关系型数据库管理系统 dBASE III 使用方法及其程序设计

张喜英 王秉湖 编
李大友 审

中国计算机技术服务公司北京分公司
培 训 中 心
一九八五年六月 北京

目　录

关系型数据库管理系统
dBASEⅢ使用方法及其程序设计

第一章 概 述

第一节 关系型数据库简介

1—1 信息、数据和数据库的产生

人类活动的历史总是离不开对反映客观世界的各种信息和数据的收集积累、加工处理、保存、传播和利用。那么，什么是信息呢？概括地说，信息是人们用以对客观世界直接进行描述的、可以在人们之间进行传递的一些知识。而数据则是信息的具体表现形式。数据本身是一些各种各样的物理符号及它们的组合，它反映了信息本身的内容。

我们对一个客观世界中的事物进行描述时，首先要弄清这个被描述的对象的各种属性。换句话说，对象本身就是它所具有的属性的集合。因此不存在无属性的客观实体。例如，对于一本具体的书，总有它的书名，作者、文字类型、字数、页数、内容摘要、价格、出版者及出版日期……等等。当我们认识了这些属性，也就认识了这个客观实体。若把这些属性表示出来，也就描述了这个客观实体，形成了概念世界中的实体模型。因此，概念世界中的实体模型是由反映这个实体属性的各种信息构成的。

进一步说，我们还可以按一定形式把这种描述数据化。例如，可以把上述各种属性的具体值写下来（或者输送到计算机中）。其中书名、作者等类的属性分别用一些文字来表示；字数、页数和价格等类的属性可由一些数字量来表示。所有这些值都是数据。它们的总和构成了这本书的数据模型。通过这样的模型，人们可以间接地了解到所反映的客观实体的各种属性，从而了解实体的本身。

由此可见，信息和数据有一定的区别。可以说前者是观念性的，后者是物理性的。信息本身可以直接反映客观事物的某些概念；而数据则是人们用来表现和传递信息的一种物理形式。另外，信息本身与载荷信息的物理设备无关；而数据形式则要随着这个物理设备的改变而改变。例如，上述关于一本书的属性，某些数字型的信息能够以十进制数形式写在纸上，这是它的一种数据形式。而在计算机中，它们则要以二进制数（一系列的“0”和“1”）的形式存放到贮设备中去。这是它的另外一种数据形式。两种不同的数据形式，但所表示的信息内容是不变的。最后应指出，不是所有信息均能变成数据。例如人们之间的某种暗示，表示了一定的信息，但是当前还无法把这种信息数据化。

当然，在许多地方信息和数据是难于区分的。信息本身有时已是数据化了的，数据本身就是一种信息。因此，在许多场合下不对它们进行区分。信息处理和数据处理往往指的是同一概念；例如与计算机之间交换数据，也可说是交换信息，等等。

有了数据随之就产生了数据处理的问题。例如，我们收集到的数据，往往需要经过组织加工才能真正地直接反映客观事物属性。然后再按一定的形式将它加以保存或传递，以便于

随时可能对其进行利用（如查阅，问询等）或进行修改，增删等。这些都是数据处理的内容。在计算机出现后，尤其是被较广泛地利用以后，大量的数据处理任务便自然地由它来承担。因为计算机（尤其是现代的计算机）可以高速度地进行各种算术、逻辑运算和数据传送操作，它具有大容量的存贮器（尤其是大容量直接访问的磁盘存贮器），可以存放大量的数据，可以随时地访问。目前，几乎人类积累的和正在收集的发展中的各种信息，均可以用计算机所要求的形式存放到计算机中去。因而，目前我们所说的数据处理系统，一般总是指计算机数据处理系统。

有了计算机这样一种雄厚的物质基础，数据处理系统便得到了很快的发展。它主要经史了三个发展阶段：第一阶段是人工管理系统的发展。这时期的计算机主要用于科学计算，实际上无专门的软件系统对数据进行管理。计算机本身只相当于一个计算工具。一个程序面向一种数据。进行程序设计时，往往也要对数据的结构，存贮方式，输入输出方式等进行设计。

第二阶段是文件系统的发展。这时期计算机不仅用于科学计算，也用于管理工作中。尤其在磁盘一类的大容量外部存贮设备使用以后，产生了专门管理数据的软件——文件系统（一般包括在操作系统中）。这时期的数据能够以文件形式长期保存在计算机外存贮器里，可以随时对数据进行查询、修改、增删等处理。文件系统也提供了逻辑文件到物理文件的转换方法，使程序和数据有了一定的独立性，不必再全部由编程序者考虑数据的物理结构，从而实现了以文件为单位的数据共享。但是，所有这些进展都是局限在一定范围内的。例如，文件本身基本上还是对应于一个或有限几个应用程序的，用户数据逻辑结构不能完全地独立于存贮数据的物理结构。文件中的最小单位是记录，但用户不能直接以记录为单位进行访问，而只能访问整个文件，等等。这些问题的解决都有待于软件的进一步发展。

基于上述的情况，六十年代末出现了数据库管理系统。初期的数库管理系统正是为解决文件系统的缺点而设计的。这就是数据处理系统的第三个发展阶段。

1—2 数据库系统的基本概念

现在讨论什么是数据库系统的问题。为此，让我们首先了解后一下数据库中的数据结构。

在观念世界中的实体可以分为对象和属性二个方面。我们称对对象进行描述的数据为记录，称对名属性进行描述的数据为数据项。记录本身就是系列数据项的组合。每个记录均有型和值的问题。就总体上说，记录或数据项均有一定的类型。例如，上面对一本书的描

表1.1 图书目录

书号	书名	作者	页数	单价	销售情况
05328	计算机原理	李坚	238	2.85	有货
05856	软件基础	史明伟	325	3.60	无货
05631	数字电路原理	朱光英	192	2.10	有货
⋮	⋮	⋮	⋮	⋮	⋮

述，书的页数便是一个数据项的类型。而对个体来说，具体的一本书的页数是一个定值。它就是这一数据项的值。一个记录是它包括的所有数据项值的一定组合。也就是说，各类据项值的集合便构成了记录的值。记录的类型则相应地由它所包含数据项的类型组合而成。因而，记录类型可表现为一个框架。例如，我们用表1.1对某书店的图书进行描述。由表中可见，每个记录由六个数据项组成。它们在不同的记录中取不同的数据值。各列首行代表了对应数据项的类型。而记录的类型则由各列首行的总和来表示。即它具有下面那样的框架：

书　号	书　名	作　者	页　数	单　价	销售情况

记录的类型与各记录值的总和就构成了表1.1这样的数据文件。

在文件中不允许有完全相同的重复记录。通常在文件中总有某个数据项或某些项的结合能够用来标识任一记录。如上例中的书号这一项，对每个记录来说，该项的值均是唯一的。因此，也称这一类的数据为关键字（Key）。在数据库中常常用这种关键字来分类或索引。这可使用户对文件中记录的访问很方便。

由此可见，数据库是动态存贮的有组织的关联数据的集合。它们有这样一些特点：数据本身是结构化了的，可为多用户的应用服务；它可以独立于使用它们的程序；对数据的插入、增删、修改、检索等均可以由公用的方法来完成。有时，也把数据库仅仅理解为上述的数据集合。而数据库管理系统，则指的是对其进行管理的软件系统。它包括数据描述语言及其翻译程序，数据操纵语言及其编译程序，数据库管理例行程序等。

1—3　关系数据库模型

数据库的设计方法很多，较为流行的是层次方法、网络方法和关系方法。dB A SE是属于后一种的数据库管理系统，因此，下面介绍这种方法的若干特点。

关系方法是发展较晚的一种数据库方法。它的数据模型称为关系模型。在这种模型中，数据均以二维表的形式出现。每个二维表称为一个关系。例如，表1.1便是一个关系。或者，可用表1.2来表示这种二维表的一般形式。

表1.2　关系的一般形式

R

A_1	A_2	… …	A_i	… …	A_n
V_{11}	V_{21}	……	V_{i1}	……	V_{n}^{1}
V_{12}	V_{22}	……	V_{i2}	……	V_{n2}
⋮	⋮		⋮		⋮
⋮	⋮		⋮		⋮
V_{1m}	V_{2m}	……	V_{im}	……	V_{nm}

每个关系均有一个名称，称为关系名。例如表1.1，我们可命名它为图书关系（或B OO KS）；表1.2可命名为R关系，等等。表中的一个列相当于上面说的一个数据项，代表一个属性。在数据库中也称它为一个字段(FIELD)。名列的上端标出的是该字段的字段名，有时也称为属性名（如A_1、A_2……）。它表示的是字段的型。而下面的各栏中写入的是具体的数据值，也称分量。横向的一行为一个元组，相当于一个记录。第一行是各字段型的集合，构成一个框架，此即为记录的型。其他的行则是各个记录值。

关系方法的主要特点就表现在它的数据描述的统一性。即，描述的对象及对象间的联系等均只能用关系来表示。而关系本身必须是规范化的。它的每个列均是单纯的字段，或者说不允许表中再有表。

关系方法有其严格的数学基础。它对数据的各种处理主要以集合代数为根据。有关这些理论问题可参阅文献〔1〕和〔2〕。

下面我们给一个简单的例子说明关系型数据库的构成及它能给用户提供的主要功能。表1.3—1.5中的三个关系可简单地用来表示某个单位有关职工情形管理的数据库：职工关系表示了每个职工本身的简单情形；工资关系表示了每个职工的工资情形；科室关系表示每个科室的概要情况。有了这样一个数据库，属于以下种类的用户问询是可以得到回答的：

表1.3职工关系

职工号	姓　名	年令	性别	科　室	职　务
0025	李小平	28	女	技术科	工　人
0038	张　伟	40	男	第一车间	工程师
⋮	⋮	⋮	⋮	⋮	⋮

表1.4工资关系

职工号	月工资	月奖金
0634	105.00	12.00
1052	89.50	10.50
⋮	⋮	⋮

表1.5科室关系

科室名	负责人名	负责人职称
技术科	杨　宁	工程师
生产科	刘建新	高级工程师
⋮	⋮	⋮

①各个对象的情形：例如某职工的年令、性别，等等。

②有关对象间的联系：如某职工的月工资是多少，某科室的负责人是谁，等等。

③指出具有某种属性的对象：如某科室中女职工都是谁，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形：例如某职工所在科室的负责人姓名、年令、职称，等等。

⑤所有某类对象的统计情形：如各类人员的平均工资是多少，工资低于（或高于）某数额的人数是多少，全单位的月工资（或月奖金）总额是多少，等等。

所有这些问题均可通过上述三个关系之一，或通过其中几个或全部，直接或间接地得到答复。当然，做为数据库管理系统，不仅要提供用户以上问询的回答，还要提供许多有关的其他操作。例如，用户建立自己数据库的方法，随时增删、修改，编辑已建数据库的手段输出各种问询答案、报表等的方式等等。所有这些均将在后面对dBASE Ⅲ的具体介绍中逐一地说明。

1—4 数据库管理系统面向用户的主要技术指标

一个数据库管理系统对用户来说提供了许多操作命令以对数据进行处理。同时，受软件运行环境及软件本身等许多因素的限制，也必定对用户的数据本身及对其进行的操作有一定的限制。这些就体现了系统的性能和指标。对用户来说，它们主要表现在下述一些方面：

1.有关数据库文件的指标：

由于文件是由记录构成的，因而每个数据库文件所能容纳的最多记录数和最多允许它占用的存贮空间量（一般用字节数表示）便是它的主要指标。

2.有关记录的指标：

记录是由字段构成的，因而对记录来说下述两个参数是很重要的：一个记录的最大长度（字节数）和每个记录允许包含的最多字段数。

3.有关字段的指标：

字段的有关指标也表现在两个方面。一是允许字段有几种类型；一是每个字段的最大宽度（字段数）。

4.有关文件操作的指标：

有多少种与数据库有关的文件种类，允许同时打开的数据库文件数，对每个数据库文件可同时打开的其他辅助文件（如索引文件、格式文件等）数，等等。这些都代表了系统对文件操作的主要性能。

5.有关数值型数据精度的指标：

系统能处理的最大数和最小正数等。

6.有关内存变量的指标：

一般为方便用户的数据处理，系统均允许定义一定数量的内存变量。在这方面的重要指标是内存变量的种类，最多允许的内存变量数目，允许占用的内存空间数，等等。

这些指标仅仅是在用户使用数据库管理系统时，对自己的数据所必须受到的约束。至于系统提供的数据处理功能的多少和方便灵活程度，以及数据的冗余度、安全性，提供的语言系统的完备性和用户性能等等，则是不完全能用数量表示出来的另一方面的性能和指标。

第二节 关系型数据库管理系统dBASEⅢ简介

dBASEⅢ是美国Ashton Tate公司在原先的类似系统dBASEⅡ的基础上发展出来的适用于IBM-PC/XT及其全兼容的16位微型计算机(如在我国流行的长城0520微型计算机)的关系型数据库管理系统。它是1984年刚推出来的。dBASE-Ⅱ是一个适用于多种8位微型计算机系统的关系型数据库管理系统，被称为“大众数据库”，在教育，科技、会计、商业、财政以及各种企业管理方面均得到了较广泛的应用。近几年来也已在我国流行，受到了好评。但是，随着数据库管理系统本身的发展，随着用户数据量的不断增加，应用领域不断扩展，尤某是计算机系统性能的迅速发展（主要是速度、容量的提高），有必要发展功能更为强大的数据库管理系统来适应之。dBASEⅢ便是在这种情形下产生的新一代系统。据美国市场的估计，dBASEⅢ会非常快地成为最为热门的微型计算机上使用的数据库管理系统。在我国，该系统处于刚开始引进的阶段，予计必定会得到迅速的推广应用。汉字化的问题也在解决之中，有的已达到可使用阶段。

2—1 dBASEⅢ的运行环境

1.硬件环境：

- IBM个人计算机：

  IBM PC—XT

  与IBM PC百分之百兼容的微型计算机系统（如长城0520）

- 内存最小容量——256k字节
- 至少两个360k软磁盘驱动器或一个360k软磁盘驱动器加一个硬磁盘驱动器。
- 任何行宽80列以上的打印机。

2.软件环境

MS—DOS或PC—DOS2.0以上版本的操作系统。

2—2 dBASEⅢ与dBASEⅡ的比较

虽然dBASEⅡ是世界上使用较广泛的在微型计算机系统中应用的数据库管理系统，但与dBASEⅢ相比，无论是性能指标上还是使用方便性上。都要逊色的多。

dBASEⅢ不仅仅是dBASEⅡ的一个改型。从下面几方面的比较可以看出它是更一代的系统：

1.技术指标的比较

在dBASEⅢ中，每个数据库文件最多可以有1千兆（(10^9)个记录。只要硬件设备满足条件就可以达到这个极限。每个记录最多可以有4000个字节。这比dABSEⅡ的每个数据库文件最多可以有65535条记录，每条记录最多可以有1000个字节的限制参数要大得多。

dBASEⅡ中规定，一个数据库中最多允许有32个字段。而dBASEⅢ中允许的字段数为此数的4倍。

在dBASEⅡ中可以同时开辟两个工作区，同时打开两个数据库文件，可完成这两个数据库间的数据传送。而在aBASEⅢ中，增加到十个工作区，最多允许同时打开十个数据库文件，并可在它们间进行数据传送。

1BASEⅡ的字段类型有三种，字符型(Character)、数字型(Numeric)和逻辑型(Logical)。dBASEⅢ中则又增加了日期型(Date)和明细型(Memo)。尤某是后者，是许多先进数据库系统中所具有的。它事实上可以使记录的长度大大加大，而且降低数据库文件的冗余度。

对于内存变量，与dBASEⅡ相比，dBASEⅢ将内存变量数从14个提高到256个。从内存变量的类型来说，增加了日期型内存变量。同时，在程序应用中又把内存变量分成了全局型和局部型两种。

2.dBASEⅢ主要功能的改善与提高

从上面的技术指标对比来看，dBASEⅢ明显地远远优于dBASEⅡ。而从下面几方面来看，可以进一步看到dBASEⅢ的功能是远远地比dBASEⅡ强大。

①丰富的命令和函数

dBASEⅢ中除了保留了几乎全部dBASEⅡ的操作命令外，还增加了一些常规命令和SET命令（设置系统参数和状态的命令）。

新增常规命令中主要有HELP和ASSIST命令。前者相当于一本有关dBASE Ⅲ 各种功能、命令、函数及使用方法的说明书，对那些还不大熟悉的用户来说，是非常方便的。后者则是一组组菜单方式的程序。可通过人机对话建立和使用用户的数据库，在这两个命令的支持下，用户不必对应用方法有较多的了解，便可使用这个系统。其他新增的常规命令和SET命令可分别见表1.6和1.7。它们的具体功能后面将逐一介绍。

dBASE Ⅲ还新增加了13个函数，列在表1.8中。它们的作用解释在下一章中。

当然，也有些dBASE Ⅱ中的个别命令在dBASE Ⅲ中功能有所变化，或者被更好的操作所取代。

表1.6 dBASE新增命令表

命　　令	功　能　简　介
ASSIST	用菜单方式帮助用户使用系统
AVERAGE	求平均值
CLOSE	关闭某类文件
COPY FILE	复制文件
HELP	用菜单方式对命令及其应用等进行说明
MODIFY LABEL (或 CREATE LABEL)	建立和编辑标签格式文件
LABEL FORM	输出标签
PRIVATE	规定内存变量为部分型的
PUBLIC	规定内存变量为总体型的
RUN	执行DOS命令或目的程序
SEEK	查找记录
TYPE	输出ASCII文件
ZAP	清除数据库中的所有记录

表1.7 dBASE Ⅲ新增SET命

命　　令	功　能　简　介
SET	用菜单方式设置系统各种参数
SET COLOR TO	设置屏幕颜色特性
SET DECIMALS TO	设置小数位数

SET DELIMITER	设置数据边界符
SET FILTER TO	列表时“过滤”掉某些记录
SET FIXED	决定是否所有数据显示时均有固定小数位数
SET FUNCTION	定义各功能键
SET HEADING	决定列表时是否列出字段名
SET HELp	决定输入错误时是否问询“Do you want some help ?”
SET MENUS	决定某些命令的全屏幕编辑方式F 是否上部附有简要控制键功能菜单。
SET PATH	决定文件查寻路径。
SET PROCEDURE	打开一个过程文件。
SET SAFETY	决定文件重写时是否发出问询。
SET UNIZUE	用以建立重复项的顺序清单。

表1.8 dBASEⅢ新增函数表

形　式	内　容
BOF ( )	文件开端
EXP ( )	自然指数
LOG ( )	自然对数
ROUND( )	对小数进行四舍五入
SQRT( )	平方根
CTOD( )	字符型数据—日期型数据转换
DTOC( )	日期型数据—字符型数据转换
MOUNTH( )	以数字形式表示月份（1—12）
DAY( )	月内日期数（1—31）
YEAR( )	年份（1900—1999）
DOW( )	日期（星期数，1—7）
CMOUNTH( )	月份名（英文名）
CDOW	日期（星期数英文名）

②改善了报表功能

提供的“CREATE REPORT”或“MODIFY REPORT”命令是一个全屏幕编辑方式命令，用户可很方便地建立报表格式文件，对它的修改也很容易。这与dBASE Ⅱ相比，灵活了很多。

③改善了屏幕输出格式

允许用内部字处理程序或辅助程序dFORMAT来生成专用的屏幕格式文件。用户也可以自己规定输入格式。

④可以运行外部程序

当系统内存大于256K时，可以执行操作系统外部命令或调用外部程序。这个功能可由RUN(或!)命令来实现。执行这些程序后，系统仍返回dBASE Ⅲ中调用该程序前的状态。

⑤整个系统更易于学习掌握，也更易于应用

上面已简要说明了ASSIST和HELP命令。这对初学者说是两个非常有用的工具。另外，在许多全屏幕编辑命令中，均提供简单菜单以说明主要控制键的功能。这也使初学者感到很方便。

⑥增加了过程文件

在dBASE Ⅲ中，还有一种称之为过程文件的新的命令文件。一个过程文件最多可以包含32个程序。系统允许将过程文件中的所有程序的名字都读入一张表之中。此后，当每次使用DO命令时，系统就搜索这张表。这样，使系统勿需每迂DO命令就从磁盘上打开文件，而只需从过程文件中取出执行程序，从而节省了时间(从软盘上打开一个文件约需1.5秒钟)。这种方法也可大量地减少磁盘目录空间。从以上几个方面可以看到dBASE Ⅲ的确要比dBASE Ⅱ优越很多，从而可估计到它将比dBASE Ⅱ获得更为广泛的应用。

2—3 怎样运行和退出dBASE Ⅲ

1.运行dBASE Ⅲ的准备工作

初次使用dBASE Ⅲ时，应当检查一下dBASE Ⅲ的系统盘上是否有一个名为CONFIG·SYS的文件。检查的方法是先将操作系统盘插入A驱动器中，然后用热启动（或冷启动）启动系统。当屏幕上出现操作系统的提示符后，取出操作系统盘，插入dBASE Ⅲ系统盘，然后输入操作系统命令：

A>DIR *.SYS↵（“↵”表示回车”）

如果看到有CONFIG.SYS文件，那么，再用下面的命令检查它是否有内容：

A>TYPE CONFIG.SYS↵

如果显示出如下内容：

FILES = 20

BUFFERS = 24

则说明CONFIG.SYS文件已建立好了。

如果只有文件名没有内容，可以用操作系统的文本编辑程序将

FILES = 20

BUFFERS = 24

加到CONFIG.SYS文件之中。

如果dBASE Ⅲ系统盘上没有CONFIG.SYS文件，则可用文本编辑程序或直接从键盘上键入下列内容以生成该文件：

A> COPY CON: CONFIG.SYS↵

FILES = 20↵

BUFFERS = 24 ^Z↵

^ Z表示按下CTRL键后手不要放开然后再按下Z键。生成文件的这种方法在操作系统手册有关COPY命令的章节中有详细叙途。

IBM—PC通常允许用户在同一个时间内最多打开3个文件，而它自己却要使用其中的5个。dBASE Ⅲ允许你在同一个时间内最多打开10个数据库文件或15个其它类型的文件。因此，PC所规定的文件数目是不够用的。FILES = 20这条命令正是为解决这一问题而设置的。它把可打开的文件数提高到20，除去PC本身使用的5个之外，有15个留给用户。

设置缓存区是使dBASE Ⅲ在内存中存放更多信息的一种技术。这种技术可以加快dBASE Ⅲ的运行速度，但却要以占据较多的内存为代价。在CONFIG.SYS文件中将缓存区设置为24个，如果内存不够用，可以减少缓存区的数目，最少要保持12个。

当dBASE Ⅲ系统盘上确实已有了一个名为CONFIG.SYS的文件后，应该用操作系统的COPY命令将这个文件复制到操作系统盘上，或者可以反过来做：先在操作系统盘上建立CONFIG.SYS文件，然后将其拷贝到dBASE Ⅲ系统盘上。无论采用哪种方法，都应确保用来引导dBASE Ⅲ的操作系统盘及dBASE Ⅲ盘上都有CONFIG.SYS文件。因为每当PC启动时，它都要检查该文件，并根据该文件的内容来设置PC工作时用到的有关文件个数及缓存区个数的参数。

只要在操作系统盘和dBASE Ⅲ盘上建立了CONFIG.SYS文件，便可一劳永逸，尔后每当你用这两块盘引导dBASE Ⅲ时，都不必再做上述工作。

2.运行dBASE Ⅲ

在A驱动器中装入dBASE Ⅲ系统盘，然后在操作系统的提示符下入键命令：

A>dBASE↵

dBASE Ⅲ先给出的响应信息："echo off"，然后显示出近一帧屏幕的文字信息。这些文字叙述的是有关版权所有的说明。最后在屏幕的底部给出dBASE Ⅲ的提示符

.Ⅲ

在圆点提示符的右边有一个闪耀的光标，此时你便可在光标闪耀处输入命令了。

直接在圆点提示符下输入命令的方式称之为命令方式；与之相对应的还有程序方式。在命令方式下，无论何时输入命令都可以得到立即响应。

如果dBASE Ⅲ系统盘是初次使用，建议你再建立一个CONFIG.DB文件。你可以在操作系统的提示符下键入如下命令来生成：

A> COPY CON: CONFIG.DB↵

SET DEFAULT TO B: ^Z↵

CONFIG.DB文件比CONFIG.SYS文件只多一条命令，它是用来设置用户文件所存放的默认盘的。设置了"SET DEFAULT TOB:"命令之后，在dBASE Ⅲ的所有操作中，如果文件名前不加盘符，系统都将在B盘上进行文件存取。当然，你也可以使用"SET DEFAULT TO C:"命令将默认盘定为C驱动器。

CONFIG.DB文件还可以包含屏幕颜色设置，系统状态设置等命令。在运行 d BASE Ⅱ时，CONFIG.DB文件中由SET命令设置的状态还可以用同一条命令的另一种状态来改变。如果不加改变，那么，在dBASE Ⅱ运行的始终都将保持CONFIG.DB文件中的SET命 令所设置的状态。

在操作系统盘上没有必要建立CONFIG.DB文件，

3.退出dBASE Ⅱ

在dBASE Ⅱ的圆点提示符下键入命令

.QUIT↵

便可退出dBASE Ⅱ，回到操作系统的提示符下。

参考：

[1]萨师煊、王珊："数据库系统概论"高等教育出版社出版（1983年）

[2]郑若忠、王鸿武："数据库原理与方法"湖南科技出版社出版（1983年）

第二章　dBASEⅢ的基本语法和规定

当我们要使用dBASEⅢ这个关系型数据库管理系统建立、维护和使用自己的数据库时，首先必须了解它的一些基本语法和规定。这些问题牵扯到数据库的基本结构、数据的类型和范围、可使用的各种函数、对数据库进行各种操作的命令以及与数据库有关的各种计算机文件的建立和使用等等基本概念。

第一节　记录、字段和数据值

前面已讲过，关系型数据库所处理的数据被表示为一个二维的表，称之谓关系。而关系中所包含的是一个个记录。这些记录的共同性是包含有相同的字段。例如，我们这里给出这样一个关系，称它为BOOKS（图书）。在计算机中，它以一个文件的形式被存放，名称为BOOKS.DBF。它的内容如下：

Record #	TITLE	PRICE	PUBLC_DATE	SELL_UP	SUMMARY
1	The Thirty Years	5.65	05/23/81	.T.	Memo
2	The Smail House	1.85	11/13/82	.F.	Memo
3	Red and Black	5.10	01/05/81	.F.	Memo
4	The Fishman and The Fish	0.46	08/26/82	.T.	Nemo
5	The Middle-aged Person	2.78	12/13/83	.F.	Memo

这里，“Record #”意为记录号；“TITLE”是“书名”；“pRICE”是“价格”；“PUB-DATE”表示“出版日期”；“SELL—Up”表示“是否已卖完；“SUMMARY”则是内容摘要。显然这样一个二维表便可表示一个书店的图书情况。这里，共有五个记录，表示五种图书。每个记录均有上述五种内容，即有五个字段。

dBASEⅢ允许一个数据库文件（表征一个关系）中最多可有10^9个记录，可有128个字段。每个记录中各个字段的内容总和最多可占4000个字节。各记录总计允许使用的最多字节数为2×10^9个。但是，允许有一种明细(Memo)型字段。以上面例子中的“SUMM ARY”字段便是。它可使记录所占空间得以加大。理论上说，如果每个字段均为明细型的，每个记录所占空间便可扩大到521K个字节（1K＝1024）。事实上，最始的限制因素将是计算机系统的磁盘容量。关于明细型字段，下面还将具体说明之。

上述例子中的记录号是按各记录输入计算机的顺序编排的。

在关系型数据库中，字段便是被处理的最基本的变量。它以字段名表示之。例如上面的“TITLE”等五个英文字便是。在dBASEⅢ中规定，字段名变量是由不多于10个的字母、数字组成，其第一个必须是字母。不允许有空格字符号夹在其中，但允许有下划线“—”。

当定义一个字段名时，使用大写字母还是小写字母均是可以的，系统显示它时，均以大写字母取代小写字母。

例如，“A5”、“XYZ”、“abcde”、“N125”、“NO—3”“LAST—NAME”等等，均是合法的字段名。但是，下述几个是不允许的“A*B”、“5X”、“X Y Z”

“Power—Output”，等等。

对每个记录来说，各字段变量取一个具体的值。它是数据库中的常量。我们称它为数据值，或简称数据。它才是用户用来反映所处理的具体事物的具体属性的数据。

在dBASEⅢ中，字段共有五种类型：字符型（C型）、数字型（N型）、日期型（D型）、逻辑型（L型）和明细型（M型）。各种字段均有一定的宽度规定，即系统允许它的数据值最多为多少个字符所表示。这个最大值称为它的字段宽度。下面我们分别对这五种字段类型进行说明：

1.字符型（C型）字段

字符型字段的数据值是由各种可印刷的字符构成的，即它是一个字符串常量。它中可包括字母、数字、专门符号及空格等等。它们以其ASCII码的形式存放于字段之中。其最大允许宽度为254个字符。例如上面的例子中，“TITLE”便是一个字符型字段变量。对应的五个数据值均是字符串，例如“The Small House”。通常，在记录以外引用它们时，均需以单括号“'”、双括号“〃”、或方括号“[“和”]”将它们括起来。

2.数字型（N型）字段

这种字段中的数据值是一个可进行算术运算的数。它又可分为整数型和小数型两种。在dBASEⅢ中允许数字型字段最大宽度为19（包括小数点位）。当为整数型时，可以是最多19位的正整数，也可以是最多18位的负整数。若为小数型的数，允许小数部分（不包括小数点）最多为15位，或者最多为字段宽度减2（留下前导0和小数点位）。例如，定义该字段宽度为10时，小数位宽度不能超过8。另外，整数部分宽度最多也不允许超过16位。否则，16位以后的位将不准确。

上述数据库例子中的价格(PRICE)字段即为一数字型的，它有两位小数位。因此，是小数型的数。

3.逻辑型（L型）字段

逻辑型字段的数据值仅有两个值可取：T(True—)真和F(False—伪)。在输入时也可用Y(Yes—是）和N（Not—非)来表示。因此，这种字段的宽度是固定为1的，用户不能选择。在上面的例子中“SELL—UP”字段便属此种类型。系统列表时均以“.T.”和“.F.表示。在该例中，T表示该种图书已经售完，而F表示尚亦售完。

4.日期型（D型）字段

这种字段用以存放表示日期的数据。它的一般形式是：“月／日／年”。其中，月、日和年均为两位数。例如“03／25／85”表示1985年3月25日。系统允许用户用其他格式表示日期。但这种字段也是宽度固定的，为8。在dBASEⅢ中，允许一个日期型数据和一个数据相加减，从而求出另一个日期来。也允许两个日期型数据相减，得到一个数（表示两个日期间的天数）。

上述例子中“PUBLC—DATE”字段便是一个日期型的。

5.明细型（M型）字段

这种字段也是宽度固定的，为10。但这个数并非对数据值所含字符数的规定。事实上，这个宽度中存放的仅是一个指针，它指向另外一个磁盘文件。在这个文件中，每个明细型字段的数据值均可用一段文字来表示。这段文字是按512个字节为一数据块来存放的。允许每个数据值最多占8个数据块，即4K字节。通常系统对数据库列表时，仅在该字段中列出“Me-

mo”，表明它是明细型字段。系统提供专门的手段对这种字段进行输入、编辑及输出列表等操作。

例如在上例中，“SUMMARY”便是一明细型字段。在这里，我们可以对每个记录（对应种图书）编写一段文字，以表示该书的内容摘要。

在这里还应说明一下的是各字段数据值在文件中的存放形式问题。对字符型字段来说，若其数据值所含字符数少于字段宽度时，则在数据值右边填以空格补齐；对数字型字段来说，当数据值宽度不到字段宽度时，则在其前面（左面）补以空格。因此，在列表输出时，字符型字段均自左边对齐。其字段名也采取这种方法。而数字型字段则相反，采取的是右边对齐的方法。对于其他字段，无此问题。它们在文件中宽度一定。这三种字段的数据值列表时，也是采取左边对齐的原则（此处是相对其字段名而言，文件中并无空格保留）。其中明细型字段仅显示“Memo”（自右边第一字符位置开始）。这些均可从上面对数据库BOOKS的列表输出中看出来。

第二节 内存变量（Memory Variables）

系统处理的另一种变量是内存变量。它是存在内存之中，而不在数据库文件中。它取的值也是一种常量，也称它为数据值。它的设置主要是给用户使用数据库系统提供更多的方便。用户可以随意定义自己的内存变量，应用于进一步的计算和数据处理之中，或用来表示对数据库进行某种分析处理后得到的数据结果。

对内存变量的规定在许多方面是与对字段变量的规定相同的。例如，对变量名的规定便是如此，也只允许为以字母开头，最多包含10个字母或数字的一串字符，其中也不允许包含有下划线“—”以外的符号。当定义一个内存变量时会发现，必须遵循上述的规定。例如，变量名中若含有11个合法字符时，仅前10个为有效，最后一个被舍掉；若含有多于11个的字符，定义将无效。

内存变量共可分为四种类型：字符型（C型）、数字型（N型）、日期型（D型）和逻辑型（L型）。与字段变量相比，无明细型。对它们的宽度规定也与对应的字段变量全同，不再具体赘述。

dBASEⅢ允许最多定义256个内存变量同时存于内存之中。但系统对它们还有一个限制：总共占用的字节数不能超过6000个。

一般情形下，在命令方式或程序方式下，均可直接用内存变量名对其进行访问。但是有时某一内存变量名可能与所使用的数据库中的字段变量名相同。为了防止混淆，在访问内存变量时，可以用下面的格式来输入内存变量：

M—>内存变量名

内存变量也可被做成文件而存贮到磁盘中，也可将这种文件再度调入内存中来。

在程序方式下，内存变量还可分为全局(PUBLIC)）局部(PRIVATE)两种。第五章中将再详细说明这一问题。

为了下面讲解的方便，这里我们先介绍一下内存变量的定义方法及读出其数值的方法。为此先介绍一个命令：STORE。我们可用它直接从键盘上定义内存变量。例如，我们欲定义一个内存变量“NAME”，它是字符型变量，赋与它一个值“Liu—Ming”（刘明）。则在命令方式下可输入下述命令：