

高等学校试用教材

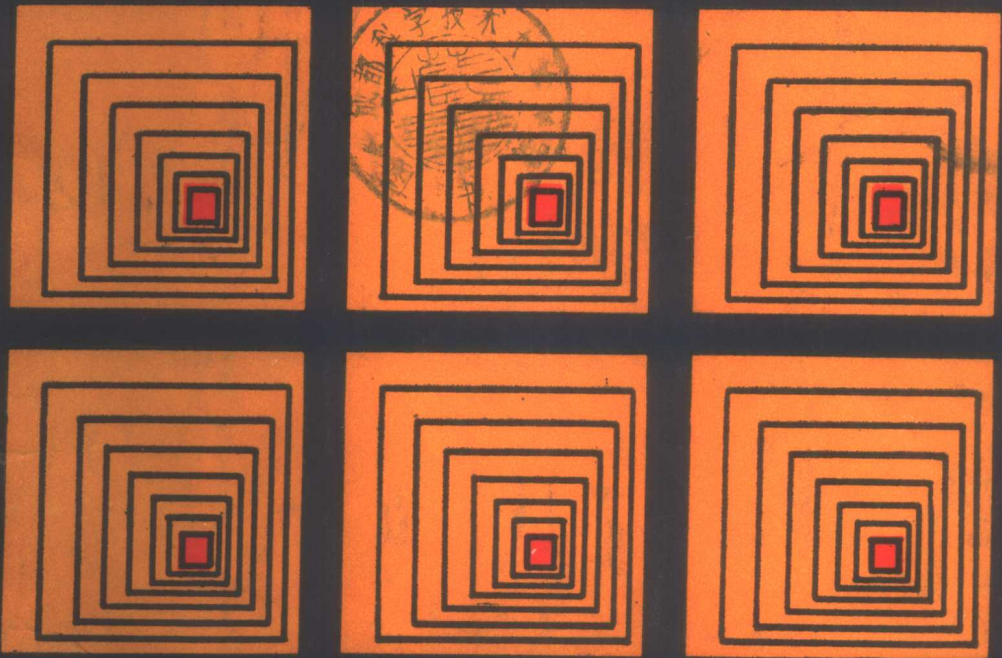
数据库原理与应用

史济民 主编

5087

5087; 1

技术大学图书馆
基本藏书



高等教育出版社

高等学校试用教材

数据库原理与应用

史济民主编

高等教育出版社

内 容 简 介

本书是集原理与应用于一书的数据库教材,含绪论、应用篇、原理篇三部份。应用篇共12章,从dBASE(包括Ⅰ、Ⅱ版)数据库管理系统入手,评述了该系统建库、简单与复杂查询、输出报表、编写程序以及汉字处理、与其它语言的连接等功能与应用,最后以一整章的篇幅介绍了一个工厂信息管理系统的开发全过程和样例程序。原理篇分6章介绍了关系数据库原理,包括直接用于设计应用系统的关系规范化、实体-联系方法等指导原则与方法,以及结合dBASE System R等实际系统讲述的关系查询语言、存储结构等主要基础知识。

本书立足于应用,又用理论来指导应用。全书概念清楚,叙述简练,有例题200余个,每章均附习题。可作为高等学校非计算机专业学生或计算机应用专业大专生(含短训班)的教材,也可供计算机专业的学生或有关科技人员阅读参考。

高等学校试用教材

数据库原理与应用

史济民 主编

高等教育出版社出版

新华书店重庆发行所发行

重庆印制一厂印装

开本787×1092 1/16 印张18.75 字数430 000

1987年10月第1版 1987年10月第1次印刷

印数 00 661—7 280

ISBN 7-04-000223-X/O·266

书号 13010·01490

定价 3.20 元

前 言

本书系根据编者在大学生中讲授《数据库应用》课的讲义修订而成，可用作高等学校非计算机专业本科生或计算机应用专业大专生（含短训班）的教材，也可供计算机专业本科生和从事计算机应用的科技人员参考。

数据库是一门实践性很强的课程。非计算机专业的学生学习数据库，主要是为了应用。但数据库技术又综合了多种软件和硬件知识，不掌握基本的原理，就很难用好和设计好复杂的应用系统。随着数据库应用的逐渐普及，越来越多的高等学校已在非计算机专业中开设了数据库课，但适用的教材却不多见。本书从应用出发，集应用与原理于一书。在上篇（应用篇）详细讲解微型机dBASE数据库管理系统的基础上，于下篇（原理篇）有重点地介绍了关系数据库系统的基本原理。两篇相互联系，又各有侧重，力求理论结合实际，用理论来指导应用。

在结构安排与内容选择上，本书采取了：

一、把应用篇置于原理篇之前，让初学者首先接触一种实际的数据库管理系统，学了就能应用；然后学一点原理，使之“更上一层楼”。实践证明，这种安排符合从特殊到一般、由感性到理性的认识规律，学生很容易接受。

二、应用篇选择当今微型机上流行最广的dBASE数据库管理系统，详细讲解了比较成熟的、已有汉字化版本的dBASE-Ⅲ单用户系统。但为了兼顾8位微机的读者，在讲完每条（组）dBASE-Ⅲ的命令后，随即在“dBASE-Ⅱ的情况”标题下说明dBASE-Ⅱ与dBASE-Ⅲ的差别，且在课文中用较小的字体印刷，以便区分。这部分内容约占全书5%，增加了少量篇幅，却同时满足了8位微机 and 16位微机两类用户，可以供读者按需选用。

三、原理篇破除了多数原理书并列介绍层次型、网络型和关系型三类数据库的传统作法，只讲被称为“数据库发展方向”的关系型数据库的原理。在其中，又突出了“数据库应用系统的设计”（第五章），以及构成数据库重要基础的“数据操作语言”（第二章）和“存储结构”（第三章）等知识。有取有舍，主次分明。

编写本书时，编者力求克服语言书的繁琐和原理书与实际脱节等弊病，坚持实用、易懂，讲清基础理论。全书共收集各类例题共200余个，并在应用篇第十二章全面讨论了一个工厂信息管理系统的详细设计过程，供读者参考。

在通常情况下，本书宜放在学过一门高级语言之后学习，但这并非必要条件。非计算机专业对数据库应用的要求不尽相同，各校能提供的课时数也各有差异，为了适应不同对象的需要，在原理篇，有部分内容的标题前加了“*”号，供教师和读者选用。如果学习全部内容，课堂学时约需要45学时左右（其中绪论2学时，应用篇约23学时，原理篇约20学时）。若课时较少，可选讲绪论、上篇、和下篇第一、四、五章，约需32学时。有些章如上篇第九至十一章，下篇第六章也可以安排学生自学。

本书由成都科技大学计算机系史济民主编。上篇第四、五、六、八、十一章由成都科技

大学彭亲容执笔，上篇第九、十章由四川广播电视大学吴发碧执笔，史济民编写了绪论，上篇第一、二、三、七、十二章与下篇全部6章，并负责全书的内容编排与修改定稿。由于同类教材尚不多见，选材容有不当，加上编者水平有限，难免有缺点错误，诚恳希望读者批评指正。

清华大学董长德副教授和中国人民大学刘惠芳副教授详细审阅了书稿，提出了许多宝贵意见，编者谨对他们表示衷心的感谢。

在本书出版前，已有北京航空学院、内蒙古大学、华东化工学院、大连海军水面舰艇学院、北京自动化学院等兄弟院校试用过本书的讲义，有些老师对这次修订提出了有益的建议。全国高等学校计算机基础教育研究会会长许镇宇教授热情鼓励我们作好修订工作，并提出了指导性的意见。成都科技大学计算机系和四川广播电视大学分别从各方面给予了支持。科大软件教研室机房的高仕福、陈跃辉、贺之勤等同志作了许多具体工作。对这些单位与同志，编者一并表示诚挚的谢意。

编者

1986年12月于成都

目 录

绪论	1	1 组成成分	3
0.1 数据管理技术的发展	1	2 SPARC分级结构	4
1 人工管理阶段	1	0.4 数据库管理系统	5
2 文卷管理系统阶段	1	1 数据描述语言	5
3 数据库系统阶段	2	2 数据操作语言	6
0.2 数据库系统的特点	2	3 管理与控制例行程序	6
1 数据共享	2	0.5 回顾与展望	6
2 数据的结构化	3	1 从格式化模型到关系模型	6
3 数据的独立性	3	2 从集中式数据库到分布式数据库	7
4 可控冗余度	3	习题	8
0.3 数据库系统的组成与结构	3		

上篇—应用篇 dBASE数据库管理系统

第一章 概述	10	卷	28
1.1 dBASE要求的运行环境	10	1 M型字段的定义	28
1.2 dBASE 语言的成份和主要性能指标	10	2 数据的装入	29
1 命令	11	3 数据的显示	30
2 程序	12	2.4 用Copy命令复制数据库	30
3 文卷	13	1 常用可选项的含义	31
4 变量	14	2 原样复制	31
5 表达式	16	3 非原样复制	32
6 函数	16	2.5 用结构文卷建立新库	34
1.3 dBASE的主要优缺点	19	1 建立结构文卷	34
1.4 启动和退出dBASE	20	2 用结构文卷定义数据库	35
1.5 编译型dBASE-Ⅲ与多用户dBASE-Ⅲ	21	2.6 修改数据库的结构	36
习题	21	1 Modify structure命令	36
第二章 建立数据库	22	2 修改结构的步骤	37
2.1 全屏幕编辑	22	3 Append from 命令	38
2.2 定义新的数据库	23	习题	38
1 Create 命令	23	第三章 数据的修改、检索和统计	40
2 Append命令与Use命令	26	3.1 显示命令	40
3 List命令	27	1 Display命令	40
2.3 记事型字段与数据库文本		2 ? 命令	42
		3.2 删除和插入数据记录	43
		1 Delete、Recall和Pack 命令	43

2	Insert 命令	43
3	Goto命令和Skip命令	43
3.3	修改数据记录	47
1	Edit命令	47
2	Change命令	48
3	Browse命令	48
4	Replace命令	50
3.4	检索命令	53
1	Locate命令	53
2	Continue命令	53
3.5	索引文卷	55
1	索引的建立	55
2	索引的打开	56
3	索引的更新	57
3.6	索引检索命令	59
1	索引检索的优点	59
2	Find命令	59
3	Seek命令	62
3.7	排序命令	63
1	Sort命令	63
2	与索引方法的比较	65
3.8	数据的统计	65
1	Count命令	66
2	Sum命令	66
3	Average命令	67
4	Total命令	67
	习题	69
第四章	报表输出	70
4.1	建立报表格式文卷	70
1	Create report命令	70
2	定义格式文卷的步骤	70
4.2	报表输出命令	75
4.3	修改报表格式文卷	77
1	调整报表宽度	78
2	在菜单工作方式下修改报表	79
4.4	打印标签	81
1	建立标签格式文卷	81
2	标签打印命令	85
	习题	84
第五章	多工作区操作	85
5.1	多区操作的特点	85

1	工作区的性质	85
2	工作区的选择	86
3	文卷的别名与作用	86
5.2	Replace命令的多区操作	87
5.3	两个数据库间的关联	88
5.4	两个索引库间的更新	91
5.5	两个数据库间的连接	94
	习题	96
第六章	建立、编写和调试程序文卷	97
6.1	程序文卷的建立、修改与执行	97
1	建立和修改程序	97
2	执行程序	98
3	过程的建立与执行	98
6.2	输入命令	99
1	Accept命令	99
2	Input命令	100
3	比较	100
6.3	程序的选择结构	101
6.4	程序的重复结构	104
6.5	程序的注释	106
6.6	一个实例	107
6.7	程序的调试	112
	习题	113
第七章	把你的程序编得更好	115
7.1	格式设计命令	115
1	作用	115
2	一般形式	115
3	几种常见的形式和用法	115
7.2	自行设计的格式输入	118
1	Read命令	118
2	Clear gets命令	120
3	版式文卷	120
7.3	自行设计的格式输出——表格型报表的设计	123
7.4	内存变量和内存变量文卷	127
1	定义和释放变量	127
2	内存变量文卷	128
3	公用和专用变量	129

7.5 过程文卷	131
1 过程文卷的作用	131
2 建立、打开和关闭过程文卷的命令	131
7.6 过程的带参数调用	132
7.7 在程序运行中输出文本	134
习题	134
第八章 选择系统工作状态	136
8.1 Set-on命令	136
8.2 Set-to命令	139
8.3 Set命令	142
习题	144
第九章 辅助功能	145
9.1 dBASE-III的菜单工作方式	145
9.2 单项help命令	145
9.3 Help命令	146
9.4 Assist命令	148
习题	149
第十章 与其它高级语言的连接	150
10.1 文本文卷与数据文卷的转换	150
1 文本文卷的格式	150
2 从DBF到TXT	151
3 从TXT到DBF	152
10.2 dBASE与BASIC的联合运行	156
1 一个简单的例子	156

2 用批命令实现BASIC与dBASE的联合运行	159
10.3 dBASE与编译型语言的联合运行	162
1 RUN命令	162
2 含有RUN命令的dBASE程序	162
习题	163

第十一章 汉字dBASE与汉字操作系统	164
11.1 汉字操作系统CC-DOS及其汉字输入/输出功能	164
1 汉字输入功能	164
2 汉字输出功能	166
11.2 启动和退出C-dBASE	167
11.3 汉字数据库的建库、操作与检索	167
1 汉字数据库的建立	168
2 汉字数据库的操作	168
3 汉字数据库的检索	169
11.4 应用举例	170
习题	175
第十二章 典型样例分析	176
12.1 建立应用系统的数据库	176
12.2 应用程序的总体设计	181
12.3 用dBASE编写应用程序	182
习题	203
附录 dBASE命令分类表	204

下篇—原理篇 关系数据库原理

第一章 基本概念	209
1.1 数学定义	209
1 域	209
2 笛卡尔乘积	209
3 关系	209
1.2 关系的性质	210
1.3 关系数据库的描述	211
1 域的描述	211
2 关系的描述	211
3 DDL语言和模式描述方式	212

关系子模式	212
1.4 关系数据库上的操作	213
1 操作的种类与特点	213
2 DML语言及其分类	213
习题	213
第二章 数据操作语言	215
2.1 关系代数语言	215
1 传统的集合运算	215
2 专门的关系运算	216
3 ISBL语言	219

4 小结	220	4.1 关系中的键	251
2.2 关系演算语言	221	4.2 函数依赖	252
*1 ALPHA语言与QUEL语言	221	4.3 规范化和范式	254
2 QBE语言	223	4.4 第一、第二和第三范式	255
3 小结	225	*4.5 BC范式	258
2.3 SQL语言	225	4.6 关系模式的分解	259
1 映射运算与映射块	225	习题	261
2 用映射块进行检索	225	第五章 数据库应用系统的设计	262
3 SQL的差、并、交运算	228	5.1 设计过程概述	262
4 介于关系代数和关系演算之间的语言	228	5.2 应用系统的要求分析	263
5 嵌入型SQL	229	1 分析用户活动	264
2.4 dBASE语言	230	2 确定系统边界	264
1 dBASE支持的关系运算	230	3 分析系统数据	264
2 dBASE的关系完备性	230	5.3 实体-联系方法	265
习题	282	1 基本思想和作用	265
第三章 存储结构	234	2 E-R模型的基本成份	265
3.1 磁盘和主存间的数据交换	234	3 E-R图	266
1 磁盘信息的存取	234	5.4 概念模式的设计	267
2 内、外存间的信息交换	235	1 建立局部概念模式	267
3 数据库文卷的管理	235	2 建立总体概念模式	267
3.2 数据库的基本文卷	236	5.5 三类数据模型	269
3.3 数据库主文卷的存储	237	1 层次模型	269
1 dBASE的主文卷	237	2 网状模型	270
*2 System R的主文卷	238	3 关系模型	270
3.4 稠密索引、稀疏索引和多级索引	239	4 一个简单的例子	271
1 稠密索引	239	5.6 逻辑模式设计	272
2 稀疏索引	240	1 E-R图向关系模型的转换	272
3 多级索引	241	2 不同情况处理示例	273
3.5 二叉树索引	242	5.7 性能预测和优化	276
1 树和二叉树	242	1 LRA方法	276
2 二叉树索引	242	2 模式的优化	278
*3.6 B树和B ⁺ 树	243	5.8 系统的运行	280
1 B树	243	*5.9 数据库工程与软件工程	281
2 B ⁺ 树	246	习题	281
3.7 索引文卷的存储	247	第六章 安全性与完整性	282
1 dBASE的索引文卷	247	6.1 安全性	282
*2 System R的索引文卷	248	1 用户鉴别	282
习题	250	2 权限控制	283
第四章 关系规范化	251	3 实际系统的安全性保护	283
		6.2 完整性	284

1 约束条件	284
2 实际系统的完整性保护	285
6.3 对数据保护的简单讨论	287

习题	288
主要参考文献	289

绪 论

数据库是计算机软件的重要分支,它体现了当代先进的数据管理技术.1968年,美国IBM公司制成了第一个商品化数据库系统——IMS系统.从那时以来,数据库技术迅速发展,赢得了社会的广泛承认与应用.80年代初,一些微型机配上了数据库语言,进一步加快了数据库技术的应用与普及.今天,学习与应用数据库,已成为广大管理和科技人员的共同需要.

0.1 数据管理技术的发展

计算机的出现,开辟了电子数据处理(EDP)的新纪元.面临着交给它处理的大量数据,计算机怎样去管理它们,怎样对它们进行组织、存储、检索与维护,是提高数据处理效率的中心问题.30余年来,随着计算机数据处理的发展,数据管理技术先后经历了三个不同的发展阶段.

1. 人工管理阶段

大约在50年代中期以前.当时的计算机主要用于科学计算,没有操作系统,没有今天常用的可以直接存取的磁盘,外存仅有磁带和卡片(或纸带).因此,数据主要靠人工管理,一切工作都要程序员来完成.这阶段的主要特点是:

(1) 没有专门的数据管理软件.在程序设计中,程序员既要规定数据的逻辑结构,还要花费大量精力为数据分配存储空间,决定存储结构与存取方法.从另一方面看,程序对数据存在严重依赖.数据在存储介质上的任何改变,都将引起程序的修改.

(2) 从客观上来说,科学计算的数据一般不需要长期保存.计算结束,数据就不再有存储介质上保留.所以这个时期基本上不使用文卷(file)^①,即使偶尔使用也大都 是 顺 序 文 卷.

(3) 数据是面向程序的.每个程序都使用自己的数据.如果几个程序都用到同一部分数据,这些数据就只好重复存储,所以数据的冗余度大.

2. 文卷管理系统阶段

指50年代末期至60年代中期.由于数据处理的应用日益增多,出现了专用于管理数据的软件,称为文卷管理系统.外存也有了磁盘、磁鼓等直接存取设备.这个时期的特点是:

(1) 文卷管理系统的使用,明显简化了程序员的数据管理工作.在程序设计中,程序员一般只考虑数据的逻辑结构.从逻辑结构到存储结构的转换,主要由文卷管理系统去执

^① file旧译文件,现据《数据处理词汇》国家标准(报批稿)改译为文卷,下同.

行。

(2) 在数据处理中,常需把数据留在外存介质上反复处理,因而促进了文卷处理技术(包括文卷的组织、存取、管理等)的发展。在文卷组织形式上,除顺序文卷外,出现了直接存取文卷、索引文卷、倒排文卷等新的文卷结构。不论哪一种文卷结构,数据的存取均以记录为单位。

(3) 虽然在上述方面比人工管理阶段有很大改进,但数据仍然是面向程序的。程序和数据互相依赖,一方的变化势必引起另一方的相应变化,否则就会不相适应。加上数据文卷都是用户专有的,处于分散的状态,数据冗余依然存在。

文卷系统的这些缺点,促使人们去寻求新的数据管理技术。到60年代末,终于出现了数据库系统。

3. 数据库系统阶段

如前所述,数据库系统体现了数据管理技术的新发展。它是在数据量急剧增长,管理规模日益庞大,用户迫切要求共享数据的背景下发展起来的。磁盘技术的改进和容量的增大,为数据库提供了重要的物质条件。

从文卷系统到数据库系统,标志着数据管理技术的跃进。它主要表现在:

(1) 对数据实行统一的、独立的管理,把数据管理从程序设计中分离出来,大大减少了程序员管理数据的劳动和数据与程序间的相互依赖。

(2) 改进了数据操作,尤其是数据检索技术。数据的存取可以整个文卷为单位,数据检索可深入到记录内部的数据项。

(3) 实现了数据共享,大大减少了数据的冗余。实践证明,对于比较复杂的数据处理系统,使用数据库比使用一般的文卷系统更加方便,程序员花费的劳动往往会减小许多倍。这就是数据库受到广大用户包括微型机用户欢迎的原因。下节将比较详细地介绍数据库系统的这些特点。

0.2 数据库系统的特点

1. 数据共享

数据共享是数据库的目的,也是它的重要特点。一个数据库内的数据,不仅可以为同一企业或组织内部的各个部门所共享,也可以为不同组织、不同地区甚至不同国家的用户共享。各个用户可以在相同时间使用同一个数据库,每个用户使用其中的一部分数据,并允许他们访问的数据互相交叉和重叠。

数据共享提高了数据的利用率,又能节省存储空间,但是也带来一些新问题:

(1) 安全保护。即保证数据只被合法的用户使用,防止数据被窃用或破坏。

(2) 完整保护。指维护数据的正确性与相容性(即一致性),防止因数据出错造成损失。正确性系就数据的范围而言。例如学生分数只能取0—100间的整数,超出范围的数据就被认为无效。相容性系指这部分数据与另一部分数据之间应该满足的关系。例如当同一个人或事的信息在不同地方出现时,其内容不应该矛盾。

关于安全性和完整性的进一步讨论,请参阅下篇第六章。

(3) 并发控制。当多用户对数据库并行存取时,要防止数据发生错误和混乱。例如两个旅客同时预定一个航班的最后一个座位时,必须等一个订好后再接受下一个的订票,以免两个旅客预订同一个座位。

2. 数据的结构化

所谓结构化,是就数据整体而言的。在文卷系统中,文卷之间不存在联系,从整体看数据是没有结构的。而数据库中的文卷则相互联系,并在整体上服从一定的结构形式。数据库之所以具有复杂的结构,不仅是因为它们通常拥有大量的数据,同时也因为在数据之间和文卷之间存在着种种联系。一个“数量大”,一个“联系多”,决定了数据结构的复杂性。J. D. Ullman说过:“数据库使用的数据结构与传统程序设计使用的数据结构有些差别。它所包含的大量数据,使一些过去人们仅从理论上感到兴趣的结构,今天有了实际的用武之地。”事实上,数据库不仅使用了多种传统的数据结构,而且创造了新的结构形式(如B⁺树),丰富和发展了数据结构的内容。

3. 数据的独立性

从0.1节可知,无论是人工管理阶段或文卷系统阶段,都存在着应用程序对于数据的依赖性,例如当应用程序原来使用磁带顺序文卷时,要想改成磁盘索引文卷,就须对应用程序作相应的修改。数据库系统力求减小这一依赖性,以实现数据独立性为目标,并已在一些数据库系统中较好地实现。数据独立性可以分为两级:

(1) 物理独立性。指不让物理结构的改变影响到逻辑结构。所谓物理结构的改变,包括外存设备、存储结构、存取方法等多种更改。如果逻辑结构不受它们更改的影响,就不需要改变应用程序。

(2) 逻辑独立性。指不让逻辑结构的改变影响到应用程序。至今为止,这一独立性还不能彻底实现。例如若删除了某一数据类型,则用到这些数据的应用程序一定要修改。

4. 可控冗余度

这实际上是数据共享所带来的优点。在文卷系统中,数据是面向应用的。以教职工档案为例,人事部门可能需要姓名、年龄、性别、工资等级、文化程度、专长等数据,财务部门需要姓名、工资等级、基本工资、奖金、扣除费等数据,教学部门需要姓名、职称、授课名称等数据,由于每个部门都使用自己的文卷,故在这些文卷中姓名将重复三次,工资等级重复两次,存储时就会产生大量冗余数据。在数据库中,由于数据集中使用,从理论上说可以消除冗余,但实际上出于提高检索速度等考虑,常常允许部分冗余存在。但这种冗余是可由设计者控制的,故称为“可控冗余度”(Controlled redundancy)。

0.3 数据库系统的组成与结构

1. 组成成分

一个数据库系统一般应包含三个部分:

(1) 数据库文卷。一个数据库,实际上是一组相互有联系的文卷的集合。这些文卷包括数据文卷(常称为主文卷)和它们的索引(文卷)。为了提高检索的速度,数据库系统大量使用索引文卷。一个数据库至少有一个主文卷。一个主文卷允许有一个或多个索引文卷,也

可以不带索引文卷。

(2) 数据库管理系统。是一个帮助用户建立、使用和管理数据库的软件系统。为便于叙述,通常简称为DBMS (Data Base Management System)。一般地说,它应该具有数据库的定义(俗称“建库”)、数据的装入、数据操作(包括检索操作与存储操作)、数据控制(包括安全性、完整性和并发控制)和数据库维护(包括数据库的整理和重定义等)等功能。DBMS既是面向用户的,也是面向整个数据库系统的。

(3) 支持数据库的硬件和软件。一定的硬件和软件环境,是保证数据库系统顺利工作的必要条件。图0.1是带有数据库的计算机系统。这种系统通常要求足够大的内存,以便存放操作系统、DBMS和应用程序,还要求能进行直接存取的磁盘存储器,要求能对DBMS提供所需支持的操作系统。如果想要系统处理汉字,还必须配备具有汉字功能的软件和硬件,如汉卡、汉字库、中文打印机、中文操作系统等。

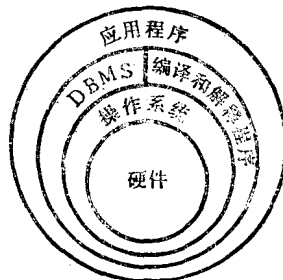


图0.1 DBMS环境结构

图0.1显示了数据库管理系统与其它语言处理程序(汇编、编译或解释程序)可以在同一系统中并存。用户可以按自己的需要,或者选用DBMS,或者选用其它语言处理程序。但是,任何应用程序如果要使用数据库,都必须通过DBMS。

综上所述,数据库系统是一个引入了数据库以后的计算机系统。它是数据库、DBMS和支持它们的计算机系统的总称。

2. SPARC分级结构

SPARC是美国国家标准学会(ANSI)所属标准计划和要求委员会(Standards Planning And Requirements Committee)的简称。1975年,该委员会公布了一个关于数据库标准化的报告,把数据库分为外模式、概念模式和内模式三级,即著名的SPARC分级结构(见图0.2)。不管实际的数据库系统有多大差异,它们的基本结构大体上是一致的,都可用这一分级结构来表示。现简述如下:

(1) 内模式(internal schema)。对应于物理级数据库,又称为存储模式。它包括存储在外存储器(通常为磁盘)上的所有数据库文卷,是一个实际存在的数据库。从系统程序员的角度看,内模式的文卷是用由他选择的文卷结构组织起来的,所以又称为系统程序员视图。

(2) 概念模式(conceptual schema)。对应于概念级数据库,它是数据库管理员所看到的系统的整体逻辑模式,所以又称为DBA视图。DBA为数据库管理员

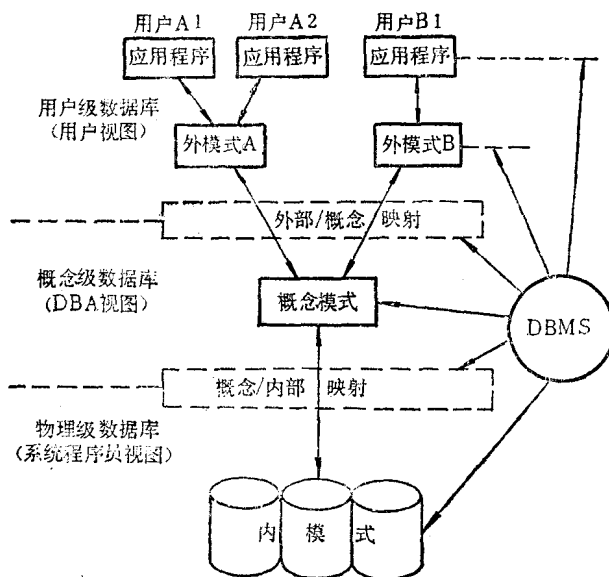


图0.2 SPARC分级结构

(Data Base Administrator) 的简称, 他是对数据库系统进行全面管理 (包括对数据库的定义、使用、维护和控制) 的负责人。

(3) 外模式(external schema). 对应于用户级数据库. 它是用户看到和使用的数据库, 因此也称为用户视图. 有些文献中称它为子模式, 因为它是概念模式的子集, 包含了概念模式中允许特定用户使用的那一部分数据。

一般地说, 不同的用户对数据有不同的需要. 对于某一特定的用户, 不必也不应该让他访问所有的数据. 通过外模式来定义允许每一用户使用的数据, 既可满足用户的使用要求, 又能起到保护数据安全的作用。

值得注意的是, 在上述三级模式中, 只有内模式是真正存储数据的, 概念模式与外模式均不存储数据. 换句话说, 磁盘上只存在一个物理数据库, 并不存在另两个数据库. 概念模式可以看作从内模式抽象出来的对数据库的整体逻辑描述, 外模式可以看作从概念模式抽象出来的对用户局部逻辑结构的描述。

在SPARC分级图中, 每两级模式之间存在着从一种模式结构到另一种模式结构的映射. 这一功能是由DBMS支持的. 实际上, 映射不仅要完成三级模式之间的转换, 也为实现数据独立性提供了可能. 当物理数据库改变时, 只要适当修改概念/内部映射, 就可能保持概念模式不变, 实现数据的物理独立性. 同样地, 数据的逻辑独立性也可通过修改外部/概念映射来实现. 当然, 要维护映射使之能保持数据的独立性, 是一项十分复杂的工作。

DBMS除了支持各级映射外, 还要保证各级模式与应用程序的实现, 控制整个数据库系统的运行。

从以上的讨论可以看出, DBMS是一个多功能的、复杂的大型综合软件. 正因为把许多困难和复杂的任务交给DBMS去完成, 才换来了用户使用数据库的方便, 使数据库系统受到众多用户的欢迎。

0.4 数据库管理系统

现在, 我们结合DBMS的组成, 进一步介绍它的功能。

DBMS通常有三个组成部分, 它们是:

- (1) 数据描述语言及其翻译程序;
- (2) 数据操作语言及其翻译程序;
- (3) 管理和控制例行程序。

以下分别对它们作一简单介绍。

1. 数据描述语言 (Data Description Language)

这是在建立数据库时用来描述数据库结构的语言. 也有些文献称之为数据定义语言(Data Definition Language), 简称都是DDL。

按照数据库的模式分级, DDL又可细分为:

- (1) 模式DDL. 用来定义数据库的整体 (或全局) 逻辑结构;
- (2) 子模式DDL (或SDDL). 用来定义子模式 (或局部) 的逻辑结构;
- (3) 物理模式DDL (或PDDL). 用来描述物理数据库的结构。

用三种DDL来描述三种不同的模式,有利于实现上文提到的数据独立性。但实际的DBMS不一定将三种DDL分开,多数DBMS仅提供一或两种DDL,同时完成这三种语言的功能。

DDL是类似高级语言的形式化语言。用DDL描述的模式称为源模式,它构成数据库系统的“描述数据库”。通过DBMS配备的DDL翻译程序,能把源模式翻译为目标模式,构成系统的“目标数据库”。目标数据库通常由一组表格构成。

在下篇第一章1.3节,还要对DDL作更多的讨论。

2. 数据操作语言 (Data Manipulation Language)

简称DML。它是DBMS提供给用户的对数据库进行检索和存储的工具,也是用户与数据库之间的接口。

检索就是查询,在DML的各类功能中,查询具有重要的地位,所以有些系统把DML直称为查询语言。存储操作则包括数据的插入、删除和修改等操作。

在实际的DBMS中,DML又可分为两类。一类是可以独立使用的,它具有单独的编译或解释程序,可以在终端上用输入命令的方式立即执行,或用DML语言先编成应用程序,然后把程序送入计算机执行。通常称这类DML为自含型(self-containable),如上篇介绍的dBASE语言就属于自含型的DML。另一类称为嵌入型。它不能独立使用,而是嵌入汇编语言或COBOL, FORTRAN等高级语言之中,与被嵌入的语言一起使用。嵌入型DML又称数据子语言(Data sublanguage,简称DSL),被嵌入的语言称为宿主语言(host language)。有些系统的DML兼有自含型与嵌入型的功能,允许用户选择使用。下篇2.3节的SQL语言就是这种语言的一例。

DML的翻译程序可采用编译方式或解释方式。dBASE语言既有解释型,也有编译型。

3. 管理与控制例行程序

这部分的内容较杂,可包含安全性控制、完整性控制、并发控制、通信控制、数据存取、数据修改等例行程序以及工作日志、数据库转储、数据库初始装入、数据库重新组织等公用管理程序。对于不同的DBMS,它们的DDL与DML功能也强弱不同,但差异最大的是这一部分。微型机上使用的DBMS一般为简易型,许多管理与控制功能都被简化或省去了。

0.5 回顾与展望

1. 从格式化模型到关系模型

前已指出,在数据库中的各个主文卷,都以一定的结构形式相互联系(见0.2节)。按照实现这种联系所采用的不同方式,数据库的整体结构可区分为层次型、网络型和关系型三类数据模型(data model),其中前两类又合称为格式化模型(formatted model)。关于数据模型,见下篇5.5节。

较早的数据库系统都属于格式化模型。1968年IBM公司制成的IMS系统(层次模型结构)和1969年CODASYL委员会提出的DBTG系统(网络模型结构),就是格式化模型的典型代表。层次模型采用树结构的形式,这类系统复杂庞大,不易维护。网络系统采用循环链的结

构形式, 系统的数据表示能力和操作功能都比层次模型强。DBTG系统的出现, 对数据库技术的发展起了重大的推动作用。

1970年, IBM公司的E. F. Codd发表了题为“大型共享库集 (data bank) 的关系模型”的论文, 提出了关系模型的概念。这种模型用表格 (Codd称之为关系) 的形式来表示数据的联系, 运用数学方法研究数据库的结构和定义对数据库的操作 (称为关系运算), 从而把数据库的设计从过去的以经验为主提高到以理论为指导。1975年以后, 国外已有了商品化的关系数据库系统, IBM公司的QBE系统就是其中的典型代表。关系方法的出现, 极大地推动了对于数据库理论的研究。作为关系理论的创始人, Codd在1981年因此获得了美国计算机协会颁发的“图灵奖”。

网络模型在数据库领域统治了近十年。许多有代表性的计算机系统, 如IBM360/370, PDP-11/45, UNIVAC7090等都配有网络型的数据库。它的优点是效率高、技术比较成熟, 但存储结构复杂, 且要求用户详细描述存储结构、存取路径和数据结构, 使用与维护都比较复杂。与此相反, 关系模型采用人们惯用的表格结构形式, 不要求用户描述存储结构与存取路径, 易学易用, 加上关系语言的功能强, 使用灵活 (网络型DML查询时只能采取事先确定的存取路径, 关系型DML不受这种限制) 等特点, 使它从一开始就吸引了人们的注意。近几年来, 不仅微型机普遍采用关系型的DBMS, 也出现了一些在中、小型机上使用的关系型DBMS, 如System R、SQL/DS、DB2等。对关系方法的推广必将为数据库带来巨大的变革, 把数据库技术推向一个新的阶段。

2. 从集中式数据库到分布式数据库

随着计算机网络的应用, 数据库系统也从集中式向分布式发展。简单地说, 分布式数据库就是把数据分散存储在多个网络结点的数据库中。这些结点在地理上处于不同的地点, 彼此间靠通信线路来连接。例如, 一个银行有多个储蓄所, 每个储蓄所都有许多储户。如果将所有储户的信息全部存入一个集中式数据库中, 则当任一储蓄所需用储户的信息时, 都要到这个集中式数据库中存取, 数据通信的开销必然很大。如果使用分布式数据库, 储户的信息存储在最经常使用它的储蓄所, 则在大多数应用中, 信息可做到就近存取, 大大节省了数据在网内传送的开销。显然, 分布式数据库既能实现不同地点的用户对数据的共享, 又能减少使用集中式数据库时难以避免的通信开销。当然在实际应用中, 有些数据库 (例如库存管理和档案管理) 仍以集中式为宜。

建立分布式数据库可以有两种方法。一种是先在一些点上分散地建立各自的数据库, 然后把它们联结成网, 并建立一个协调系统来管理它们。另一种方法, 是在建立一个新网时, 就统一规划各个结点的数据库分布。这两种方法都有成功的实例。

设计分布式数据库的一个重要目标, 就是对用户来说, 它应该象一个集中式数据库一样工作。换句话说, 当用户存取数据时, 可不必过问这些数据实际上存放在什么地点。这样, 就能在改变数据的存放地点时, 作到不更改用户的应用程序, 这就是分布式数据库的数据独立性。但这是一个困难的目标, 目前已经实现的分布式数据库多数还达不到这个目标。

分布式数据库是当前数据库研究和发展的又一重要方向, 同时也给数据库设计提出了许多新的问题。例如当数据存放在多个结点时, 数据定义软件和数据管理软件应怎样配置和分布; 怎样实现各个结点之间的通信与协调; 怎样在网络条件下保持数据的独立性、完整性和