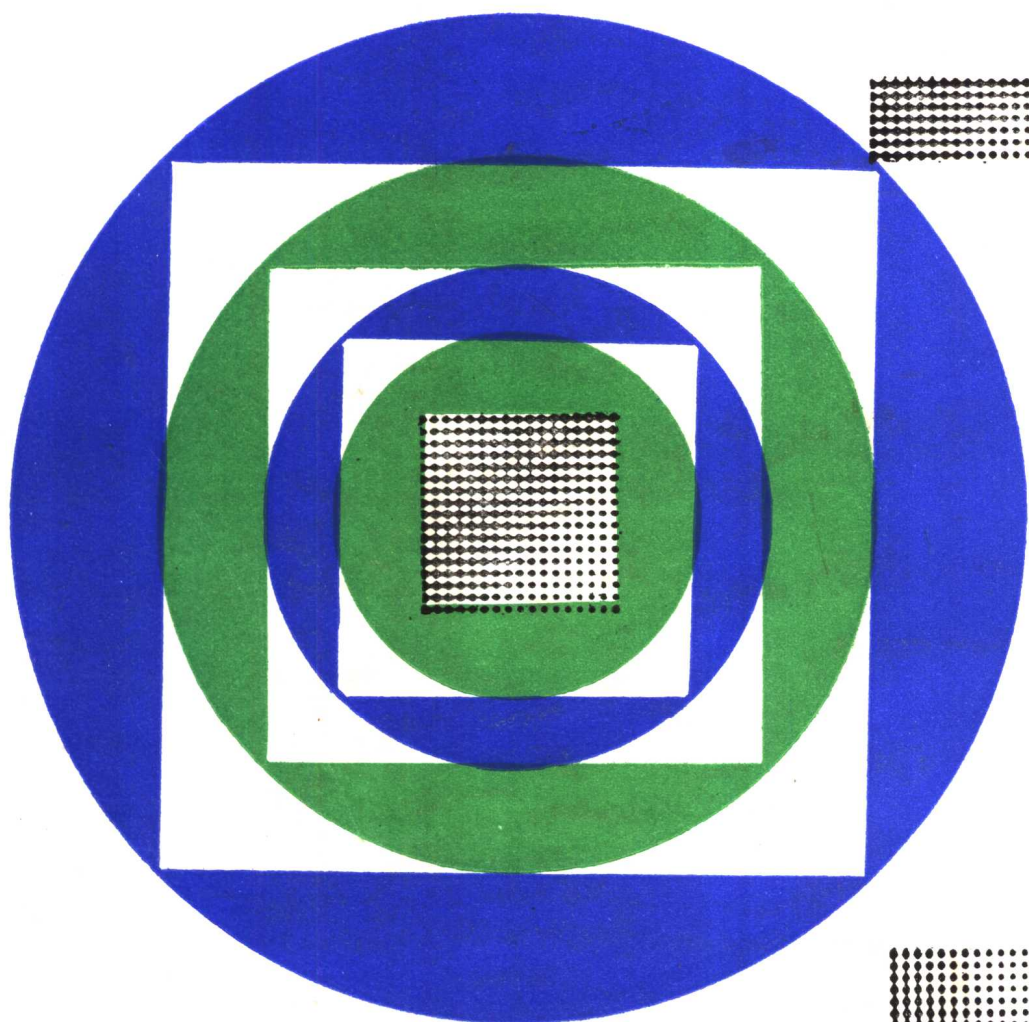


高等学校试用教材

微型机数据库管理系统

——汉字dBASE IV

董长德 崔巍



高等教育出版社

微型机数据库管理系统

—— 汉字 dBASE IV

董长德 崔巍

高等教育出版社

(京)112号

内 容 提 要

本书主要介绍汉字 dBASE IV 的使用方法和程序设计技术。dBASE IV 除对 dBASE III 的基本命令进行了扩充以外,还增加了许多新的功能,如菜单和窗口、应用程序生成系统、SQL 语言、网络 dBASE IV 与数据库安全性等。

本书内容全面,讲解清楚,举例丰富。可作为信息管理、财政金融类专业以及其他非计算机专业的教材或参考书,也可作为各种培训班的教材使用。

高等学校试用教材

微型机数据库管理系统

——汉字 dBASE IV

董长德 崔 巍

高等教育出版社出版

新华书店北京发行所发行

高等教育出版社激光照排技术部照排

高等教育出版社印刷厂印装

开本 787×1092 1/16 印张 25.5 字数 630 000

1992年4月第1版 1992年4月第1次印刷

印数 0 001—5 720

ISBN 7-04-003761-0/TP·101

定价: 6.60 元

前 言

随着微型计算机技术的迅速发展以及微型计算机的普及、在微型机上开发的数据库管理系统 dBASE 的使用也越来越广泛，并在实际工作中取得了良好的效果。目前 dBASE 系统软件已在我国拥有众多的用户。

dBASE 是美国 ASHTON-TATE 公司进入 80 年代以来研制的关系数据库管理系统。已相继推出了 dBASE II、dBASE III 以及 dBASE III PLUS。该系统最大的特点是硬件运行环境要求低，功能齐全，操作方便，简单易学。在世界上有“大众数据库”之称。

dBASE IV 是 1988 年 10 月推出的新一代微型机数据库管理系统。该系统是在 dBASE III PLUS 的基础上增加了许多命令和函数：增加了结构化查询语言 (SQL)；提供了 dBASE IV 和其它数据库系统之间有效地连接；增加了联网的功能。所有这些特点使 dBASE IV 的功能更接近于大型数据库管理系统。

随着计算机的推广应用，目前很多高等院校非计算机专业都把计算机管理列入教学计划中。其中微型机数据库 dBASE IV 已作为主要列选教材之一。本书就是在作者编著 dBASE III 的基础上，按着 dBASE IV 的特点及内容，结合教学上的要求新编著的一本通用教材。可作为高等院校非计算机专业的教材以及工程技术人员的自学参考书。全书共有十八章，其中第二、三、四、五、六、七、八、九、十、十一、十二章及附录由董长德同志编写；一、十三、十四、十五、十六、十七、十八章是由崔巍同志编写。

在编写过程中，承蒙刘惠芳先生的认真审核并提出了许多宝贵的意见，在此表示衷心感谢。由于编者的实际经验及水平所限，书中会有不当之处，希望广大读者批评指正。

编 者

1991 年 2 月

目 录

第一章 概述	(1)	2.4.2 常用参数设置	(49)
1.1 数据库基础	(1)	第三章 数据库文件	(55)
1.1.1 数据库的发展和基本概念	(1)	3.1 数据库文件的结构	(55)
1.1.2 三种范围与信息结构	(4)	3.1.1 数据库文件结构的定义	(55)
1.1.3 三大数据模型及描述数据的方法	(5)	3.1.2 建立数据库文件的命令	(57)
1.1.4 关系模型的几个概念和术语	(7)	3.1.3 应用实例	(59)
1.2 dBASE IV 简介	(9)	3.2 数据库文件 (.DBF) 的调用及关闭	(66)
1.2.1 dBASE IV 的新特点	(9)	3.2.1 调用数据库文件	(66)
1.2.2 dBASE IV 的运行环境	(16)	3.2.2 数据库文件的操作	(67)
1.2.3 dBASE IV 的装配	(17)	3.2.3 关闭数据库文件	(67)
1.2.4 dBASE IV 的技术指标	(19)	第四章 数据库基本操作	(68)
1.2.5 dBASE IV 的操作	(22)	4.1 数据库记录的定位	(68)
1.3 功能键与宏	(23)	4.1.1 GO/GOTO	(68)
1.3.1 功能键	(23)	4.1.2 SKIP	(68)
1.3.2 宏 (Macros)	(24)	4.2 数据库数据的输入	(69)
1.4 数据库目录 (CATALOG)	(25)	4.2.1 添加记录	(69)
1.4.1 数据库目录的概念	(25)	4.2.2 插入记录	(71)
1.4.2 数据库目录的使用	(25)	4.3 数据库编辑修改	(71)
1.5 控制中心简介	(26)	4.3.1 修改	(71)
1.5.1 控制中心的屏幕格式及主菜单	(26)	4.3.2 文件及记录的删除	(74)
1.5.2 控制中心工作台面菜单	(29)	4.4 数据库的显示	(76)
1.6 符号及有关约定	(31)	4.4.1 显示数据库记录	(76)
第二章 数据库系统常量、变量、表		4.4.2 显示数据库结构	(77)
达式、常用函数及参数设置	(32)	4.4.3 显示内存变量信息	(77)
2.1 常量	(32)	4.4.4 显示历史缓冲区中的命令	(78)
2.1.1 数字常量	(32)	4.4.5 显示文件	(79)
2.1.2 字符常量	(32)	4.4.6 显示数据库系统环境参数	(79)
2.1.3 逻辑常量	(33)	4.5 浏览命令	(80)
2.2 变量	(33)	4.6 数据库复制	(82)
2.2.1 内存变量与内存变量数组	(33)	4.6.1 文件的复制	(82)
2.2.2 系统内存变量	(37)	4.6.2 数据的复制	(84)
2.3 表达式	(37)	4.6.3 数据库结构复制	(85)
2.3.1 算术运算表达式	(38)	4.7 数据库其它操作	(86)
2.3.2 关系运算表达式	(39)	4.7.1 进入控制中心命令	(86)
2.3.3 逻辑运算表达式	(40)	4.7.2 显示磁盘文件目录命令	(86)
2.3.4 字符串运算表达式	(41)	4.7.3 系统求助命令	(87)
2.4 常用函数及参数设置	(42)	4.7.4 清除命令	(87)
2.4.1 常用函数	(42)	4.7.5 文件改名命令	(88)

4.7.6 文本文件显示	(88)	9.1.2 三角运算函数	(118)
4.7.7 文件关闭命令	(88)	9.1.3 字符串运算函数	(121)
第五章 数据库重新组织	(90)	9.1.4 日期运算函数	(127)
5.1 分类排序	(90)	9.1.5 数据类型转换函数	(131)
5.2 索引排序	(91)	9.1.6 状态测试函数	(134)
5.2.1 单索引文件	(91)	9.1.7 参数测试函数	(138)
5.2.2 多索引文件	(92)	9.1.8 文件标识测试函数	(142)
5.2.3 索引文件、多索引文件的调用及 关闭	(93)	9.1.9 数据库文件程序函数	(145)
5.3 重新索引排序	(95)	9.1.10 金融专用函数	(150)
第六章 数据库检索	(96)	9.1.11 菜单函数	(151)
6.1 顺序检索	(96)	9.1.12 键盘操作函数	(153)
6.2 索引检索	(97)	9.2 自定义函数	(155)
6.2.1 FIND 命令	(97)	第十章 参数设置命令	(158)
6.2.2 SEEK 命令	(98)	10.1 SET 命令	(158)
6.3 建立查询设计屏幕	(99)	10.2 系统参数设置命令	(159)
第七章 数据库统计	(100)	10.2.1 调试程序参数设置命令	(159)
7.1 数值统计	(100)	10.2.2 数字运算参数设置命令	(160)
7.1.1 求和	(100)	10.2.3 文件操作参数设置命令	(161)
7.1.2 求平均值	(100)	10.2.4 时间参数设置命令	(166)
7.1.3 计数	(101)	10.2.5 数据库的辅助操作参数设置命令	(167)
7.2 财务统计	(101)	10.2.6 环境参数设置命令	(185)
7.3 分类统计	(103)	10.2.7 其它有关参数设置命令	(190)
第八章 多重数据库的操作	(104)	第十一章 数据库系统内存变量	(192)
8.1 数据库之间数据传输	(104)	11.1 系统变量的概念和作用	(192)
8.1.1 数据库数据的输出	(104)	11.2 系统内存变量的分类	(192)
8.1.2 数据库数据的输入	(104)	11.2.1 控制数据流的系统内存变量	(192)
8.2 数据库工作区	(105)	11.2.2 控制打印机特性的系统内存变量	(196)
8.2.1 工作区的概念	(105)	11.2.3 控制打印作业特性的系统内存变量	(199)
8.2.2 工作区的选择	(105)	11.2.4 控制文本打印特性的系统内存变量	(202)
8.2.3 数据库之间命令的操作	(106)	第十二章 数据库程序设计	(205)
8.3 数据库文件间的更新	(107)	12.1 交互式输入/输出命令	(205)
8.4 数据库文件间的关联	(109)	12.1.1 交互式输入命令	(205)
8.4.1 选择〈关键字表达式〉	(109)	12.1.2 交互式输出命令	(207)
8.4.2 选择 RECNO ()	(110)	12.2 格式输入输出命令	(209)
8.4.3 选择〈数字表达式〉	(110)	12.2.1 @ 命令	(209)
8.4.4 多个相匹配记录的操作	(111)	12.2.2 格式输出命令	(210)
8.5 数据库文件的连接	(112)	12.2.3 格式输入命令	(211)
第九章 函数	(114)	12.2.4 数据描述符	(216)
9.1 基本函数	(114)	12.3 程序设计	(224)
9.1.1 算术运算函数	(114)	12.3.1 程序结构	(225)
		12.3.2 程序文件与过程文件	(235)

12.3.3 全局变量及局部变量	(238)	15.3.1 简介	(282)
12.3.4 程序设计的有关命令	(240)	15.3.2 转换dBASE IV 数据文件到图形文 件	(282)
12.4 事务	(241)	15.4 二进制程序模块的加载和运行	(284)
12.5 打印控制命令PRINTJOB/ ENDPRINTJOB	(244)	15.4.1 二进制程序模块的加载	(284)
12.6 程序文件的建立与运行	(247)	15.4.2 二进制程序模块的运行	(285)
12.6.1 程序文件的建立与修改	(247)	15.4.3 二进制程序模块的显示和清除	(285)
12.6.2 程序文件的调试与编辑	(247)	15.5 在 dBASE IV 中执行DOS 命令或 程序	(286)
12.6.3 程序文件的运行	(250)	第十六章 应用程序生成器	(287)
12.6.4 错误处理	(251)	16.1 应用程序生成器的工作环境	(287)
第十三章 菜单和窗口	(255)	16.1.1 应用的定义	(287)
13.1 菜单的设计和使用	(255)	16.1.2 功能键	(288)
13.1.1 菜单的定义	(255)	16.1.3 工作菜单	(288)
13.1.2 菜单的激活	(257)	16.2 应用程序生成器使用实例——开 发一个数据库应用系统	(295)
13.1.3 菜单的选择	(259)	16.2.1 应用概述	(295)
13.1.4 和菜单有关的其它几条命令及函数	(261)	16.2.2 设计过程	(296)
13.2 窗口功能	(263)	第十七章 SQL	(302)
13.2.1 窗口的定义	(263)	17.1 简介	(302)
13.2.2 窗口的激活和使用	(263)	17.2 dBASE IV 的SQL 数据库	(303)
13.2.3 和窗口有关的其它几条命令	(265)	17.2.1 dBASE 方式与SQL 方式的转换	(303)
第十四章 屏幕表格、报表和标签	(266)	17.2.2 SQL 数据库	(303)
14.1 屏幕表格的设计与使用	(266)	17.3 SQL 操作准备	(306)
14.1.1 如何进行屏幕表格设计	(266)	17.3.1 SQL 命令的键入	(306)
14.1.2 屏幕表格的使用	(272)	17.3.2 样本数据库的建立	(306)
14.2 报表的设计与打印	(273)	17.4 查询功能	(309)
14.2.1 报表的设计	(273)	17.4.1 SQL 运算符、函数和表达式	(310)
14.2.2 报表的打印	(275)	17.4.2 一般查询	(312)
14.2.3 报表的快速生成和打印	(277)	17.4.3 对显示结果进行排序	(319)
14.3 标签的设计与打印	(278)	17.4.4 记录的分组与统计运算	(322)
14.3.1 标签的设计	(278)	17.4.5 连接查询	(324)
14.3.2 标签的打印	(279)	17.4.6 嵌套查询	(328)
第十五章 数据库与其它应用软件 的接口	(280)	17.4.7 SAVE TO 子句与UNION 运算	(335)
15.1 数据库数据的传出	(280)	17.4.8 SQL 系统目录的查询	(339)
15.1.1 COPY 命令	(280)	17.4.9 FOR UPDATE OF 子句的作用	(340)
15.1.2 EXPORT 命令	(281)	17.5 操作功能	(341)
15.2 数据库数据的传入	(281)	17.5.1 插入	(341)
15.2.1 APPEND 命令	(281)	17.5.2 删除	(342)
15.2.2 IMPORT 命令	(282)	17.5.3 修改	(342)
15.3 dBASE/CHART-MASTER 图形 接口	(282)	17.6 定义功能	(343)

17.6.1 表的定义、修改与删除	(343)	18.1 dBASE IV 的安全措施	(360)
17.6.2 视图的定义与删除	(346)	18.1.1 dBASE IV 安全性概述	(360)
17.6.3 索引的定义与删除	(348)	18.1.2 组的概念	(361)
17.6.4 同义名的定义与删除	(349)	18.1.3 注册保密	(361)
17.7 控制功能	(349)	18.1.4 访问级别	(361)
17.7.1 授权	(350)	18.1.5 数据加密	(362)
17.7.2 收权	(351)	18.2 dBASE IV 安全系统的建立	(362)
17.8 SQL 与dBASE 命令的嵌套使用	(351)	18.2.1 启动PROTECT 命令及数据库管 理员口令	(362)
17.8.1 在 SQL 中使用dBASE 命令和函数	(351)	18.2.2 定义用户档案	(362)
17.8.2 在dBASE 中 SQL 的程序化使用 方式	(354)	18.2.3 定义文件特权和字段特权	(363)
17.9 几个实用命令	(357)	18.2.4 打印安全方案信息	(364)
17.9.1 DBDEFINE	(357)	18.2.5 退出 PROTECT	(364)
17.9.2 LOAD/UNLOAD	(357)	18.3 在网络 dBASE IV 上使用的命令 和函数	(365)
17.9.3 DBCHECK	(357)	18.3.1 用于网络的命令	(365)
17.9.4 RUNSTATS	(358)	18.3.2 用于网络的函数	(368)
17.10 SQL 宿主使用的一般方式	(358)	附录 出错信息	(370)
第十八章 网络 dBASE IV 与安 全性	(360)		

第一章 概 述

随着计算机科学技术的发展,计算机已从重点应用于科学计算领域转移到重点应用于数据处理领域。特别是微型计算机的普及,使得越来越多的数据处理和数据管理工作使用上了计算机。例如,学生成绩的管理、科研档案的管理、财务管理、仓库货物的管理、企业生产的计划管理、人事档案管理、图书管理等等。数据管理或者说数据资源管理对任何一个企业都是至关重要的,它直接影响着企业的生产和经济效益。而用于数据管理的最好工具就是数据库管理系统(DataBase Management System——DBMS)。适用于中、小型企业管理的数据库管理系统——dBASE III已在国内普及,本书则将向读者介绍dBASE的新一代产品dBASE IV。

1.1 数据库基础

1.1.1 数据库的发展和基本概念

数据处理是指人们在生产活动和社会活动中对数据进行收集、组织、加工、储存、抽取、传播等工作。其基本目的就是大量的、杂乱无章的、难以理解的数据中抽取并推导出对于某些特定的人们来说是有价值和有意义的数据,为进一步的活动提供决策的依据。

数据处理工作由来已久,根据它的发展过程和特点可以划分为三个不同的阶段:人工管理阶段、文件系统阶段和数据库系统阶段。在解释这几个阶段之前我们有必要解释一下数据的物理组织和逻辑组织的概念。

数据在物理存储设备(磁盘、磁带等)上的存放方式称为数据的物理组织,一般以文件形式组织。根据其组织结构的特点可分为顺序文件、索引顺序文件、散列文件等。数据在用户面前所呈现的组织方式称为数据的逻辑组织。对于一种数据的逻辑组织,可以用不同的物理组织来实现。物理组织的好坏影响系统的性能和效率,所以一种数据的逻辑组织在运行阶段中由于性能要求或存储设备的更新,就会引起数据的物理组织的改变,这种改变称为数据的再组织。用户在编制应用程序时,是根据数据的逻辑组织进行操作的。我们把应用程序对数据的物理组织的依赖程度称为数据的物理独立性,把应用程序对数据的逻辑组织的依赖程度称为数据的逻辑独立性。

在数据处理的人工管理阶段(至60年代早期),数据的逻辑组织和它的物理组织是相同的,计算机系统仅提供基本的输入、输出操作,应用程序员需亲自设计数据的物理组织。当数据的物理组织或存储设备发生变化时,其应用程序必须重新编制。另外,由于数据的物理组织是由应用程序员根据应用的要求设计的,所以很难实现多个应用程序共享数据资源。由此看来,在此阶段数据的独立性差,程序的维护代价很高;另外,数据冗余度高,即数据大量重复,给数据的维护带来很多困难。

数据处理到文件系统阶段(至60年代后期)有了很大改进, 设置了专门的数据管理软件——文件管理系统, 负责对数据进行管理, 提高了数据的物理独立性. 在数据的物理组织和逻辑组织之间由存取方法实现转换, 以使数据的逻辑组织和物理组织之间可以有所区别, 当物理组织改变时可不影响逻辑组织, 从而提高了数据的物理独立性. 在此阶段, 实现了以文件为单位的数据共享, 但未能实现以记录或字段为单位的数据共享, 数据的逻辑组织是根据具体的应用要求设计的, 数据仍存在大量的冗余.

为了满足数据处理的要求, 提出了更高的数据共享和更高的数据独立性, 以降低应用程序的开发和维护费用, 到60年代后期产生了数据库技术. 在此阶段, 数据不再面向应用进行组织, 而是面向数据组织数据, 从而提高了共享程度, 减少了数据冗余. 文件系统与数据库系统的根本区别在于: 在文件系统中各种文件的记录之间是无联系的, 而在数据库系统中不同类型的记录之间允许有联系, 这种联系反映了自然界诸对象之间的关系. 只有面向数据组织数据才能反映数据之间的联系, 而实现联系正是数据库系统所要解决的问题.

我们把反映整个数据库数据之间的逻辑关系的数据结构称为全局逻辑结构, 又称为概念模式或概念数据库; 把具体用户涉及的部分数据结构称为数据库的局部逻辑结构, 又称为外部模式或外部数据库. 图1-1示意了数据库的抽象层次. 从中可以看出, 数据库共有三个抽象层次, 其中概念数据库是全部概念(逻辑)文件的集合, 外部数据库是从概念数据库派生出来的, 每个外部数据库是一个具体的用户使用的概念文件(这样的概念文件称作外部文件)的集合. 概念数据库和外部数据库都只是一个框架, 只有存储数据库是物理上真正存在的. 概念数据库是存储数据库的抽象表示, 而每个外部数据库又是概念数据库的部分抽取.

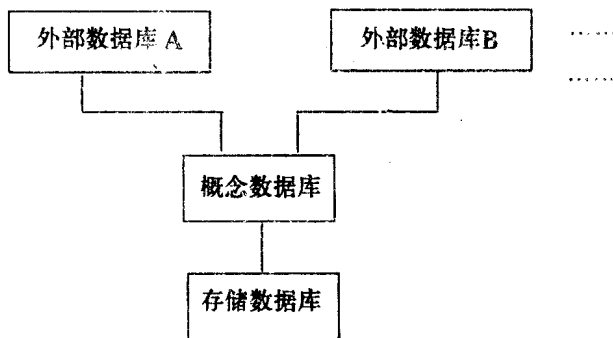


图1-1 数据库的抽象层次

更进一步, 我们可以把这种具有三个层次的数据库称作三级数据库. 对三级数据库中的每一级数据库都有一个框架, 或者叫结构, 或者叫模型. 显然, 对每一级模型都必须能够用数据描述语言给出精确定义, 我们把这种定义称作模式. 确切地说, 定义存储模型的模式称作存储模式; 定义概念模型的模式称作概念模式, 也简称模式; 定义外部模型的模式称作外部模式, 也称作子模式. 子模式是概念模式的子集, 它可以从概念模式推导出来. 图1-2示意了各级模式的关系. 根据前面的介绍可以看出, 外部模式和概念模式之间的映象提供了逻辑数据独立性, 概念模式和存储模式之间的映象提供了物理(存储)数据独立性.

一个完备的数据库应该具有以上三个层次. 能够完成管理数据库并能向用户提供各种操作的软件称作数据库管理系统. 数据库管理系统的总体结构如图1-3所示. 为了操作数据库, 用户(或用户程序)需要数据库控制系统(DBCS)的服务, DBCS完成外部模式和概念模式之间的转换. DBCS必须通过数据库存储系统(DBSS)来操作存储数据库, DBSS完成概念模式和存储模式之间的转换, 并通过存取方法进行记录存取. DBCS和DBSS共同构成数据库管理系统(DBMS).

个“仓库”中的数据彼此之间是有联系、有规则的，不是独立的、杂乱无章的。为了使用“仓库”中的数据，必须有一个对这些数据进行维护和访问的机构，这个机构就是数据库管理系统。我们以图书馆为例来说明数据库和数据库管理系统之间的关系。众所周知，图书馆是负责存储和借阅图书的部门，书库是各类图书的集合。图书馆若要很好地为读者服务，首先必须有图书馆工作人员收集图书并为图书建立完善的书目，其次要按照一定的规则分别存放不同类别的图书，最后还要规定图书的借还手续，实施图书馆的职能——借阅图书。从中可以看出，书库相当于数据库，图书馆管理规则和工作人员相当于数据库管理系统。

这样我们可以定义：数据库是存储在计算机上的有规则的、关联的数据的集合。数据库管理系统是数据库管理软件，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。而数据库系统则是指计算机系统中引入数据库后的系统构成，它包括支持数据库管理系统的硬件和软件环境、数据库管理系统、数据库及使用和管理数据库的人。

1.1.2 三种范围与信息结构

在数据处理中将涉及不同的范围。例如，在图书馆管理中首先涉及的是图书的购买、图书的存放、图书的借阅等，这种管理称为现实世界管理。在现实世界管理中，被管理的对象称为实体(Entity)。实体可为实际存在的而又可以区分的客观事物，如图书、学生、教师等等，也可抽象的事件，如足球比赛等。一个实体具有一定的特征(性质)，如图书有书名、作者、定价、页数等；再如足球比赛有比赛时间、地点、参加比赛的队等。具有相同特征的实体的集合称为实体集。例如在图书馆中，所有图书可组成图书实体集，所有图书管理员可组成图书管理员实体集等。构成实体集的实体必须是可区分的，也就是说必须有特征能够区分实体集中的一个个实体，这种特征称为标识特征。

在进行现实世界管理时，这些客观事物必然在人们头脑中产生反映，我们把这种反映称为信息。我们需要对这些信息进行记录、整理和归类，使之成为格式化的信息。如在图书管理中，可通过图书编目对图书进行管理，通过借阅卡对图书借阅进行管理，这种管理可称为信息管理，所以信息是现实世界状态的反映，信息管理是现实世界管理的反映。要管理好现实世界，必须借助信息管理。在这个范围内用实体记录表示实体，用实体记录集表示实体集，用属性表示实体集的特征，用标识属性表示标识特征。比如，书名、作者、定价、页数都是表述图书实体集的属性。

当数据管理借助计算机进行时，则进入机器世界。由于计算机只能处理数据化的信息，所以对信息世界的信息必须进行数据化，数据化后的信息称为数据，因此我们可以说数据是信息的符号表示。在机器世界范围内，实体记录用记录值表示，实体记录集用文件表示，实体记录的属性用数据项表示，属性值用数据项值表示，标识属性用关键字(码)表示。

对在三种不同范围使用的不同术语，在表1-1中做了简单归纳。

在现实世界中实体不是孤立的，实体和实体之间是有联系的。例如，图书管理员和图书之间就有联系，这种联系是图书管理员管理图书。实体之间的联系是复杂的，根据联系的特点可将联系分为三种基本类型，即一对一的联系、一对多的联系和多对多的联系。

一对一的联系是最简单的联系。假设有 A, B 两个实体集, 如果给定 A 中的一个实体, B 中仅有一个实体与之对应; 反之, 如果给定 B 中的一个实体, A 中也仅有一个实体与之对应, 则 A 和 B 之间的联系是一对的。例如, 有图书馆和馆长两个实体, 那么这两个实体之间的联系是一对的。因为一个图书馆只有一个馆长。一个人一般也只会在一个图书馆任馆长。

一对多的联系是最常见的联系。假设有 A, B 两个实体集, 如果给定 A 中的一个实体, B 中可能有多个实体与之对应, 但是, 给定 B 中的一个实体, 在 A 中只可能有一个实体与之对应, 则 A 和 B 之间的联系是一对多的。例如, 图书馆和图书馆管理员之间的联系是一对多的联系。因为一个图书馆可有多个图书管理员, 而一个图书管理员只在一个图书馆工作。

多对多的联系是比较复杂的一类联系。假设有 A, B 两个实体集, 如果给定 A 中的一个实体, 在 B 中可能有多个实体与之对应, 反之, 给定 B 中的一个实体, 在 A 中也可能有多个实体与之对应, 则 A 和 B 之间的联系是多对多的。例如, 图书馆管理员和图书之间的联系是多对多的。因为一个图书馆管理员可以管理多种图书, 而一种图书也可由多个管理员来管理。

我们常用 1:1, 1:N 和 M:N 分别来表示一对一、一对多和多对多联系。

现实世界中实体与实体之间的联系反应在机器世界中就是记录与记录之间的联系, 所以说数据库中的数据是有联系的。

表 1-1 三种不同范围使用术语对照

现实世界	信息世界	机器世界
所有客观对象	条理化信息	数据库
实体集	实体记录集	文件
实体	实体记录	记录
特征	属性	数据项或字段
标识特征	标识属性	关键字

1.1.3 三大数据模型及描述数据的方法

上一小节我们叙述了信息结构, 在机器世界我们用数据模型表示信息结构。数据模型是数据库系统的基础。目前常用的数据模型有三种, 即层次模型、网状模型和关系模型。我们说关系数据库管理系统, 就是说该数据库管理系统所支持的数据模型是关系的, 或者说该数据库管理系统采用的是关系方法。类似地, 我们有网状数据库管理系统和层次数据库管理系统。目前关系数据库已占主导地位。

1. 层次模型与层次方法

用树型结构来表示实体之间联系的模型称为层次模型。

构成层次模型的树是由结点和连线组成的。结点表示实体集(文件), 连线表示相连两实体之间的联系, 这种联系只能是 1:n 联系。通常把表示 1 的实体放在上方, 称为父结点; 而表示 n 的实体放在下方, 称为子结点。

层次模型表示 1:n 联系非常直接、方便。但由于层次模型有以下两点限制:

- 有且仅有一个结点无父结点, 该结点为根结点;
- 其它结点有且仅有一个父结点。

这样就使得 m:n 联系不能直接用层次模型表示, 必须用特殊的方法进行处理。

图 1-4 是一个层次模型的示例。这里 R1 是根结点, R2 和 R3 的父结点是 R1, R4 和

R5 的父结点是 R3。

这种模型能够如实地表达现实世界的层次结构(如企、事业单位的行政隶属机构),层次清楚,各结点之间的联系简单,只要知道每个结点(根结点除外)的父结点,就可以描绘出整个模型的结构。

支持层次模型的典型系统是 IBM 公司的 IMS (Information Management System)。

从逻辑上看,IMS 对记录有两种不同的组织方法:一是集中同型的逻辑记录构成逻辑文件,二是集中有联系的记录构成层次型。逻辑文件反应了实体集,层次型反应了联系。

从物理上看,IMS 是按照层次型或值来组织记录的,具体是用指针来实现的。

2. 网状模型与 CODASYL 网络方法

如果取消层次模型中的两个限制,即允许每一个结点可以有多个父结点,这样便形成了网络。我们把用网状结构来表示实体之间联系的模型称为网状模型。图 1-5 是一个网状模型的示例,这里 R3 有两个父结点 R1 和 R2, R4 有两个父结点 R3 和 R2。

由于网状模型没有层次模型的那两点限制,所以可以直接表示 $m:n$ 联系。但由于 $m:n$ 联系实现起来非常复杂,所以很多实际的网状数据库系统对这点都做了限制,即也只允许处理 $1:n$ 联系。

网状模型和层次模型在本质上是一样的。从逻辑上看,它们都是用结点表示实体集,用连线表示实体之间的联系;从物理上看,层次模型和网状模型都是用指针来实现两个文件之间的联系。其差别在于网状模型中的连线或指针更加复杂、更加纵横交错,从而使数据结构更加复杂。

在网状模型中我们同样使用父结点和子结点这样的术语,并且同样把父结点安排在子结点的上方。

网状模型的典型系统是 CODASYL (Conference On DATA SYstem Languages 的缩写)系统,它是 CODASYL 组织的标准建议的具体体现。CODASYL 建议认真遵循了 ANSI/SPARC 的三级数据库的系统方案。也就是说,在 CODASYL 数据库中,有一个概念模式、一个存储(内部)模式和多个外部(子)模式。

像 IMS 系统一样, CODASYL 在逻辑上也把记录按两种不同的方法进行组织:一是集中同型的逻辑记录构成逻辑文件;二是集中有联系的记录构成 CODASYL 系(SET)。形成系的逻辑记录具体反应了实体集之间的联系。

以上两种模型统称为格式化模型,它们对用户有较高的技术要求,使用起来比较困难,目前正逐渐被关系模型取代。

3. 关系模型和关系方法

表格是大家熟悉的数据表示方法。通俗讲关系就是表格,所以我们把用表格数据来表示

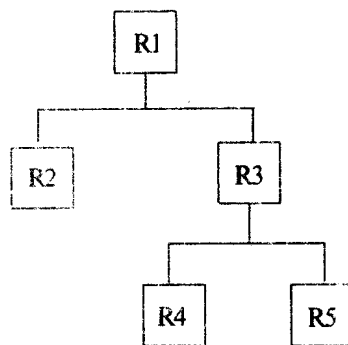


图1-4 层次模型示意

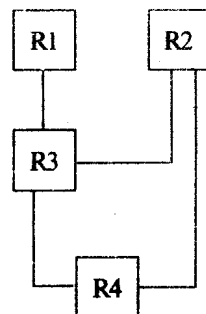


图1-5 网状模型示意

实体和实体之间联系的模型称为关系模型。

在层次模型和网状模型中，我们可以理解为：文件中存放的是数据，各文件之间的联系是通过指针来实现的。而在关系模型中，文件中存放两类数据：一是实体本身的数据，二是实体之间的联系。也就是说，在关系模型中实体以及实体间的联系都是用关系来描述的。具体的联系是通过连接字段来实现的。

关系模型源于数学，它把数据看成是二维表中的元素，而这个二维表就是关系。表中的每一行是一个元组，它相当于一个记录值；每一列是一个属性值集，列可以命名，称为属性名，这里的属性和前面讲的实体的属性及记录的字段意义相同。

关系是元组的集合，如果表格有 n 列，则称该关系为 n 元关系。关系应满足：

- 表中的每一列都是不能再分的基本字段；
- 各列被指定一个相异的名字；
- 各行相异，不允许重复；
- 行、列次序均无关。

综合以上四点，我们可以说：一个关系是一个逻辑文件，该文件中的每个记录是唯一的，所有记录具有相同个数和类型的字段，也就是说，所有记录有同样的固定长度和格式。

对于用户，关系方法似乎非常简单，但是关系数据库系统却非常复杂。这种方法之所以对用户简单，是因为把大量的困难转嫁给了数据库管理系统。由于这个原因，尽管在 20 多年前就有了关系数据库的设想，但是在商业上可行的关系数据库系统的开发却花费了比任何人的想像都要更长的时间，在 70 年代初对关系数据库系统做了大量研究，但一直到 70 年代末，甚至 80 年代初，各类关系数据库管理软件包才真正成为商品并投入使用。

关系模型最容易被理解和使用，已成为当今数据库的主流。

1.1.4 关系模型的几个概念和术语

由于 dBASE IV 是关系数据库管理系统，所以我们在本小节再对关系数据库的一些概念和术语做一些进一步的解释和定义。

1. 笛卡尔积与关系定义

设 $D_1, D_2, \dots, D_n$ 为任意集合，定义 $D_1, D_2, \dots, D_n$ 的笛卡尔积为：

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

其中每一个元素 $(d_1, d_2, \dots, d_n)$ 叫做一个 n 元组 (n -tuple)，简称元组。元组中的每一个 d_i 叫做元组的一个分量。

比如我们设

$$D_1 = \{ P_2, P_4, P_7, P_9 \}$$

$$D_2 = \{ \text{晶体管, 电感器, 电容器} \}$$

则

$$D_1 \times D_2 = \{ (P_2, \text{晶体管}), (P_2, \text{电感器}), (P_2, \text{电容器}), \\ (P_4, \text{晶体管}), (P_4, \text{电感器}), (P_4, \text{电容器}), \\ (P_7, \text{晶体管}), (P_7, \text{电感器}), (P_7, \text{电容器}), \dots \}$$

$(P9, \text{晶体管}), (P9, \text{电感器}), (P9, \text{电容器})\}$

笛卡尔积实际上是一个二维表，如图 1-6 所示。图 1-6 中表的任意一行就是一个元组，它的第一个分量来自 $D1$ ，第二个分量来自 $D2$ 。笛卡尔积就是所有的这样的元组的集合。

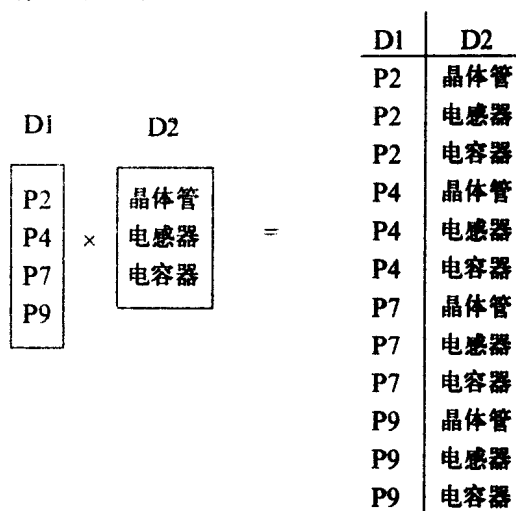


图 1-6 笛卡尔积

笛卡尔积 $D1 \times D2 \times \dots \times Dn$ 的任意一个子集都称为 $D1, D2, \dots, Dn$ 上的一个 n 元关系。

同样可以把关系看成二维表，给表的每一列取一个名字，称为属性， n 元关系有 n 个属性，属性的名字要唯一。属性的取值范围 $Di(i=1, \dots, n)$ 叫做值域 (Domain)。

比如对刚才的例子，我们取子集

$R = \{(P2, \text{晶体管}), (P4, \text{电感器}), (P7, \text{电感器}), (P9, \text{电容器})\}$

构成一个关系，二维表的形式如表 1-2 所示，我们把第一个属性命名为器件号，第二个属性命名为器件名称。

表 1-2 一个关系

器件号	器件名称
P2	晶体管
P4	电感器
P7	电感器
P9	电容器

2. 关键字

如果一个属性集的值能唯一标识一个关系的元组，而又不含有多余的属性值，则称该属性集为候选关键字。

有时一个关系中有多个候选关键字，这时可以选择其中一个作为主关键字，简称关键字。每一个关系都有一个并且只有一个主关键字。

3. 关系模式

关系模式就是二维表的表框架，相当于记录格式。设关系名为 REL ，其属性为 $A1, A2, \dots, An$ ，则关系模式可表示为：

$REL(A1, A2, \dots, An)$

当然对每个 $Ai(i=1, \dots, n)$ 还包括该属性到值域的映象，即属性的取值范围。

4. 关系模型

关系模型是所有的关系模式、属性名和关键字的汇集，是模式描述的对象。

5. 关系数据库

对应于一个关系模型的所有关系的集合称为关系数据库。

关系数据库下的术语与其它数据库中用到的术语是类似的，现列举如下：

属性——数据项(字段)；

元组——记录(值)；

关系——文件(值)；

关系模式——记录格式(类型)；

关系名——记录名；

数据库模式——概念模式。

最后我们概括一下关系的性质：

- 列是同质的，即每一列中的分量是同类型的数据，来自同一个值域。
- 不同的列可出自同一个域，每一列称为属性，要给予不同的属性名。
- 列的次序无关紧要，即列的次序可以任意交换。
- 任意两个元组不能全同。
- 行的次序无关紧要，即行的次序可以任意交换。
- 每一分量必须是不可分的数据项。
- 每个关系都有一个主关键字唯一标识它的各个元组。

1.2 dBASE IV 简介

dBASE IV 是 dBASE 系列产品的最新一代。它是以 dBASE III PLUS 为基础的全新的数据库管理系统。它增加了全新的用户接口，完善的举例查询和 SQL 能力，另外还增加了完善的应用程序生成系统，使得应用更加方便。

1.2.1 dBASE IV 的新特点

1. 新的用户接口

1) 控制中心

可以说控制中心是 dBASE IV 的核心，是用户访问 dBASE IV 的独立系统。用户可以通过控制中心访问除 SQL 以外的所有 dBASE IV 单元。比如，通过控制中心可以打开和关闭文件，浏览数据，修改库结构，达到设计屏幕进行屏幕表格设计、报表格式设计、标签格式设计、举例查询设计和应用程序设计等，还可以运行程序、管理文件目录和数据库目录等。

控制中心采用全菜单驱动方式工作，非常直观和容易掌握，是一个全新的用户友好接口。

2) Browse 和 Edit

在 dBASE IV 中，Browse 和 Edit 就好像同一枚硬币的两个表面被紧密地联系在一起，用户可以使用 F2 键方便地在 Browse 显示方式和 Edit 显示方式之间进行切换。

这两条命令都扩充了条型菜单，Memo 字段也更容易编辑并且可以检索。