

FOXBASE⁺ 教程

● 主编 张大方 罗可

湖南师范大学出版社



TP311.13
41

FOXBASE+ 教程

主编 张大方 罗 可

编委 王杰文 王耀南

史伟奇 杨光辉

李晓秋

湖南师范大学出版社

【湘】新登字 011 号

内 容 简 介

本书全面介绍了 FOXBASE+2.1 的基本内容。包括数据库系统基础知识, FOXBASE+ 概述, 数据库的建立和修改, 数据库文件的使用, FOXBASE+ 函数, FOXBASE+ 程序设计, 输入输出设计, 实用编程技巧, FOXBASE+ 与其它语言的接口, 应用实例, 多用户 FOXBASE+ 等。

本书是在作者多年教学研究实践基础上编写而成的, 详略恰当, 适合于教学。可作为大专院校相关课程的教材或参考书, 及各类 FOXBASE+ 应用培训班的教材。特别对于广大的计算机用户来说, 本书是一本实用的自学或上机参考资料。

FOXBASE+ 教程

主 编 张大方 罗 可

责任编辑 刘 晓 李 琪

湖南师范大学出版社出版发行

(长沙岳麓山)

湖南省新华书店经销 长沙东方印刷厂印装

787×1092 16 开 17 印张 450 千字

1994 年 4 月第 1 版 1994 年 12 月第 2 次印刷

印数: 8121~16150 册

ISBN 7-81031-366-5/TP·006

定价: 13.80 元

前 言

计算机应用目前已深入到社会的各个领域,将逐渐成为人们工作、学习、生活的必备工具。导致这种发展的动因有许许多多,但其中两种技术起着举足轻重的作用。第一是微型机技术的发展,它使得计算机这种神秘而又昂贵的高尖技术深入普及到社会的各个角落,甚至寻常百姓家;第二则是数据库技术的发展,它使得人们可以方便、快捷、高效地使用计算机完成各类日常管理工作,使得计算机能充分地为社会服务。为满足社会的需要,使更多的读者逐步掌握、应用这一现代技术,我们组织编写了该系列教材:(1)《微型计算机实践教程》(国防科大出版社 1993 年 1 月出版,至今已第四次印刷,发行 5 万余册,注:至 94 年 12 月本书第二次印刷时,发行已达 10 万余册);(2)《FOXBASE+ 教程》;(3)《计算机会计信息系统及其实践》(国防科大出版社 1992 年 10 月出版)。其中教材(1)是微型机技术与操作的一本入门教材,教材(2)是数据库技术与应用的一本入门教材,教材(3)是在学习教材(1)、(2)的基础上,侧重会计电算化方面读者所必需的实用教材。

数据库技术是数据管理的最新方法,也是信息管理中的一项非常重要的新技术。数据库管理系统是帮助人们处理大量信息,实现管理科学化和现代化的强有力工具,数据库管理系统已发展成为通用的系统软件。目前我国流行最广、应用面最大的是微机关系数据库管理系统 dBASE III,这是因为 dBASE III 具有很强的数据管理功能,编程方便、灵活,并具有通俗易懂、易学易会、易于维护和便于推广等特点,因而深受广大用户的欢迎。但 dBASE III 存在计算能力弱、处理速度较慢等弱点,大大限制了其应用范围。因此人们迫切需要更好的类似软件问世,FOXBASE+ 就是这类软件最杰出的代表作之一。它与 dBASE III 兼容,但速度快 6~7 倍,并在性能和功能上作了许多重大改进和扩充。因此,FOXBASE+ 一经推出便引起了人们的极大兴趣,得到普遍的应用。

为使众多的读者在学习 FOXBASE+ 之前掌握必备的数据库基础知识,我们在第一章浓缩了数据库技术的基本理论与方法,读者可根据需要选读其中部分或全部内容,亦可跳过此章;第二章概述了 FOXBASE+ 基本概念;第三、四、五、六章分别介绍了 FOXBASE+ 的基本命令;第七章对 FOXBASE+ 的程序设计方法、命令进行了深入地阐述;第八、九章介绍了 FOXBASE+ 的输入输出设计、系统参数设置;第十章精选了使用最广泛并具有一定代表性的程序实例及编程技巧,读者从中可获得许多有益的启示;第十一章介绍了 FOXBASE+ 与其它语言的接口;第十二章介绍了一个完整的 FOXBASE+ 应用实例;第十三章简单介绍了多用户 FOXBASE+。附录给出了全书所有 FOXBASE+ 的命令索引,ASC II 码字符表,FOXBASE+ 错误信息及说明,以供读者查阅。本书可作为不同专业、不同层次《数据库及应用》、《FOXBASE+ 原理及应用》、《微机管理》等课程的教材或教学参考书,教材内容可根据具体情况适当取舍。在教学过程中,要强调上机实践环节及加强课后练习,以便收到更好的效果。

全书共十三章。第一、十、十一章由张大方(湖南大学计算机科学系, 邮码: 410082, 电话: (0731)8883171 转 2273)编写; 第二、三、四、七章由罗可(长沙水电师院)编写; 第五章由杨光辉(深圳大学)编写; 第六章由王耀南(华东地质学院)编写; 第八、九章由王杰文(湖南大学衡阳分校)编写; 第十二章由李晓秋、张洪生同志(湖南大学)编写; 第十三章由史伟奇(湖南省公安专科学校)编写。最后由张大方、罗可总纂并定稿。

由于编者水平有限, 加上时间仓促, 书中难免有缺点与错误, 恳请读者批评指正。

编 者

1994年2月28日于长沙

目 录

第一章 数据库系统基础知识

§ 1.1 概述	(1)
§ 1.2 数据库管理系统	(6)
§ 1.3 数据模型	(7)
§ 1.4 关系代数	(14)
§ 1.5 SQL	(17)
§ 1.6 关系数据库设计理论	(21)

第二章 FOXBASE+ 概述

§ 2.1 FOXBASE+ 2.1 简介	(30)
§ 2.2 FOXBASE+ 2.1 基本概念	(32)
§ 2.3 FOXBASE+ 的常量、变量和表达式	(39)

第三章 数据库的建立和修改

§ 3.1 数据库的建立	(51)
§ 3.2 数据库的基本操作	(56)

第四章 数据库文件的使用

§ 4.1 分类与索引	(73)
§ 4.2 数据查询	(79)
§ 4.3 复制数据库	(84)
§ 4.4 数据库的统计及汇总	(89)
§ 4.5 多工作区操作	(92)

第五章 FOXBASE+ 函数

§ 5.1 数学运算函数	(103)
§ 5.2 字符操作函数	(105)
§ 5.3 日期时间函数	(108)
§ 5.4 转换函数	(109)
§ 5.5 数据库函数	(111)
§ 5.6 测试函数	(115)
§ 5.7 输入函数	(118)
§ 5.8 SYS 函数	(119)

第六章 数据库的辅助操作

§ 6.1 文件操作	(122)
§ 6.2 清理命令	(124)
§ 6.3 其它命令	(125)

第七章 FOXBASE+ 程序设计

§ 7.1 FOXBASE+ 程序及其特点	(128)
§ 7.2 命令文件的建立与执行	(130)
§ 7.3 顺序结构程序设计与交互式命令	(132)

§ 7.4	分支程序设计	(135)
§ 7.5	循环程序设计	(139)
§ 7.6	内存变量数组	(144)
§ 7.7	过程及其调用	(148)
§ 7.8	过程文件生成器及文件编译器的应用	(157)
§ 7.9	其它命令	(159)
第八章	FOXBASE+ 输入输出设计	
§ 8.1	输入输出格式设计	(162)
§ 8.2	报表格式文件	(173)
§ 8.3	标签格式文件	(177)
§ 8.4	用 CCED 输出数据报表	(179)
第九章	系统参数设置	
§ 9.1	SET 命令集	(186)
§ 9.2	配置文件	(195)
§ 9.3	SET 命令分类索引表	(199)
第十章	实用编程技巧与实例	
§ 10.1	菜单技术	(201)
§ 10.2	口令技术	(203)
§ 10.3	全屏幕编辑	(206)
§ 10.4	查询技术	(209)
§ 10.5	表格程序文件	(212)
§ 10.6	绘图技术	(215)
§ 10.7	宏代换应用技巧	(216)
§ 10.8	提高 FOXBASE+ 应用程序运行效率的若干途径	(218)
§ 10.9	其它	(220)
第十一章	FOXBASE+ 与其它语言的接口	
§ 11.1	调用外部程序	(223)
§ 11.2	调用二进制程序文件	(223)
§ 11.3	FOXBASE+ 与高级语言的间接数据转换	(224)
第十二章	应用实例	
§ 12.1	系统设计	(229)
§ 12.2	系统源程序	(233)
第十三章	多用户 FOXBASE+	
§ 13.1	多用户系统概述	(248)
§ 13.2	加锁与解锁	(250)
§ 13.3	多用户命令	(257)
§ 13.4	多用户函数	(259)
附录一	FOXBASE+ 命令索引	(261)
附录二	ASC II 码字符表	(262)
附录三	FOXBASE+ 错误信息及说明	(263)
参考文献		(266)

第一章 数据库系统基础知识

§ 1.1 概 述

在计算机应用的广泛领域中,数据处理所占比重越来越大,进入九十年代后已达 70~80%。而数据库系统是数据处理的核心部分,因而学习和掌握数据库系统的原理和技术,已逐渐成为人们的共识。本章将介绍数据库系统的基础知识,包括数据库技术的概述、数据库管理系统、数据模型、关系代数、SQL、关系数据库设计理论等,为今后学习关系数据库系统 FOXBASE+ 奠定基础。

数据在人类社会的发展中是一种非常重要的资源。因而如何妥善地保存和科学地管理这些数据一直成为人们十分关注的课题。数据库系统就是迄今为止人们研究这个课题最成功的成果之一。数据库系统研究的对象就是如何高效、巧妙地进行数据处理,而又花费最少。数据库技术已成为计算机领域最主要的技术之一,它是软件方面的一个独立分支。

一、数据库概念

数据库是实现有组织地、动态地存储大量相关数据的集合。其主要特征是:数据的充分共享,具有较高的数据独立性和数据的安全性、完整性。事实上,现在数据库的定义是各式各样的,不尽相同。这是人们从不同角度用不同观点来看待数据库的结果。另一原因是数据库技术本身也是逐渐形成,直到现在还在发展之中。所以很难用几句话严格、简明地概括它的全部特征。

为使读者能从感性上对数据库有个完整的了解,我们以图书馆为例进行比较说明。事实上,数据库与图书馆在许多方面是有相似之处的。众所周知,图书馆是存储和负责借阅图书的部门;而数据库则是存储数据并负责用户访问数据的机构。正象图书馆不能简单与书库等同起来一样,我们也不能把数据库仅仅理解为存储数据的集合,而应视为一个系统,即数据库系统。就图书馆而言,如果把书籍胡乱地堆放在书库中,几乎无法从数以百万册计的浩瀚书海中查找出读者要借的一本书来,因此,没有一套完整的书卡作为图书馆藏书的模型,查借书籍就很困难,管理员也很难掌握藏书全貌。如果不知道书卡与书架的对应关系,管理员也难以按借书单找到该书的存放位置,一个图书馆要想很好地为读者服务,必须完成以下工作:

(1) 建立完善的书卡。书卡的内容和格式常包括:书号(分类号)、书名、作者名、出版社名、出版时间、内容摘要和其他细节。有时为了方便读者,也按不同分类编排书卡,如以书名、作者名或其它为索引进行编排。

(2) 图书应有组织地存放在书库中。图书馆藏书数量很大,书库中房间、书架很多,需要按照一定的顺序和规则(物理结构)分放图书,并列出具类书籍存放的对应关系表,使管理人员能按此表快速查找。

(3) 规定借阅权限。不同类型读者的借阅权限不同。如善本书只供特定的人借阅;机密图书只供有特权的人借阅;某些书只供读者在馆内翻阅等。

(4) 建立周密的借阅管理制度。规定图书的借还手续,读者借书要先出示借书证,图书管理员验明读者身份和借阅权限后,根据读者填写的借书单(访问请求),按照书籍与书架的对应关系表,到书库中查找所借图书交与读者(响应),并作某些登记(日志);还书时管理员要按书号把交还的图书送回原来存放的书架位置上;如果还书较多,管理员又要立即“响应”其他读者借书,那么,他也可以把还书按归还次序放到一个“当日暂存书架”上,待一天末了有空闲时,再清理“当日暂存书架”上的还书并送回到书库中存放它们的原来位置上。

对数据库来说,也要完成类似于图书馆的上述工作。(1) 数据库要建立数据模型,使用户可以根据数据模型访问数据库中的数据(如检索、插入、删除和修改),而不必关心数据在数据库中的物理存储位置,就象读者可以按书卡填写借书单,而不用顾及书籍存放在书库的什么位置一样。当然数据模型也要象书卡那样能反映各种数据之间的内在联系。(2) 数据也应有组织地存放在存储设备上,并建立数据模型到物理存储位置的对应表(这种对应称为映射),使数据库管理系统能够按照用户的访问请求,找到被访问数据的存储位置。建立数据模型和设计数据的物理存储(组织)方法,其目的是使用户对数据的应用与数据的存放位置和存储结构无关,后者的变动不影响前者(正象改变图书的存放位置不影响读者按书卡借书一样),这一点也称为数据独立性,这是数据库的重要特征之一。现用表 1.1 来归纳数据库与图书馆两者的类比。

表 1.1 数据库与图书馆类比

序号	数 据 库	图 书 馆
1	数 据	图 书
2	外 存	书 库
3	用 户	读 者
4	用户标识	借 书 证
5	数据模型	书 卡 格 式
6	数据库管理系统	图书馆管理员
7	数据的物理组织方法	图书的物理存放办法
8	用户对数据库的操作 (使用计算机语言检索、 插入、删除、修改)	读者对图书馆的访问 (用普通语言) 借书、还书
9	第 8 项独立于第 7 项	第 8 项独立于第 7 项

二、数据管理技术的进展

数据管理指的是对数据的组织、编目、存储、检索和维护等,它是数据处理的中心问题,数据管理随着计算机软件 and 硬件的发展而不断发展。三十年来经历了如下三个阶段:人工管理阶段,文件系统阶段,数据库系统阶段。

(一) 人工管理阶段(五十年代中期以前)

这一阶段数据处理的方式是批处理,其数据管理的特点如下:

1. 数据不保存。因为计算机主要用于科技计算,一般不需要将数据长期保存。只在计算某一课题时将数据输入,用完就撤走。

2. 没有软件系统对数据进行管理,程序员不仅规定数据的逻辑结构,而且在程序中还

要设计物理结构,包括存储结构、存取方法、输入输出方式等。程序中存取的子程序也随着存储的改变而改变,即数据与程序不具有独立性。

3. 这一时期基本没有文件的概念,即使有文件,也大多是顺序文件。

4. 一组数据对应于一个程序,即数据是面向应用的。程序与程序之间有大量重复数据。

(二) 文件系统(五十年代后期~六十年代中期)

这一阶段在软件方面有了操作系统,文件系统包含在操作系统之中,这一阶段的数据管理有如下几个特点:

1. 由于计算机大量用于数据处理等方面,数据需要长期保留在外存上反复处置,即对文件进行查询、修改、插入、删除等操作。

2. 由于有软件对数据进行管理,程序和数据之间有存取方法进行转换,有共同的数据查询修改的例行程序存放在程序库中。文件的逻辑结构与存储结构有一定区别,即程序与数据有一定的独立性。

3. 文件已经多样化。由于已有了直接存取设备,也就有了索引文件、链接文件、直接存取文件等。

4. 上述各点比第一阶段有很大改进。但这种方法仍存在着很大弱点,即文件本身还是基本上对应于一个或几个应用程序,或者说数据还是面向应用的。文件仍然是一个不具有弹性的无结构的信息集合,存在着冗余度大、空间浪费、不易扩充、修改费时间、有可能引起不相容等缺点,从而反映不了现实世界事物之间广泛的内在联系。

5. 数据的存取基本上还是以记录为单位。

(三) 数据库系统(六十年代后期)

由于数据管理规模更加庞大,数据量急剧增加,共享性更强。硬件方面又有了大容量的磁盘。由此就发展了数据库这样的数据管理技术。其重要标志为:

(1) 1968年美IBM公司的数据库管理系统IMS(INFORMATION MANAGEMENT SYSTEM)问世,它是层次数据库的典型代表。

(2) 1969年美CODASYL委员会公布了它的研究成果DBTG报告,它是网络数据库中的经典报告。

(3) 1970年起,美IBM公司的高级研究员E. F. Codd连续发表了一系列论文,奠定了关系数据库的理论基础。

概括起来,数据库技术的管理方式具有如下特点:

1. 复杂的数据模型(结构)。文件是相同记录的集合,而数据库可包含各种不同的记录,如一个人的工资记录、人事记录和业务记录等。记录还可是变长的,数据库中的数据被结构化了。这就要求在描述数据时不仅描述数据本身,还要描述数据之间的联系,这种联系是通过存取路径来实现的。通过一切存取路径来表示自然的数据联系是数据库与传统文件根本区别。这样就大大减少了数据的冗余度。此外由于从整体上看数据,所以数据不再是面向应用,而是面向系统,弹性大,可扩充,使用灵活。数据结构化是数据库主要特征之一。

2. 数据的独立性,这是数据库的另一个主要特征。(具体含义见§1.2)

3. 由于在数据库中强调共享,因此必须提供以下三个方面的数据控制功能。

(a) 数据的安全性。数据的安全性是指保护数据以防止不合法的使用。这就要采取一定

的安全保密措施,如用口令或其它手段检查用户身份,检查用户能否查询或修改数据。检查通过,才能执行允许的操作。

(b) 数据的完整性。数据的完整性包括数据的正确性、有效性、相容性,即系统有检验的措施,以控制数据在一定的范围内方为有效,或这一部分数据与另一部分数据之间必须满足一定关系等。

(c) 并发控制,即避免并发程序之间互相干扰。

4. 对数据的存取不一定以记录为单位。

从文件技术发展数据库技术是信息处理领域的一个重大变化。在文件技术阶段,信息处理的传统观点如图 1.1 (a) 所示,人们关注的中心问题是系统功能的设计,因此程序设计处于主导地位,数据只起着服从程序设计需要的作用;而在数据方式下,信息处理的传统观点已为图 1.1 (b) 所示的新体系所取代,数据开始占据了中心位置。数据的结构设计、一致性和安全性成为信息系统首先关注的问题,而利用这些数据的应用程序设计则退居以既定的数据结构为基础的外围地位。

三、信息结构

信息是向人们提供关于现实世界新的事实的知识,数据是载荷信息的各种物理符号。因此在数据处理中我们首先要将现实世界转换为信息世界,然后再将信息世界转换为机器世界。

现实世界的事物经过人脑的选择、命名、分类的认识过程之后,进入信息世界。信息世界的主要对象是实体 (Entity),实体是我们所关心的“事物”。这个“事物”可以指人,也可指物;可以指实际的东西,也可以指概念性的东西;还可以指“事物”与“事物”间的联系。

实体由若干个属性组成,属性是事物的某一方面的特征。例如姓名、学号、年龄、性别、英语成绩、导师姓名等等是学生的几个属性,每一个属性都有一个取值范围称为属性值的域。如性别的属性值的域为(男、女),英语成绩的属性值的域为 0 到 100 之间的整数等。

由若干个属性的属性值所组成的集合表征一个实体,而由若干个属性所组成的集合表征了一种实体的类型,称之为实体型。同型的实体的集合称为实体集。

信息经过加工编码进入机器世界,机器世界的对象是数据,相应于每一实体的数据为

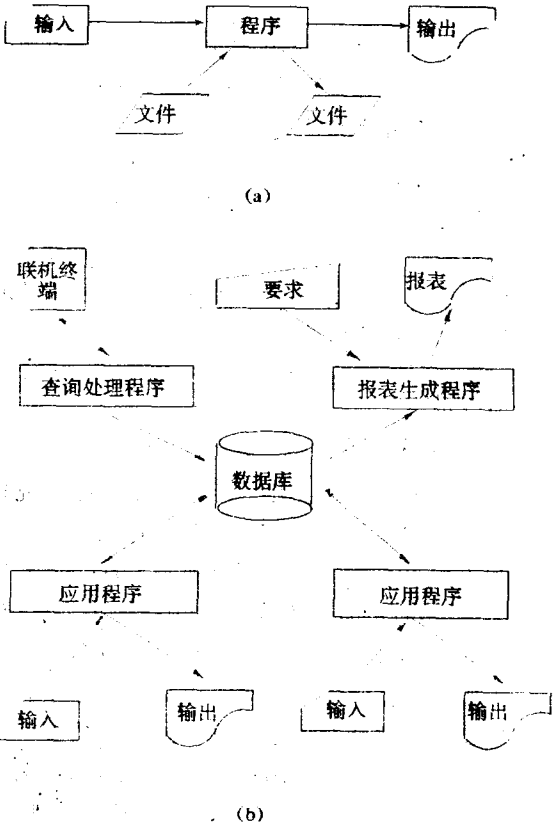


图 1.1 信息处理方式的演变
(a) 传统方式; (b) 数据库方式

记录, 相应于属性的数据为数据项或字段。从而相应于实体型的为记录型, 相应于实体集的为文件。

§ 1.2 数据库管理系统 (DBMS)

数据库系统是一个复杂的系统, 通常是由硬件、软件、数据库和数据库管理员四部分组成。

数据库系统对硬件的主要要求如下: 需要足够大的内存以存放操作系统、DBMS、应用程序、系统缓冲区、数据库等; 需要大容量的供直接存取的外存及较高的通道能力。

数据库系统软件包括: 操作系统, 主语言(如某一种高级语言), DBMS, 应用程序。其中核心部分是 DBMS, 它包括:

- (1) 语言: 模式、子模式、物理数据描述语言, 数据操纵语言 (DML), 查询语言等;
- (2) 实用程序: 装配程序, 重组程序, 日志程序, 恢复程序, 统计分析程序等;
- (3) 数据库运行控制程序: 数据库管理程序, 并发控制程序, 数据存取程序, 有效性检验程序, 合法性检验程序, 存取控制程序, 数据更新程序, 完整性、安全性保护程序。

以上是一般 DBMS 所大致包括的内容, 至于一个具体的 DBMS 到底包括哪些, 这并不受上述列举内容的限制。例如关系模型就没有数据的物理描述语言, 而有些层次、网络模型的 DBMS 没有查询语言。因此一个具体的 DBMS 完全由其设计者据他们的条件和要求来确定它的处理策略和内容。

数据库管理员 (Data Base Administrator, 简称 DBA) 是一个 (或一组) 负责整个 DBS 的建立、维护、协调工作的专门人员。他们对于程序语言和系统软件如 DBMS、操作系统等都要熟悉, 还要熟悉该部门的所有业务工作。

一、数据库管理系统的抽象层次

数据库管理系统 (Data Base Management System 简称 DBMS) 的主要功能是允许用户逻辑地、抽象地处理数据, 而不必涉及这些数据在计算机中是怎样存放的。在文件系统中, 用户对于他所存文件的物理组织、存贮细节等都要自行处理, 这给用户带来很大不便。数据库系统的一个目标就是解决这个问题。它把一切琐碎事务都交给 DBMS 来处理。

在数据库系统中, 用户看到的数据与计算机中存放的数据是两回事, 当然这中间是有联系的。实际上它们之间已经过了两次变换。一次是系统为了减少冗余, 实现数据共享, 把所有用户的数据进行综合, 抽象成一个统一的数据视图。第二次是为了提高存取效率, 改善性能, 把全局视图的数据按照物理组织的最优形式来存放。当计算机向用户提供数据时, 则做相反的变换。

用户使用的数据视图叫外模型。外模型是一种局部的逻辑数据视图, 它表示了用户所理解的实体、实体属性和实体关系。全局的逻辑数据视图叫概念模型, 又叫数据模型。它是数据库管理员所看到的实体、实体属性和实体间的联系。物理数据存贮的模型叫内模型。

整个系统分为三层: 外层, 概念层, 内层。用户只看到外层, 其他两层他们是看不到的。外模型有多个, 而概念模型和内模型只有一个。内模型是整个数据库的最低层。

用数据描述语言精确地定义数据模型的程序称为模式。对应于不同的模型, 也有三个

相应的模式：外模式或子模式，概念模式或模式，内模式或物理模式。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 表示了数据库系统的结构，由此可看出各级模式的关系。

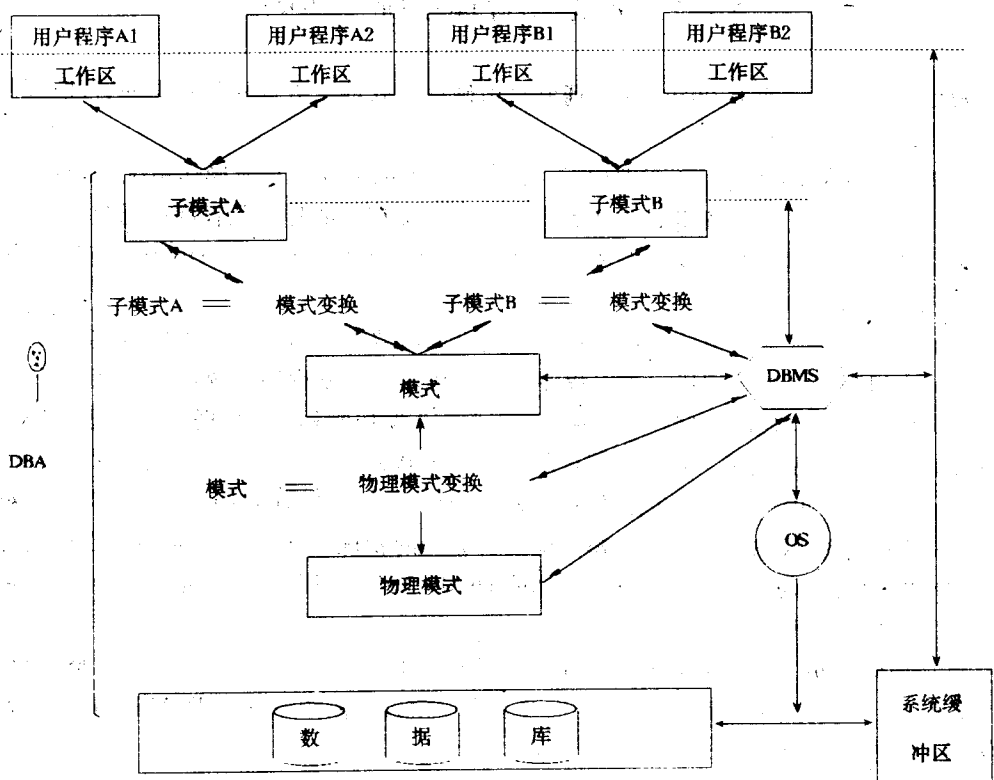


图 1.2 数据库系统结构

图 1.2 有二层变换（或映射）：

(1) 子模式 $\rightleftharpoons$ 模式变换。它定义了各子模式与模式之间的映射关系。当整个系统要求改变模式时，可以改变映射关系而保持子模式不变。这种用户数据独立于全局逻辑数据的特性叫逻辑数据独立性。

(2) 模式 $\rightleftharpoons$ 物理模式变换。它定义了模式与物理模式的映射关系。当为了某种需要，例如提高某文件的存取效率时，可以改变物理模式，同时改变模式 $\rightleftharpoons$ 物理模式映射而保持模式和子模式不变。这种全局的逻辑数据独立于物理数据的特性叫物理数据独立性。

这二种数据独立性统称数据独立性。由于有了数据独立性，数据库系统就把用户数据和物理数据完全分开，使得用户摆脱了繁琐的物理存贮细节，由于用户程序不依赖物理数据，这减少了应用程序的维护开支。

无论那一级模式都只能是处理数据的一个框架，而按这些框架填入的数据才是数据库的内容。但要注意的是框架和数据是两回事，它们放在不同的地方。所以模型、模式、具体值是三个不同的概念。具体值也包括三个层次：用户数据库、概念数据库、物理数据库，它们分别对应于以子模式、模式、物理模式为框架的数据库。用户数据库是概念数据库的部分抽取；概念数据库是用户数据库的汇集和综合，又是物理数据库的抽象；物理数据库

是概念数据库的具体实现。在这三层数据库中,仅物理数据库才真正存在,但 DBMS 能够从相应的物理数据库构造出概念数据库和用户数据库。因此用户可以认为它们是存在的。

在设计数据库时,主要考虑的是数据库的根系统中,而使用数据库时用户最关心的是数据库的内容。数据库的框架是相对稳定的,而数据库的内容是随时变化的。

二、数据库系统的工作过程

为了深入了解数据库系统的工作过程,以检索作为例子,解剖一下数据库系统是怎样进行工作的。

(1) 应用程序在运行时的某时刻,要求从数据库系统中检索一个用户数据记录,通常是一条 DML 语句,或者一条查询语言的命令,不管是哪种情况,都要指出何种操作、操作对象、输出要求、满足的条件等信息。此时 CPU 的运行权由应用程序(或 DBMS 的总控程序)转给 DBMS,而应用程序停止在此点。DBMS 分析命令及其各项参数,检查是否合法,以决定后续操作。若合法,则调进子模式。

(2) 进行存取权限检查,符合后继续下步。

(3) 调进模式,检查模式与子模式变换关系,确定所需子模式记录所在的模式关系。

(4) 查看物理模式及模式到物理模式的变换,确定应存取的物理记录及其存取方法。

(5) DBMS 向 OS 发出读进所需物理记录的命令。

(6) OS 执行输入/输出(I/O)命令,把数据从外存读到 I/O 缓冲区,再从 I/O 缓冲区传到 DBMS 的系统缓冲区。

(7) DBMS 比较模式和子模式,在系统缓冲区中组织成用户所需的记录形式。

(8) DBMS 把用户记录送到用户工作区。

(9) DBMS 将命令执行的结果状态信息送用户程序。

如果要进行其他操作,其过程与读一个记录是类似的。若是修改,首先把要改的记录读到用户工作区,由用户程序在工作区中进行修改,然后向 DBMS 发出写记录的命令,DBMS 在系统缓冲区进行必要的转换——与读记录时的转换相反,最后 DBMS 向 OS 发写命令。插入和删除都是先找到所在的位置或记录后,再执行。

§ 1.3 数据模型

一、实体间的联系

现实世界中的事物是彼此关联的,任何一个实体都不是孤立存在的,因此描述实体的数据也是互相联系的。联系有两种,一种是实体内部的联系,反映在数据上是记录内部即字段间的联系;另一种是实体与实体之间的联系,反映在数据上就是记录之间的联系。在文件系统中只考虑记录内部的联系,而不考虑文件与文件之间的联系,因而从整体上看数据是无结构的。这就是文件系统简单的原因,也是文件系统存在各种弊病的原因。数据库系统中除了考虑记录内部的联系外,还必须考虑记录之间的联系,也就是文件之间的联系。这种联系比较复杂,这也就是数据库系统复杂的原因。而这些都是由于实体间关系的复杂性引起的。

实体间的关系虽然复杂,但抽象化以后,可把它们归结为三类:

1. 1-1 (one-to-one) 关系

如果两个实体集 E_1 、 E_2 中的每一个实体至多和另一个实体集中的一个实体有联系, 则 E_1 、 E_2 叫“一对一关系”, 记为“1-1 关系”。

这是最简单的实体联系。例如, 一个公司只有一个经理, 同时一个经理只能在一个公司任职, 所以实体集公司和经理是 1-1 关系。再如学生班与班长, 实数轴上点和实数集等都是 1-1 关系。但必须注意, 1-1 关系不一定都是一一对应的关系。

2. 1-m (one-to-many) 关系

有两个实体集 E_1 和 E_2 , 如果 E_2 中每个实体与 E_1 中任意个实体(包括零个)有关, 而 E_1 中每个实体至多和 E_2 中一个实体有关, 则称该关系为从 E_2 到 E_1 的“一对多关系”, 记为“1-m 关系”。

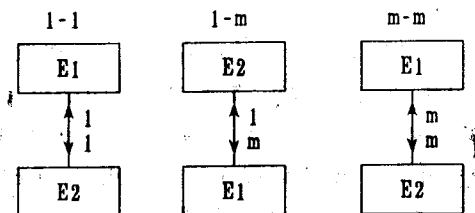


图 1.3 实体关系的图形表示法

1-m 关系是比较普遍的关系。例

如, 一个公司有多名职员, 而一个职员最多在一个公司工作, 所以公司与职员是一对多关系。此外, 经理与职员, 班长与同学, 教练与运动员及母子关系、院系关系等都是 1-m 关系。

3. m-m (many-to-many) 关系

如果两个实体集 E_1 、 E_2 中的每一个实体都和另一个实体集中任意个实体(包括零个实体)有关, 则称这两个实体集是“多对多关系”, 记为“m-m 关系”。

注意, 1-1 关系是 1-m 关系的特例。而 1-m 关系又是 m-m 关系的特例。它们之间的关系是包含关系。有时也可用图形来表示实体间的各种关系, 如图 1.3 所示。

二、数据模型概述

数据库系统的一个核心问题就是研究如何表示和处理实体间的联系。我们把表示实体及实体之间联系的模型叫数据模型。数据模型是表示现实世界的一种方法与工具。数据模型是数据库系统中的一个关键概念, 它是实体间联系的一个轮廓视图。数据模型的不同将决定不同的 DBMS。

本世纪六十年代末七十年代初在文件系统后相继出现了层次数据模型、网状数据模型和关系数据模型, 即现在常称的三大经典数据模型。

上述三大经典数据模型是目前应用最广泛的模型, 现行大部分实用 DBMS 均是在它们基础上建立的。三大经典模型的主要特点是基本上面向机器的, 即它们使得能有效地存贮和处理数据。但其追求的目标不是数据对象本身的内涵, 它们有限的表达能力也远不能自然地按直接方式来描述和模拟现实世界中的复杂应用, 因而它们基本上不是面向用户的。

由于经典数据模型对诸如决策支持系统 (DSS) 和计算机辅助设计/制造 (CAD/CAM) 系统这样一类复杂应用描述和建模能力的不足, 因而随着面向对象的程序设计语言和人工智能的发展, 在三大经典数据模型之上产生了一批语义数据模型, 如实体-联系模型, 函数数据模型, 面向对象的语义联系模型等等。这些语义数据模型的共同特点是尽可能表达数据对象本身的内涵及其之间的联系, 而不是象经典模型那样着重于具体描述数据的组织结构。由于舍弃了低层的信息而着重于更接近用户的高层的数据说明, 因此语义数据模型

对于现实世界具有更直接的表达能力,基于这些模型的数据库系统也更便于操作。

需要指出的是,尽管目前已出现了不少面向对象的语义数据模型,但三大经典数据模型仍是当今应用的主流。网状数据库,特别是关系数据库在实用系统中还处于方兴未艾的状态。其实,目前已实现的一些语义模型的原型系统,其低层也仍然是基于经典模型,它们只不过是在其上层加了一个语义模型而已。所以,三大经典数据模型在现实世界中仍然具有极其重要的尚不可取代的地位。

三、层次模型

用树型结构或森林来表示实体与实体间联系的模型称层次模型。实体用树中的结点表示,实体间的联系(仅限于1-1,1-m)用树中的连线表示。较严格地说,层次模型指满足下列条件的基本层次联系的集合:①每棵树有且仅有一个节点无父节点,此节点即树根;②除根以外的每个节点有且仅有一个父节点。在现实世界中有许多实体之间的联系就是一种很自然的层次关系,例如行政机构、家族关系等。图1.4给出了一个层次模型的示例,其中工厂实体为树根,各层父子节点之间均为一对多的关系,即一个工厂可有多个分厂,一个分厂可有多个车间,一个车间可有多个班组等。对于层次模型还需指出的是:

(1)层次模型中的树为有序树。所谓有序树是对树中任一节点的所有子树规定了先后次序的树。对层次模型来说,都按从左至右的顺序规定了任一节点所有子树的先后次序。这一限制隐含了对层次型数据库存取路径的一种控制。

(2)树中实体间的联系关系是单向的。即由父节点指向子节点,而且在一对父子节点间不存在多于一种的联系关系。这一约定限制了单一层次结构对两个结构间可能存在的多种联系关系的建模,即对树中的任一节点,只有唯一的一条自根节点到达它的路径。

(3)层次模型中的联系关系只能是父节点对子节点的一对多关系,这一规定限制了层次模型对多对多联系关系的直接表达,必须设法先将该关系分解为两个1-m关系。

(4)树节点中任何记录的任何属性均为不可再分的简单型数据,即具有原子性。

总之,树形结构的特点限定了层次模型的层次性、有序性、父子节点间的一对多联系以及自顶向下、从左至右的检索路径。层次模型具有一定的局限性,但它具有简单、清晰等优点。

层次模型的数据库管理系统是最早出现的。它的典型代表是IBM公司的IMS系统,于1968年问世,是世界上第一个DBMS,它集中体现了层次模型的特点。

IMS (Information Management System)有好几个版本,1968年第一版IMS-1,1971年第二版IMS-2,1974年IMS/VS。早期的IMS是宿主语言系统,它所使用的宿主语言有PL/1, COBOL, 汇编语言等,后来也发展了自含的查询语言。数据操作语言DL/1是以调用方式嵌入宿主语言中的。IMS对数据逻辑结构、物理存储组织和工作环境的描述叫数据描述DBD,它相当于模式加上一部分映射定义。对用户使用的那部分数据的逻辑结构描述称为程序说明块PSB,它是一组程序通讯块PCB的集合,PCB相当于子模式加上有关的子模式到模式的映射。

四、网状模型

如果取消层次模型中的二个限制,即每一个结点可以有多个父结点,便形成了网络,又称为丛。我们把用丛结构来表示实体之间联系的模型叫网络模型。由于网络模型没有层次

模型的限制,所以用它可以直接表示 $m-m$ 关系。当然在一些已经实现的网络模型的 DBMS 中,对这一点做了限制,例如 DBTG 系统只允许处理 $1-m$ 关系。显然,层次是网络的特殊形式,网络是层次的一般形式。网络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连线表示实体间的联系,用结点表示实体集(记录类);从物理上看,层次模型和网络模型都是用指针来实现两个文件之间的联系。基差别在于网络模型中的连线或指针更加复杂,更加纵横交错,从而数据结构就更复杂。

在丛结构中,每一结点都可以和其他任一结点相连,图 1.5 是一些丛结构的例子。由图 1.5 我们可以看到,与树结构一样,丛结构也可以用父结点和子结点来描述,并且同样把父结点安排在子结点的上方。

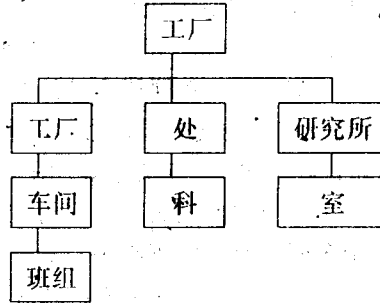


图 1.4 层次模型示例

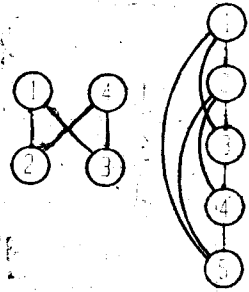


图 1.5 网络结构的例子

与层次模型相比,网状模型具有较强的数据建模能力;与关系模型相比,仍然缺乏形式化基础和操作的代数性质。另外在网状系统中实体间的联系是用复杂的指针链来实现的(所谓“导航链”),这样其数据操纵语言的非过程化程度低,比关系模型更面向机器,但正是具有“导航链”更面向机器这一点给网状 DBMS 带来了存取率高的优点。在某些应用领域,如 CAD/CAM 图形数据库系统中,由于网状 DBMS 的高效,更由于网状数据模型提供了描述三维图形信息的更为自然的结构形式,是支持实现图形边界表示法(B-rep)的有力工具,因此得到了广泛的采用,并仍有其较强大的生命力。

网状数据模型中最有代表性的是美国 CODASYL(Conference of Data System Language)组织的下属机构数据库任务组(DBTG)于 1969 年提出的 DBTG 报告。在 DBTG 报告中首次提出了网状数据模型中的一些最基本的概念,目前运行的大部分网状数据库管理系统都是在不同程度上实现了 DBTG 报告或其后的修改文体。需要指明的是,由于有后面所述系构成规则的限制,所以 DBTG 模型是一种简单型网状模型。此外还有复杂型和受限型网状模型。

DBTG 报告中除了有定义概念模式的模式数据描述语言(DDL)外,还有子模式数据描述语言(SDDL)以及数据操纵语言 DML。DBTG 网状数据模型有两种基本构件,即记录类型与系类型。以它们为基础,按照一定的规则可构出网状数据模型以描述现实世界中实体与实体间的联系。

五、关系模型

(一) 概述

表格是大家所熟悉的数据表示方法。例如学生成绩管理中经常用到下面 3 个表格(为叙述简便,其中许多内容均用西文或代号表示)。