是根据应用的需要，一旦用户的应用需求发生变化，就要修改数据之间的结构关系，即数据库的结构，严重时可能要使整个应用系统受到影响。

鉴于存在上述的缺陷，70年代初提出了第三种类型的数据库——关系型数据库。

关系型数据库是借助逻辑运算和关系运算的理论建立起来的，以二维表的形式出现。二维表即是指有M行N列的一张表格，每行为一个元组，相当于一条记录（由多种数据组合而成）；每列称为一个字段，相当于表中各栏目标题。图1-3为一个通讯录，它共有6个栏目，即6个字段，名称分别是序号、姓名、单位、通讯地址、邮政编码和电话，每一个字段都有多个取值，这种类型的数据库简单灵活，存取方便，独立性强。

通 讯 录

序号	姓名	单 位	通讯地址	邮政编码	电话
1	李 雪	北京十四中学	宣武区牛街	100040	367231
2	强佳佳	世纪电脑公司	中关村北路14号	100082	2564367
3	林明玉	云南沧江机械厂	云南景洪西路40号	666100	24487
4	吴耀珀	上海电器科研所	上海武宁路505号	200063	2574990
5	王家卫	交通部信息中心	德外安翔北里	100101	2025549
6	马 希	华东工学院	南京孝陵卫200号	210014	443215

图1-3 关系型数据库示意图

三、数据库的应用

按照某种形式建立数据库并向其中输入数据以后，即可依据一定的条件，随时进行查询、检索、排序、修改或删除等处理。例如，在图1-3中，若某人的电话号码改变，则可通过编号找到此人的全部信息，以新的电话号码替换旧的电话号码，重新存入库中。今后若再次查询，显示出来的即是新的电话号码。若发现某人的信息重复了，则可以直接删除。

由此可见，数据库不仅有利于对大量数据进行统一的保存，根据需要做相应处理，而且能减少数据重复，在各种需要使用这些数据而且条件具备的地方，可随时共享使用。

因此数据库管理系统的出现，在计算机发展史上具有及其重要的意义。

第二节 FoxBASE⁺数据库的特点

一、FoxBASE⁺数据库的提出

随着数据库管理理论的提出，各种形式的管理系统相继推出。1984年7月，美国ASHTON－TATE公司开发了dBASE Ⅱ关系数据库管理系统。与先前的dBASE Ⅱ相比，性能指标有了较大的提高，功能也做了一定的扩充，用户可以方便地对数据库中的数据实现增加、删除、修改、统计、显示、打印等操作。但是，经过一段时间的运行和使用，人们发现，这一关系型数据库的内存量极其有限，每个库中能够存储的记录数也很少，数据读取速度慢，特别是只限于单机使用，不能实现较高层次上的数据库资源共享。此后的dBASE Ⅲ plus和dBASE Ⅳ，虽然实现了在局域网上的多个用户对数据的共享，但在多用户操作系统环境下，仍然不能实现数据共享的要求。

1986 年 6 月，美国 FOX 公司成功地开发出 FoxBASE+ 多用户关系数据库系统，不但能够在以 MS－DOS 作为操作系统环境的单机或局域网上正常运行，而且也可以在以 80386 为芯片的全 32 位微机上，以 XENIX System V 为操作系统的环境下使用。

二、FoxBASE⁺数据库的特点

FoxBASE⁺数据库是在 dBASE Ⅱ的基础上发展而来的，在源文件和数据文件一级与 dBASE＋完全兼容，支持标准 dBASE 的全部功能。这种兼容性可以把在 DOS 下运行的 dBASE Ⅱ 和 dBASE Ⅲ plus 应用程序与数据直接移到单用户的 FoxBASE⁺环境下运行。因此，凡学过 dBASE Ⅱ 的用户，只要稍加学习即可掌握和使用 FoxBASE⁺数据库，这就为 FoxBASE⁺关系数据库的推广创造了条件。

FoxBASE⁺发展到今天，已经有 1.0、2.0、2.1 几个版本，本书主要是针对 FoxBASE⁺ V2.1 版的命令、函数以及程序的编制进行系统地介绍。

与 dBASE Ⅲ plus 相比，FoxBASE⁺系统具有如下特点：

1. 运行速度快。测试表明，FoxBASE⁺的运行速度一般比 dBASE Ⅲ plus 快 5 倍以上。

2. FoxBASE⁺是用 C 语言开发的，易于移植和推广。FoxBASE⁺可运行于多种操作系统和多种硬件环境，多用户版本与单用户版本完全兼容，用户具有的软件随着硬件和系统软件的发展，很容易向高档机上移植。

3. 提供了多种运行方式，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边解释边执行，也可以编译后执行。

4. 在用户环境方面，FoxBASE⁺注重在语句功能和外部界面功能上的设计，这对于用户进行信息处理、数据计算和程序设计都有很大帮助。

三、FoxBASE⁺数据库的基本功能

FoxBASE⁺具备 dBASE Ⅲ plus 的全部功能，但这并不是说它原样照搬了 dBASE Ⅲ plus 的命令和函数，而是在 dBASE Ⅲ plus 的基础上又增加了一些新功能，同时，在某些方面也做了一些改进。现简述如下：

1. FoxBASE⁺允许的内存变量个数最多由 dBASE Ⅲ plus 的 256 个增加到 3600 个，允许同时打开的文件数由 dBASE Ⅲ plus 的 16 个增加到 48 个，在一个过程文件中允许包含的最多过程数由 dBASE Ⅲ plus 的 32 个增加到 128 个。

2. 可以通过 DIMENSION 语句建立一维或二维变量数组。值得注意的是，数组的大小受内存变量总数的限制，且数组中的数据可以传到数据库中作为记录存储、数据库中的记录也可快速传到数组中。总而言之，数组和数据库之间可以进行数据交换。

3. 具有屏幕信息的存储功能。即将当前屏幕信息存入缓冲区或内存变量，在需要时可以从缓冲区或内存变量中恢复所保存的屏幕信息。

4. 在运行中可自动地动态调整缓冲区、程序存储区及其他资源在内存中的位置，从而能使当前计算机所配置的内存资源得以充分使用。

5. 增加了菜单设置命令，包括建立免条菜单、激活菜单、接收对菜单选项的选择等。菜单的显示方式有下拉式和上弹式。

6. 增加了调程序缓冲区、将缓冲区数据嵌入磁盘文件、向磁盘缓冲区写字符和自动清

屏等功能。

7. 增加了一些新函数，诸如返回指定工作区中数据库的别名、字段数、工作区号函数，返回现行目录名称函数、光标隐/显函数等。

8. 通过使用关键字 ADDITIVE，可以使任何在先前就已存在的关系保持不变。

9. 在表达式中引进了两个新的操作符。

！：表示 NOT

＝＝：与"＝"类似，不同之处是它仅用于字符串比较，而且是精确比较，即串尾的空格会影响比较的结果。

四、FoxBASE⁺数据库的文件及主要技术指标

FoxBASE⁺使用的文件类型有：

1. *.DBF 数据文件
2. *.PRG 程序文件
3. *.FOX 程序文件的目标文件
4. *.IDX 索引文件
5. *.MEM 内存变量文件
6. *.FMT 屏幕格式文件
7. *.FRM 报表格式文件
8. *.LBL 标签文件
9. *.TXT 文本文件
10. *.DBT 备忘文件
11. *.BIN 汇编语言的二进制目标文件
12. *.BAK 数据文件的后备文件
13. *.TAK 备忘文件的后备文件

dBASE Ⅲ 的 *.PRG 文件、*.DBF 文件、*.MEM 文件、*.FMT 文件、*.FRM 文件和 *.LBL 文件不作任何修改即可在 FoxBASE⁺环境中直接使用，但是 FoxBASE⁺中的索引文件 *.IDX 与 dBASE 中的索引文件 *.NDX 格式有所不同，数据查询的速度加快，而索引文件体积减小，这是由于 FoxBASE⁺系统使用的是新的索引技术。如果在 FoxBASE⁺中打开 dBASE 的索引文件，则系统会自动根据数据文件重建索引.生成 .IDX 文件。另外，也可以在系统配置文件 CONFIG.FX 文件中用 INDEX＝NDX 将 FoxBASE⁺环境中默认的索引文件扩展名说明为.NDX，这样，再次引用 dBASE＋ 格式的索引文件时，就消除了 dBASE＋格式的索引文件。

最后，为便于读者更好地使用 FoxBASE⁺，现将 FoxBASE⁺的主要技术指标列出如下：

.每个数据文件最多存储记录数：	1000000000
.每条记录最多包含字段数：	128
.每条记录允许最多字节数：	4000
.每个字段最多允许字符数：	254
.每个字符串最多包含字符数：	254
.每个命令行最多包含字符数：	254

. 每个报表标题最多包含字符数：	254
. 每个索引关键字最多包含字符数：	100
. 数值运算的数字精度：	16 位
. 内存变量最多可定义：	3600 个
. 最多可建数组：	3600 个
. 每个数组中数组元素最多可定义：	3600 个
. 可同时打开最多文件数	48 个
. 可同时打开最多数据文件：	10 个
. 可同时打开最多索引文件：	21 个
. 每个数据文件最多同时打开索引文件：	7 个
. 每个过程文件中最多含有过程：	128 个

第三节　FoxBASE⁺数据库使用的软硬件环境

用任何一种高级语言编制的程序或使用任何一种形式的数据库，都要求在一定的环境下运行和使用。FoxBASE⁺数据库对于软、硬件环境也有一定的要求。

一、软件环境

FoxBASE⁺数据库可以在 DOS 和 XINEX 两种操作系统环境下运行，所以 FoxBASE 系统分 MS—DOS 版本和 UNIX 或 XENIX 版本。当使用 FoxBASE⁺系统时，一定要注意版本标签说明，即在什么环境下运行的，版本号是多少等等。版本越高，操作系统的版本也要相应的提高。

二、硬件环境

1. 为保证能够正常运行，计算机至少需有 1.5MB 的内存。多数情况下，它能在 1MB 内存的系统下运行，但为了得到最佳性能，应当为每位用户准备 0.4MB 的有效内存空间。当然，内存容量越大，运行速度就越快。

2. 最好有 80X87 系列的协处理器，以便使程序运行速度达到最优。在某些大型数据库系统中，甚至必须有协处理器。

3. 必须有一台由 TERMCAP 支持的终端。TERMCAP 是一个正文文件，它系统地描述了许多常用终端的功能特性。该文件中包含有 WY－60 终端能力，这些能力可支持画双线方框的绘图字符，若这些能力没有出现在系统 TERMCAP 文件中，那么在绘制图形时就只能使用单线字符。此外，这种 TERMCAP 文件使 FoxBASE⁺能使用终端的标准接口，并且终端的特定行是相对独立的。在必要的时候，用户可以在 TERMCAP 文件中增加对新终端的描述。

使用非 MS－DOS 和非 IBM 兼容计算机的用户需要注意以下两个问题：

1. 光标控制键

光标控制键的功能是在 TERMCAP 文件中说明的。因此，如果在使用的计算机上这几个控制键不能正确地执行其功能，就需要查看 TERMCAP 文件或者询问系统管理员，以了

解这些键是如何定义的。

2. 中断键

在 MS—DOS 版本下，FoxBASE⁺的中断字符是 ESC 键，而在 XENIX 环境下，中断字符是 DELETE 键。

习　题

1. 数据库分为哪几种类型？各有何特点并举例说明。
2. 简述 FoxBASE⁺关系数据库的特点。
3. 说明 FoxBASE⁺关系数据库对软硬件环境的要求。
4. FoxBASE⁺关系数据库有哪些文件类型？

第二章　数据库的基本操作

第一章已经介绍了数据库的分类、用途、FoxBASE⁺数据库的特点以及使用时对软硬件环境的一些基本要求。本章将阐述 FoxBASE⁺的使用，包括进入/退出 FoxBASE⁺系统的方法、数据库文件的建立、打开和关闭以及数据记录的追加、修改、删除等操作。

第一节　如何进入/退出 FoxBASE⁺系统

FoxBASE⁺数据库可以在两种不同的操作系统下运行，即 DOS 系统和多用户 XENIX 操作系统。然而不论是哪种环境，要运行 FoxBASE⁺的命令和程序，都必须在 FoxBASE⁺的特定环境下，运行结束之后也需要退出 FoxBASE⁺，才能执行操作系统的其他操作。因此，在一台已经安装了 FoxBASE⁺系统的计算机上，掌握进入/退出 FoxBASE⁺的方法是首要解决的问题。

首先将计算机打开，当屏幕上出现操作系统提示符(DOS 下是“>”，XENIX 下是“$”或“%”)后，从键盘敲入：

FOXPLUS 或 MFOXPLUS（其中 MFOXPLUS 是网络版本的启动命令）

然后按回车，即进入了 FoxBASE⁺的交互执行方式。屏幕上即出现 FoxBASE⁺的提示符——“.”(一个圆点)。在这种状态下，用户可以逐条输入 FoxBASE⁺命令，以便一条一条地执行相应的功能。

另外，还有一种方法，可以在进入 FoxBASE⁺系统的同时，运行一个 FoxBASE⁺程序。具体方法是在操作系统提示符下，键入：

FOXPLUS <程序文件名>

或 MFOXPLUS <程序文件名>

按这种方法执行后，系统不是停留在“.”提示符下，而是屏幕闪动一下就自动进入程序执行方式，并运行指定的程序文件。FoxBASE⁺数据库规定，不论是采取第二种方式运行程序还是在“.”提示符下运行程序，程序文件名均可不写扩展名，如果源文件(*.PRG)和编译后的目标文件(*.FOX)同时存在，则 FoxBASE⁺会自动选择执行目标文件。

当想做的工作已经完成，不需要再进行其他操作时，可以退出 FoxBASE⁺系统，操作方法是键入：

.QUIT

于是，经过短暂的间歇，就退回到操作系统，屏幕上重新出现相应的操作系统提示符。

利用上述方法，可以正常地进入和退出 FoxBASE⁺系统。切忌在运行 FoxBASE⁺程序的过程中强行关机退出，否则会破坏当前正在打开的数据库，造成数据丢失。

第二节　数据库文件的建立

一、数据库文件的建立

在数据库管理系统中，数据文件是最基本的，也是最重要的处理对象，几乎所有的工作都是围绕着它来进行的。由于组成每一个数据库文件的数据都有一定的数据结构，因此，要实现对数据库文件的操作，必须首先建立数据库文件的结构，数据结构可以在 FoxBASE⁺ 环境下建立。

在给出命令之前，须注意，FoxBASE⁺ 的所有命令和函数在书写时采用大写或小写都是允许的。建库的具体命令是：

.CREATE <数据库文件名>

文件名必须以字母开头，由不多于 8 个的字母、数字或下划线构成。若以单个字母作为文件名，则不能取 A～J 中的字母。各文件不能重名，且为了便于理解、记忆，在定义文件名时，用户最好取与库内容有联系、具有一定含义的名字，例如保存银行储户档案的数据库，可以定为 YHZH. DBF（YHZH 即银行帐户拼音的缩写），记录学生成绩的数据库，取名为 STUDRESU. DBF（学生成绩英文缩写）。

在 CREATE 命令中，扩展名可以省略，系统会自动赋予 .DBF。当按下回车键后，屏幕立即出现一个框，框中显示的是在当前状态下，可以使用的各种控制键及其相应的功能，就称此框为提示框 1（见图 2-1）。

MS—DOS 环境

CURSOR ← →	INSERT	DELETE	Up/Down Field: ↑ ↓
Char: ← →	Char: Ins	Char: Del	Cursor Menu: ^ Home
Word: Home End	Field: ^ N	Word: ^ Y	Exit/Save: ^ End
Pan: ^ ← ^ →	Help: F1	Field: ^ U	Abort: Esc

XENIX 环境

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: ^ Z ^ B	Help: ^	Field: ^ U	Abort: ^ Q

图 2-1　提示框 1

框下面分左右两栏，同时显示下面各项：

Field name　　Type　　Width　　Dec

在屏幕最底行显示出执行的命令、数据库所在的位置和名称、光标当前所在的字段号等信息。

任何一个数据库文件，都是由许多记录组成的，每个记录又包含若干个字段，这些字段分别定义为字段名称、类型、宽度和小数部分的宽度等。具体要求如下：

字段名（Field_name）也是以字母开头，由字母、数字或下划线组成的字符串，长度不超

过 10 个字符，字段名中的字母在输入时无论是小写或大写，一律以大写形式显示并保存。另外，字段名也可以是汉字串，但长度不能超过五个汉字。

字段类型(Type)可以从下面五种情况中选择一种：

C：字符型：用于存储字母、数字、符号和空格。

N：数值型：用于存储整型、实型数值。

L：逻辑型：用于存储逻辑值。逻辑值只有两种，逻辑真值(T 或 t)和逻辑假值(F 或 f)，也可以用 Y 或 y 表示逻辑真，N 或 n 表示逻辑假。

D：日期型：用于存储日期。日期型格式为 MM/DD/YY(月/日/年)。

M：备注型：又称记忆型或备忘型。可以输入任意长度的内容。

宽度(Width)指该字段可存储的数据的最大长度。若是字符型的，则指字符的最多个数，FoxBASE⁺系统规定最多不能超过 254 个；对于数值型的数据，正负号和小数点均各占一位，而小数部分的位数是在 Dec 中说明的；逻辑型数据和日期型数据的宽度、是由系统自动定义的，分别是 1 位和 8 位；备注型的字段，虽然其内容可是任意长度，但系统仍然对其有限定，存储字符最多不超过 4096 个，而且系统为每个数据库文件的备注字段自动建立一个和数据库文件同名但扩展名为.DBT 的备注文件。在每条记录的备注字段中，均包含一个由系统建立的，长度为 10 的指针，指向备注文件中相应记录的该字段内容。

例 2.1　建立一个银行帐户档案，假定包含下列内容：

帐号：	8	C
密码：	10	C
存取日期：		D
存取类型：	1	C
数额：	8.2	N
备注：		M

数据库名为 YHZH.DBF，于是按下列步骤操作：

.CREATE YHZH

当屏幕显示出提示框 1 后，光标就停留在第一个字段处，从键盘敲入“帐号”并回车，光标就跳到类型定义(Type)，系统直接给出默认类型为 CHARACTER 的字符。因帐号就是字符型的，故直接按回车，光标即跳到宽度定义项(Width)，键入 8 并回车，至此，第一个字段“帐号”定义完毕，光标即跳到下一个字段名定义项等待输入。按上述操作方法，依次将其余几个字段“密码”、“存取日期”、“存取类型”、“数额”和“备注”的定义内容通过键盘录入。当最后一项“备注”的类型定义完成后，光标跳到下一个空行，此时再按回车键，屏幕底部的注释行上，就出现一行信息：

Press ENTER to confirm, Any other key to resume

这时，按 Enter 键，则表明用户对已输入的信息给予确认，即全部字段定义都是满意的；按其他键，则表示用户对输入的信息还不肯定，光标会再次返回最后一个字段的名称定义处，可以利用提示框 1 中的各种控制键进行修改。再次确认无误后，在最底行显示出：

Input date record now? (y/n)

这时询问用户是否立即向库中输入记录，此处最好敲入“N”、回车后，即退出建库操作状态，重新回到 FoxBASE⁺的提示符“.”状态下。

DOS 中有一个显示当前目录下存在的文件的命令 DIR，在 FoxBASE⁺中也有类似功能的命令，其格式为：

.DIR [路径] [通配符] [TO PRINT]

在 FoxBASE⁺提示符"."下，能够显示当前目录下存在的数据库和每一个库的记录数、最后修改时间以及总字节数，最后一行显示的是总文件数和盘上余留的字节数。如果指定一个通配符，那么只显示该类型的文件。TO PRINT 选项用于将显示结果送到打印机打印。

对于前面已经建立的数据库 YHZH.DBF，如果按照下列格式使用 DIR 命令：

.DIR YHZH *

就会发现，不仅显示出 YHZH.DBF，同时还有一个 YHZH.DBT 文件，这是因为.DBF 中含有备注字段，系统对此自动建立一个备注文件，到目前为止，这二个文件虽然已经建立，但均无数据。

例 2.2　在仓库管理中，通常要求定期查点库存并与帐本上的记录进行比较，这称作盘点。为了完成盘点，建立一个盘点的工作库 GPD.DBF，以便记录每次盘点的结果。信息包括：

BM(零件编码)：	7C	每种零件的代码
SJCL(实际存量)：	3N	仓库中某零件的实际拥有量
ZMCL(帐面存量)：	3N	按帐本记录零件应有的数量
YKSL(盈亏数量)：	3N	实际存量与帐面存量的差值
KCJ(库存价)：	6.2N	零件的单价
YKJE(盈亏金额)	8.2N	单价与盈亏数量的乘积

这样一个数据库，请读者试用 CREATE 命令建立，然后将结果与下面的内容进行比较。

Field name	Type	Width	Dec
BM	C	7	
SJCL	N	3	0
ZMCL	N	3	0
YKSL	N	3	0
KCJ	N	6	2
YKJE	N	8	2

如果库已存在或有重名，当使用 CREATE 命令时，会在屏幕上出现下列提示信息：

GPD.DBF always exits, overwrite it(y/n)?

意思是 GPD.DBF 已经存在，询问是否要覆盖原有内容。如果不需要保存 GPD.DBF 以前的内容，即相当于要重新建立一个库名叫 GPD.DBF 的数据库，那么就敲"Y"键，否则敲"N"，原库不受任何影响，屏幕重新退到提示符"."状态。

二、数据库文件结构的显示

对于已经建立的数据库，若想显示它的库结构，再次检查是否正确，可以使用下面的命令：

.LIST STRUCTURE

或 .DISPLAY STRUCTURE

执行这二条命令，都能将结构完整显示，且提供下列信息：

1）数据库名(带绝对路径)

2）当前记录数

3）最后修改日期

4）记录的总长度(各字段宽度总和再加1)

使用 LIST 命令来显示 GPD.DBF 的结构，其操作步骤和显示结果如下：

```
.LIST STRUCTURE
STRUCTURE FOR DATABASE: C:\FOX\GPD.DBF
NUMBER OF DATA RECORDS: 0
DATA OF LAST UPDATE:07/10/93
FIELD     FIELD NAME     TYPE     WIDTH     DEC
    1     BM             C            7
    2     SJCL           N            3       0
    3     ZMCL           N            3       0
    4     YKSL           N            3       0
    5     KCJ            N            6       2
    6     YKJE           N            8       2
** TOTAL **                          31
```

可以用 DISPLAY 命令，再次显示 GPD.DBF 的库结构，验证一下结果是否相同。

实际上这两种方法只有一微小的区别，主要体现在对一个结构较为复杂的数据库，用 LIST 命令时，不论有多少字段，都是一次连续将全部内容显示完毕，而用 DISPLAY 命令，当字段数≥16 时，则采用分屏显示，满一屏后在屏幕最底行显示：

PRESS ANY KEY……

等待用户按了任意键才显示下一屏，这样有利于用户将一个比较复杂的数据库的结构看得更清楚。

三、数据库文件结构的修改

当一个数据库的结构已经建成之后，若发现其内容不太合理，例如类型不妥或宽度不够，就有必要重新定义。FoxBASE⁺为此提供了如下的命令：

.MODIFY STRUCTURE

这是一个对数据库结构进行全屏幕编辑的命令，可以完成字段的增加、修改和删除。按下回车键，会出现与建库时基本相同的格式，所不同的是，此时屏幕中间显示出所有定义过的字段，屏幕底行显示出数据库名，光标所在的字段、总字段个数，同时光标停留在第一个字段，等待用户利用屏幕上部提示框中的内容进行各种操作。

下面将其中各控制键及其功能按 MS—DOS 和 XENIX 二种环境对照说明(见图 2-2)。

在进行全屏幕编辑时，要注意当前正处于"插入"状态还是"替换"状态。可以看屏幕底部是否显示"INS"，如果有，就表示处于插入方式，否则处于替换状态。

如果想在某一个字段前面增加一个新的字段，须将光标移到该字段的首字符处，再按 CTRL—N，则立刻出现一个空行，可输入新的字段信息。

MS-DOS	XENIX	功　能
← →	^S ^L	光标向左、向右移动一个字符
Home End	^A ^F	光标向左、向右移动一个词，且光标停在词首
^← ^→	^Z ^B	左右半边屏幕切换
Ins	^V	插入/替换字符的方式切换
^N	^N	进入/退出字段插入状态
F1	^\	提示框消除/显示切换
Del	^G	删除光标所在处的字符
^Y	^Y	删除光标所在处的一个词
^U	^U	删除光标所在字段
↑ ↓	^E ^X	光标上移、下移一行
^Home	^]	显示/消除屏幕顶行的选择菜单
^End Esc	^W ^Q	保存/放弃修改的内容并退出编辑

图 2-2　控制键说明 1

用“HOME”切换的选择菜单内容包括：

BOTTOM　　TOP　　FIELD#　　SAVE　　ABORT

可以通过光标的移动选择其中的一项，按回车键后，执行某种操作。

BOTTOM：光标由当前字段移到最后一个字段。

TOP：光标由当前字段移到第一个字段。

FIELD#：光标移到指定的字段。选择此项后，还需输入要移到的字段号。

SAVE：将修改的内容保存。

ABORT：修改的内容不保存。

仍然以 GPD. DBF 为例。考虑到单价有时会达到千元以上，导致总金额 8 位宽度不够，需扩充为 11 位，所以原库中“KCJ”和“YKJE”二个字段的 WIDTH 要改为 8 和 11。利用 MODIFY STRUCTURE，将 GPD. DBF 的结构列出，移动光标（↓或 END）到“KCJ”的“WIDTH”处，将 6 改为 8 并回车（一定是处于替换状态下），再将“YKJE”的 8 改为 11 并回车，按“^End”，即将修改内容保存，并退到“.”状态。

再次用 LIST STRUCTURE 显示 GPD. DBF，结果正是修改后的。

用 DIR 命令执行 GPD. *，就会发现，显示结果不仅包含一个. DBF 文件，而且还有一个. BAK 文件。这个 GPD. BAK 文件是 GPD. DBF 的备份。每当对某一数据库修改时，FoxBASE⁺自动建立一个与之同名，但扩展名为. BAK 的备份文件，同时将. DBF 中的内容复制到. BAK 中。如果. DBF 中含有备注字段，则同时还会生成一个与之同名，扩展名为. TBK 的文件，存储原文件的备注字段的内容。

修改数据库的结构是经常会遇到的，但是当库中已有数据记录时，修改就会有下面二个问题：

1. 若想保留原数据，则一次只能对字段名或字段宽度中的一项内容进行修改。例如，

首先修改字段名，存盘退出后，再修改字段宽度。如果同时修改这二项，则数据将全部丢失。

2. 若修改某字段的类型，则全部数据记录都不再保留。所以，一旦数据库中已有若干条记录，那么对库结构进行修改时，需要特别慎重。

四、数据库结构的复制

当一个数据库文件已经存在，为完成某项工作，又需要用到该库的内容时，为防止意外发生，保持原库不受毁坏，就要将其复制，根据需要，生成一个结构完全或部分相同，但名字不同的新库，以便于进行各种处理。命令格式如下：

COPY STRUCTURE TO <新数据库名> [FIELDS <字段名表>]

需要复制的数据库为当前库，新数据库为目的库，FIELDS 为可选项，缺省则表示复制整个库结构，否则只将字段名表中列出的字段复制到新库中。字段名表可以是一个或多个字段，如果数量大于 1，则每个字段名中间用“,”隔开。

运用这条命令，只是将处于活动状态的数据库的结构加以复制，其中的内容（数据记录）并不复制。

若将 YHZH. DBF 的“帐号”和“数额”二项内容复制，则执行下列命令：

. COPY STRUCTURE TO YHZHBF FIELDS 帐号，数额

于是，一个新的数据库 YHZHBF. DBF（银行帐户备份）就建成了，并且含有二个字段：帐号和数额。

根据 YHZH. DBF 建立 ZH_CODE. DBF，使其含有“帐号”字段，然后再修改结构，增加“姓名”字段，按下列步骤进行即可实现：

. COPY STRUCTURE TO ZH_CODE FIELDS 帐号

. MODI STRU ZH_CODE

结果用 LIST 显示如下：

Field name	Type	Width	Dec
帐号	C	8	
姓名	C	8	
* * TOTAL * *		17	

注意：在 FoxBASE⁺ 中，所有命令中出现的英文单词，均可以采用简写方式，即只取前四个字符，这样做，系统是完全认可的。因此在书写分屏显示的命令 DISPLAY 时，可以写成 DISP，而 MODIFY STRUCTURE 与 MODI STRU 也是等价的。另外，所有选项，若是写在[ ]中，则在执行命令时可有可无，若是以< >形式出现，则执行命令时，必须写出，只是内容不定。

第三节　数据库的打开与关闭

一、数据库的打开

通常，一个数据库管理系统所涉及的数据库都不只一个，而是多个。DBF 文件。但

FoxBASE 数据库系统对所包含的数据库的状态是有一定控制的，具体体现在任意时刻只允许一个数据库文件是处于活动状态，其他的库只能是静止等待，而且只有活动的库才能进行各种操作。那么什么是活动的呢？如何做才能使所需要的数据库处于活动状态呢？打开数据库是唯一的办法。

由于 FoxBASE⁺ 是用 C 语言编写的，C 语言的基本操作对象是文件。所以，打开数据库文件，是一个比较复杂的文件操作过程。对于用户来说，没必要对打开过程的每个细节有详细的了解，只要知道打开就是指将指定的文件由磁盘调入内存工作区的操作。如果指定的文件在磁盘上根本不存在，那么，系统将给出提示信息，此打开过程也就失败了。如果需要打开的文件不在当前的目录下，那么该文件可以通过绝对路径或相对路径的形式写出。在 FoxBASE⁺ 中，路径的符号可以是"\"或"/"。

最简单的打开文件的命令是：

. USE <文件名>

文件名只能写一个，而且不必带扩展名. DBF。现在要将 YHZHBF. DBF 增加一个字段"存取类型"，且宽度是一个字符，即 1C，于是可在". "下，按下列顺序完成操作：

. USE YHZHBF

. MODIFY STRUCTURE

待显示出结构以后，利用前面讲过的方法，将光标移到最后一个字段并按"^N"，然后将字段"存取类型"插在"帐号"后面，再按"^W"存盘退出，在". "下执行

. LIST STRUCTURE

即将新的库结构显示如下：

1	帐号	C	8
2	存取类型	C	1
3	数额	N	11.2

如果当前没有活动的数据库，那么在执行 MODIFY STRUCTURE 命令时，屏幕将提示：

NO DATABASE IN USE. ENTER FILE NAME:

输入库名以后，才能进入到库中进行修改。

因为 FoxBASE⁺ 要求在任意时刻，只能有一个数据库是活动的，一旦对此库的操作完成，或暂时不再操作，要转向其他库进行处理，就要改变它的状态，从活动变为静止，只有这样，才能激活其他的库。

二、数据库的关闭

激活一个库是利用打开库实现的，使它再恢复到静止则可通过关闭实现。关闭的命令是 USE 后不带任何文件名。请看下面的操作过程：

```
. USE GPD
. DISP STRU
STRUCTURE OF DATABASE: GPD. DBF
NUMBER OF DATA RECORDS: 0
DATE OF LAST UPDATE: 07/20/93
```

FIELD	FIELD NAME	TYPE	WIDTH	DEC
1	BM	CHARACTER	7	
2	SJCL	NUMBER	3	0
3	ZMCL	NUMBER	3	0
4	YKSL	NUMBER	3	0
5	KCJ	NUMBER	8	2
6	YKJE	NUMBER	11	2
* * TOTAL * *			36	

```
. USE
```

为显示 GPD. DBF 的库结构，首先就要用 USE GPD 命令将其激活，DISP STRU 则是将当前的库结构分屏显示，因不需要再做其他工作，故立即使用 USE 命令关闭。因此时已经没有活动的数据库，若再执行 DISPLAY STRUCTURE，则屏幕会提示：

NO DATABASE IN USE.

假如要再次显示，就必须重复执行上述过程。这样做似乎有些麻烦，但这对于初学者来说，是一个很好的习惯。

当对某一个库的操作全部完成，或者暂时要转向其他库，也可以不使用 USE 命令将其关闭，而直接打开新的库，那么前一个库在新库打开的同时就自动关闭了。这样做既快速方便，又能达到要求。

继续上面的处理，当显示 GPD. DBF 以后，如果想将 GPD. DBF 复制，则可将最后一句 USE 命令变换如下：

```
. COPY STRU TO GPDC FIELDS BM, YKJE
. USE GPDC
. DISP STRU
STRUCTURE OF DATABASE: GPDC. DBF
NUMBER OF DATA RECORDS: 0
DATE OF LAST UPDATE: 07/20/93
```

FIELD	FIELD NAME	TYPE	WIDTH	DEC
1	BM	CHARACTER	7	
2	YKJE	NUMBER	11	2
* * TOTAL * *			19	

```
. USE
```

此时，GPD. DBF 和 GPDC. DBF 均被关闭，表面上虽然只是少写一句 USE 命令，但如果是一个比较复杂的程序，把这些没必要的命令语句都去掉，则会节省大量时间，对提高效率十分有利。

有一点需要说明，如果全部的库操作都已完成，则立刻敲入 USE 命令，这样做不但是一种好习惯，更重要的是在多用户系统环境下给其他用户使用数据库带来方便，这在第八章中将详细介绍。

前面讲到的打开和关闭数据库的操作，都是在 FoxBASE⁺状态下，也就是在". "提示符下进行的，一旦执行了 QUIT 命令，则所有的库操作命令就失去功效，而且所有打开的数据

库将自动关闭。因此，退出 FoxBASE⁺ 系统，也是关闭数据库的一种方法。

如果前面打开的数据库比较多，记不清当前的数据库是哪一个时，可以利用下面的函数进行跟踪反馈：

DBF()

此函数可以返回当前数据库的文件名。

例如：.USE YHZH

.? DBF()

显示结果：/USR/FOX/YHZH.DBF

.USE

因为刚刚打开的是 YHZH.DBF，所以执行函数后，立即显示出上述结果。

注意，所有 FoxBASE⁺ 的函数在使用时，都要在函数前写出"?"。

第四节　数据库记录的添加与插入

数据库结构的建立，这只是数据库操作的第一步，是打基础的过程，而建立数据库的更主要的目的是利用它们存储数据。于是，向各个库中添加记录，就成为用户要做的第二项工作。

在第二节介绍用 CREATE 命令建立数据库时，曾经提到过，每当一个库结构建好后，屏幕下面就会出现一询问信息：

DO YOU WANT TO ADD RECORD(Y/N)?

如果键入"N"，即退出创建状态；键入"Y"，则记录号立刻变为 1，于是可以输入一条记录，并且数据直接存入库中。若想连续输入，则记录号随之逐渐递增，直到退出为止。这是最简单，也是最原始的数据输入方法，并且只能在建立数据库结构时才能使用。

下面将介绍几种输入记录的命令：APPEND、INSERT 和 BROWSE。

一、数据记录的追加

命令格式共有二种，具体如下。

格式一：APPEND [BLANK]

格式二：APPEND FROM <数据库名> [FIELDS <字段名表>]
[FOR/WHILE <表达式>][TYPE <文件类型>]

第一种格式中，如果不加选择项 BLANK，则表示在当前数据库尾，即所有数据记录的后面追加一条新记录，若库中已有 N 条记录，则该记录即为第 N+1 条；如果写出选择项，那么就在库尾追加一条空记录，总记录数仍然加 1。

例如　.USE YHZH

.APPEND

屏幕上边即出现一提示框，与提示框 1 并不一样，称之为提示框 2(见图 2-3)。