

数据库原理与设计

吴鹤龄 编

国防工业出版社

.87
39

数据库原理与设计

吴鹤龄 编

国防工业出版社

内 容 简 介

本书较全面地介绍了数据库系统的基本概念、工作原理、组织方式和设计方法。全书分四部分,共十二章,包括:绪论,数据和文件组织,数据模型,数据库的系统结构,数据描述语言DDL,数据操纵语言DML,关系数据库理论基础, System R介绍,数据库的逻辑设计、物理设计和可靠性设计。已基本上覆盖有关数据库系统原理与设计的主要课题。多数章节配有适量的习题。书末附有VAX-11 DBMS简介。

本书内容选材精炼,重点突出,尤其注重理论联系实际。叙述通俗,图文并茂,适于自学。

本书可用作高等院校计算机及有关专业的数据库课程教材或教学参考书,也可供从事数据处理的工程技术人员阅读。

数 据 库 原 理 与 设 计

吴鹤龄 编

*

国防工业出版社出版

新华书店北京发行所发行 各地新华书店经售

国防工业出版社印刷厂印装

*

787×1092 1/16 印张17¹/₂ 403千字

1987年12月第一版 1987年12月第一次印刷 印数: 0,001—3,800册

ISBN7-118-00018-3/TP3 定价: 2.95元

前 言

在计算机科学与技术这个公认为现代最重要的尖端科技领域中,数据库是一个比较新的分支。但是它的发展十分迅速,应用愈来愈广,在理论上和实践上都有重大的意义。有人认为,未来计算机与当今计算机的主要区别之一,在于前者将具有更加完善、功能更强的数据库系统,能在它的基础上建立知识信息处理系统,实现智能决策。

数据库技术于60年代中期出现不是偶然的。一方面,计算机硬件、软件在60年代得到迅速发展,IBM360系列的推出使计算机体系结构更加完善,数据存储技术愈益成熟;另一方面,整个科学技术和社会生产力在60年代也有空前的发展,需要由计算机处理的信息量愈来愈庞大,对数据处理技术提出了新的、更高的要求。数据库技术就是在这两方面的推动下出现的。比如,著名的、最早的数据库管理系统之一IMS就是Rockwell公司在实现阿波罗飞船登月计划中与IBM公司联合研制的。反过来,数据库技术的发展又推动了整个计算机科学技术的发展和其它科技领域中重大课题的解决。比如,1979年3月28日,位于美国宾夕法尼亚州哈里斯堡附近的三里岛上的核电站二号机组(TMI-2)因技术故障而发生震惊世界的放射性物质逸出事故(逸出散落物相当于一次大规模核试验带来的散落物)。为了防止类似的原子辐射事件,保障安全,美国政府和有关部门成立了专门机构进行了深入研究,最后决定采取有效的措施,其核心就是建立若干数据库,用于记录核电站设备运行情况,快速提供信息,进行故障率估算和早期预警(这些数据库后来被叫做CREDO, SACRD, SEE-IN等)。如果说,今天,已经建成和运行着的成千成万个数据库,以及将要建立起来的更多个数据库,是支持信息社会的全部活动的重要工具,在世界性的新技术革命中扮演着十分重要的角色,这是一点也不夸张的。

基于上述情况,数据库不但早已成为高等学校计算机专业大学生的必修课程之一,而且许多非计算机专业也要求学生具有数据库的基本知识,或者至少学会利用数据库来解决本专业的问题。本书就是为适应以上需要编写的。

本书初稿写于1982年上半年。1984年进行了修订充实。几年来,除了作者以该书为教材在北京工业学院计算机科学与工程系讲授数据库课程外,一些兄弟院校的计算机专业或研究生部也采用它作为教材或教学参考书。此外,本书还曾用作许多培训班的数据库教材培训在职技术人员。这次,由国防工业出版社正式出版前,作者根据几年来教学、科研实践的体会和用户的意见,对本书再次作了全面修改。修改时遵循的原则,一是尽可能保留原书的特色,二是尽可能增补原书的不足。因此,在内容体系上,不图表面的“全”,但求重点“深入”。对于三种主要的数据库模型,仍然只着重讨论网状和关系式数据库;在数据库设计部分,不介绍常见于文献的E-R图等方法,而介绍了一种实用的工程设计法;在编写方法上,仍然力求通俗易懂,理论联系实际,以便于读者自学。至于关系数据库部分的第八章,则几乎是完全重新编写的,以适应关系数据库理论当前的发展。

由于数据库的意义和作用日益重要,计算机上是否配置功能完善、性能良好的数据

库管理系统已成为计算机是否具有竞争能力的重要标志之一。因此,各计算机公司竞相推出自己的数据库管理系统DBMS。据资料统计,目前全世界约有300多种DBMS产品(这还不包括流行于微型机上的dBASE II等)。在一些发达国家,计算机上DBMS的普及率达90%以上。本书除对先进与典型的System R关系式DBMS专辟一章介绍,以附录形式简要介绍比较流行的典型网状DBMS——VAX-11 DBMS外,其它众多的具体DBMS限于篇幅不一一细述。但根据作者的亲身体会,读者在学习本书的过程中,必须按照可能条件,在任意一个DBMS支持下进行建库、编程、查询等多方面的实际练习,才有可能充分深入地理解数据库的基本概念与原理,掌握数据库设计的方法和工具。这一点是十分重要的。

最后,作者向在使用本书的前两个版本中提出过意见和建议的同行专家、读者和同学们表示感谢,这些意见和建议对于提高本书这次出版时的质量无疑起了重要的作用。作者特别感谢主审本书稿的华东工学院朱宗正教授[●],感谢他提出的许多宝贵而中肯的意见。对于为本书的出版付出辛勤劳动的兵器工业部教材编审室的贺白眉同志,作者表示深深的谢意,并且期待和热诚欢迎读者对本书的一切批评和改进意见。

吴鹤龄

● 朱宗正教授于1987年1月不幸去世,谨表示深切的哀悼。

目 录

第一部分 基础知识

第一章 绪论	1
§ 1-1 数据处理历史的简短回顾	1
§ 1-2 数据库的定义	4
§ 1-3 数据库的种类	6
§ 1-4 数据库的优缺点	9
第二章 数据和文件组织	12
§ 2-1 数据项	12
§ 2-2 记录	14
§ 2-3 串行文件	16
§ 2-4 顺序文件	17
§ 2-5 索引文件	19
2-5-1 索引非顺序文件	19
2-5-2 索引顺序文件	20
§ 2-6 直接文件	29
2-6-1 直接文件的装入和读出过程	30
2-6-2 影响直接文件效率的因素	31
2-6-3 动态直接文件——线性散列	37
2-6-4 一个实例	39
§ 2-7 倒排文件	40
2-7-1 什么叫倒排文件	40
2-7-2 倒排文件的分类及优缺点	41
2-7-3 一个例子	42
§ 2-8 表结构文件	44
2-8-1 什么叫表结构文件	44
2-8-2 指针的意义及分类	45
2-8-3 链	46
2-8-4 环	48
2-8-5 链和环的改进	48
§ 2-9 小结	49
习题	50
第三章 数据模型	52
§ 3-1 什么是数据模型	52
§ 3-2 数据项和记录间的联系及其表示	54
§ 3-3 层次式(树形)数据模型	56
3-3-1 树的基本概念	57
3-3-2 树形结构的物理实现	57
3-3-3 树形结构的优缺点及其应用	64

§ 3-4 网状数据模型	64
3-4-1 网状结构的基本特征及其由来	64
3-4-2 循环和回路——特殊的网状结构	67
3-4-3 简单网状结构的物理实现	68
3-4-4 复杂网状结构的物理实现	71
§ 3-5 关系数据模型	73
§ 3-6 树、网和关系式模型的相互变换	73
§ 3-7 小结	74
习题	75
第四章 数据库的系统结构	77
§ 4-1 三模式结构	77
§ 4-2 模式及 DDL	78
§ 4-3 分模式及 SDDL	79
§ 4-4 存储模式和 DSDL	80
§ 4-5 数据库管理系统	80
4-5-1 DBMS 的作用和地位	80
4-5-2 数据库管理系统的组成	81
4-5-3 数据库管理系统的工作过程	84
4-5-4 数据库管理系统的工作方式	85
§ 4-6 小结	86

第二部分 网状数据库

第五章 CODASYL数据描述语言	87
§ 5-1 CODASYL 与数据库系统	87
§ 5-2 CODASYL 数据库的数据层次	87
5-2-1 系	83
5-2-2 区	90
§ 5-3 CODASYL 模式DDL概要	90
5-3-1 DDL 的组成	91
5-3-2 记录存放方式和区的指定	92
5-3-3 对数据项的描述	93
5-3-4 系的实现方式和系序	95
5-3-5 入系类别和出系类别	102
5-3-6 排序键和查找键	102
5-3-7 系值选择	104
5-3-8 密码	107
§ 5-4 一个实例	110
§ 5-5 SDDL	113
§ 5-6 小结	116
习题	116
第六章 数据操纵语言 DML	119
§ 6-1 DML和主语言	119
§ 6-2 DML命令的分类及简介	120
§ 6-3 FIND/FETCH 命令	126

§ 6-4 小结	132
习题	132
第七章 查询语言	135
§ 7-1 源记录和虚记录	135
§ 7-2 QUERY命令简介	138
习题	141

第三部分 关系数据库

第八章 关系数据库理论基础	143
§ 8-1 关系的基本概念	143
8-1-1 关系的定义	143
8-1-2 域和属性	145
8-1-3 键	145
8-1-4 完整性规则	146
8-1-5 关系的内涵和外延	147
§ 8-2 关系代数	147
8-2-1 传统的集合运算	148
8-2-2 专门的关系运算	149
8-2-3 关系代数运算举例	152
§ 8-3 关系演算	154
8-3-1 面向元组的关系演算	154
8-3-2 面向域的关系演算	155
§ 8-4 关系中的数据依赖	157
8-4-1 函数依赖	157
8-4-2 函数依赖的蕴涵性	158
8-4-3 Armstrong 公理	159
8-4-4 计算闭包	162
8-4-5 函数依赖集的等价、覆盖和最小集	164
8-4-6 多值依赖	166
8-4-7 连接依赖	168
§ 8-5 模式分解	169
8-5-1 无损连接分解	170
8-5-2 分解的依赖保持性	173
§ 8-6 关系模式的规范化	175
8-6-1 第一范式	175
8-6-2 第二范式	176
8-6-3 第三范式	177
8-6-4 Boyce/Codd 范式	178
8-6-5 第四范式	178
8-6-6 第五范式	179
习题	180
第九章 System R 介绍	183
§ 9-1 System R 的总体结构	183
§ 9-2 System R 中的数据组织	185
9-2-1 System R 的逻辑数据组织	185

9-2-2 System R 的物理数据组织	187
§ 9-3 System R 中的数据操作	190
§ 9-4 System R 中的查询优化	190
§ 9-5 System R 的数据子语言SQL	193
9-5-1 SQL DDL	193
9-5-2 SQL DML	194
9-5-3 嵌入式SQL	200
§ 9-6 小结	204
习题	205

第四部分 数据库设计

第十章 数据库的逻辑设计	206
§ 10-1 数据库逻辑设计的内容和步骤	206
§ 10-2 示例	207
§ 10-3 数据库逻辑设计的优化	212
10-3-1 逻辑存取次数的分析计算	213
10-3-2 优化设计	215
§ 10-4 小结	219
习题	220
第十一章 数据库的物理设计	221
§ 11-1 数据库物理设计的任务	221
§ 11-2 数据表示	221
§ 11-3 数据库的分区、分页	224
§ 11-4 数据分配	227
§ 11-5 小结	233
第十二章 数据库的可靠性设计	234
§ 12-1 数据库环境下的可靠性问题	234
§ 12-2 数据的专用性、安全性和完整性	235
§ 12-3 数据保护	236
12-3-1 数据专用	236
12-3-2 数据校验	238
12-3-3 对并发操作的处理	237
§ 12-4 故障恢复	241
12-4-1 故障恢复的基本手段	241
12-4-2 故障恢复方式	245
§ 12-5 System R 对数据的保护	248
12-5-1 安全性	248
12-5-2 完整性	251
12-5-3 并发控制	251
12-5-4 故障恢复	252
§ 12-6 小结	252
习题	253
附录 VAX-11 DBMS 简介	254
主要参考书目	272

第一部分 基础知识

第一章 绪 论

这一章的目的是让读者对数据库获得一个基本的认识,即对数据库建立起一个粗略的、初步的、大致的概念。为此,我们在第一节中先简短地回顾一下数据处理发展的历史,分析传统电子数据处理方式的缺点——为克服这些缺点而努力的结果是产生了计算机的数据库系统。第二节介绍数据库的一个定义,给出组成数据库的三个要素,即数据、物理存储器和数据库软件。第三节按数据结构也即数据之间的联系把数据库分成层次式的、网状的和关系式的三大类予以介绍,并初步地作一对比。最后一节分析数据库系统主要的优缺点,并展示了数据库系统发展的广阔前景。

§ 1-1 数据处理历史的简短回顾

数据处理也叫信息处理。所谓信息(*information*),通俗说来,就是消息、新闻、情报。所谓数据(*data*),是表示信息的数字、字母和符号。可见这是两个不完全相同的概念。但由于它们之间有着紧密的、直接的关系,人们往往不区分它们。对反映现实世界的大量原始数据进行收集、整理、加工、存储和传播等一系列活动的总和就叫数据处理。经过处理的数据是精炼的,能够反映出事物或现象的本质、特征以及事物间的固有联系,从而帮助人们在生产斗争和科学实验中找出规律,作出决策。

数据处理的历史可以追溯到远古时代。原始人类用手指、石块、贝壳、木棍等来统计捕获野兽的数量,可以认为是数据处理的开端。我们今天常用的英语 *digital* 这个词,来源于拉丁语的“*digitus*”,原意就是“手指”。以后,随着文明的发展,数据处理经历了三个阶段:

1. 手工数据处理阶段

从原始社会到十九世纪末,由于社会生产力还比较低,科学技术不够发达,数据处理处于低级的手工阶段。在这阶段,人们致力于计算工具的改善,计算方法的革新,并取得了一系列辉煌的成就。我们的祖先在这方面有过杰出的贡献,如算盘的发明,圆周率的计算,簿记制度的创造等等。其他民族也都有各自的贡献,如 1614 年苏格兰贵族 J. Napier 发明对数;1641 年法国数学家 B. Pascal 制成基于齿轮机构的六位加法机;1650 年 Partridge 发明计算尺;1703 年德国数学家 G. W. Leibniz 创造二进制系统;1833 年英国剑桥大学教授 C. Babbage 发明微分机等。总的来说,这个阶段的数据处理离不开手工操作,效率低,处理的数据量小。

2. 机械数据处理阶段

1890 年,美国中央统计局的 H. Hollerith 为编制人口统计表发明了卡片制表机,使数据处理出现了一次重大的突破。Hollerith 发明的卡片机能以半自动方式进行穿孔、校验、分类、整理、制表等工作,这是机械数据处理的开始。比起前一阶段,处理效率有

很大提高。但机械设备的性能和效率仍然使数据处理受到很大的限制。

3. 电子数据处理阶段

四十年代电子计算机的出现为数据处理开辟了新的天地。电子计算机具有运算速度快、存储容量大、输入输出灵活多样等特点，正是数据处理所需要的。用电子计算机作为工具进行数据处理就叫电子数据处理 (*electronic data processing*, 缩写EDP)。

最初，电子计算机主要用于计算，解决科学与工程中的数学问题。世界上第一台电子计算机 ENIAC 就是为美国国防部解决弹道计算问题和编制射击表而研制、生产的。1951年，世界上第一个批量生产的商用计算机型号 UNIVAC-1 上开始具有不但能处理数字，也能处理字母的能力；同时，第一次用能正转和反转的高速磁带作为计算机的外存储器：这是对计算机进入数据处理领域具有决定意义的两个技术进展。从此，电子数据处理技术就以一日千里之势向前发展着。

用磁带作为数据的存储介质比卡片叠优越得多了。卡片容易穿错孔，容易搞乱，不便保管和使用。记录长度也受卡片限制。磁带则没有这些缺点，它的存储容量大，处理速度也相当快。但是磁带上的文件组织方法和卡片文件没有什么两样，也只能以顺序方式处理。要读出文件中的某个记录，必须先读出位于这个记录之前的所有不需要的记录，因而大大降低了工作效率。1952年，人们开始提出磁盘存储器的方案并进行试验。经过几年努力，1956年IBM公司和Remington Rand公司先后开始提供商用的磁盘系统。磁盘存储器可以随机存取，这是数据处理发展史上的又一个重大突破。

与此同时，数据管理软件也在发展。开始时，对数据的管理完全由程序员在程序中予以规划，每个应用程序直接对应于一个数据文件，见图 1-1。

后来出现了文件管理系统作为程序和数据文件之间的接口。应用程序可以通过文件管理系统和已经建立的一个或若干个文件打交道，增加了数据处理的灵活性(见图 1-2)。

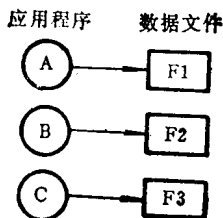


图1-1 电子数据处理的第一阶段
——程序高度依赖于数据

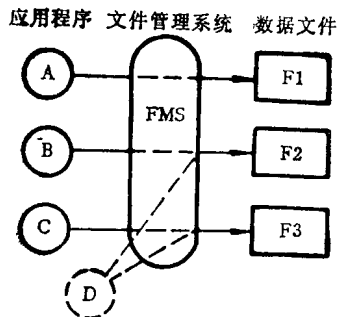


图1-2 电子数据处理的第二阶段
——文件管理系统

但这种处理方式仍然是以分散的、互相独立的数据文件为基础进行的。

这种数据处理方式有以下缺点：

1. 数据冗余。同一数据可能在存储器上多次重复存储，浪费宝贵的存储空间。

造成数据冗余的原因在于一个数据文件只为某一用户的某一特定用途服务。其中为别用户所同样需要的数据只好重复出现在别的数据文件中。比方说，一个大学，教务处有教学档案文件，科研处有科研档案文件，人事处有人事档案文件，财务科可能有个职工工资文件。在这些文件中，职工姓名、所在系、室等数据就会多次重复出现。在某

些系统中，一些常用数据重复存储的次数高达5次、6次甚至更多，使存储器的有效利用率极大降低。

2. 可能产生数据不一致性。这是数据冗余带来的必然后果。因为同一数据重复存储而由不同的用户使用和管理着，这就不能保证数据更新的同时性和正确性，从而可能使同一数据在不同的数据文件中有不同的值。例如，一个教员由某个系调往另一系后，人事处当然会立即修改其人事档案文件中的相应记录项目，使文件与已经变化了的现实一致。而教务处、科研处等却可能未及时收到有关信息而使他们的文件中保留着已经过时的记录项目。在上面这个例子中，这种不一致不会造成什么严重后果；而在某些情况下（比如有关军事、经济情报的文件系统中），则这种不一致的数据可能带来极大的危害。

3. 数据依赖。在这种数据管理方式之下，应用程序是与数据在存储器上的存储方式和存取方法紧密相关的，即编写应用程序的人必须清楚知道他要利用的数据的存储格式和记录定位方式，并由此决定他将采用的相应的存取方法。这一切都必须明白无误地反应在他的程序里，否则就不能达到取用和处理数据的目的。比如，在COBOL程序的数据部分就有文件节（DATA DIVISION, FILE SECTION），具体说明文件中各个数据项的类型和长度。在设备部分的输入-输出节（ENVIRONMENT DIVISION, INPUT-OUTPUT SECTION）中又要把文件和具体设备连系起来（*select*和*assign*子句），并规定文件的组织和存取方法（*organization*语句和*access mode*语句）。根据这些具体规定，程序员要在过程部分（PROCEDURE DIVISION）中用一系列命令语句“导航”，才能使系统完成预期的数据处理任务。应用程序与数据的存储、存取方式密切相关这种情况，我们称之为“数据相关”，或“数据依赖”（*data dependence*），也即编写程序依赖于具体数据。

数据相关给程序员增添了麻烦，也给程序的维护造成困难。一旦换用新的存储部件，或者改变数据文件的存储方式，所有有关的应用程序都要相应修改并重新调试，这是十分麻烦的。

4. 不灵活。每个已经建立的数据文件只限于一定的应用，且难于对它进行修改、扩充。在出现这种需要时，往往要重新建立一个新的数据文件，而这又使原有的应用程序无效。事实上，除了存档的历史性文件外，大多数数据文件的一个重要特点就在于它们是处于经常的变动与发展之中的。

这些缺点在规模较大的管理系统中尤为明显。比如，美国在60年代初制定了阿波罗登月计划。阿波罗飞船由约二百万个零部件组成。这二百万个零部件是分散在世界各地制造生产的。为了掌握计划进度及协调工程进展，阿波罗计划的主要合约者Rockwell公司开始时曾研制了一个基于磁带的零部件生产计算机管理系统，系统共用了18盘磁带，虽然可以工作，但效率极低。系统中60%是冗余数据，只能以批处理方式工作，维护也十分困难。这个系统的状况曾一度成为实现阿波罗计划的严重障碍。

针对上述问题，各国学者、计算机公司、计算机用户以及国际性的计算机学术团体都作了极大的努力，为改革数据处理系统进行探索、研究和试验。其目标主要在于突破文件系统分散管理的弱点，实现对数据的集中统一控制。上述Rockwell公司就与IBM公司合作，在当时新推出的IBM-360系列机上研制成功了世界上最早的数据库管理系统之一IMS（*information management system*），保证了阿波罗飞船1969年顺利登月。

在这类系统中,各应用程序不再和具体的物理存储器上的某一文件相对应,它们只各自相应于一个“逻辑”数据文件。这些逻辑文件通过管理软件同存储器上实际存储的数据建立联系,如图 1-3 所示。在这样的系统中,所有数据就象货物仓库中的各种物资被统一保管着那样,用户需要什么就到仓库中去提取,无需分散各自管理。因此,有人把这样的系统形象地称之为“数据库”(data base)。这个名称一经提出,立即得到公众的承认并迅速流传开来。从此,数据库系统就成为计算机科学中的一个新的也是十分重要的领域。当然,数据处理技术还会不断向前发展,但普遍认为,数据库系统是目前和将来数据处理的重要基石。

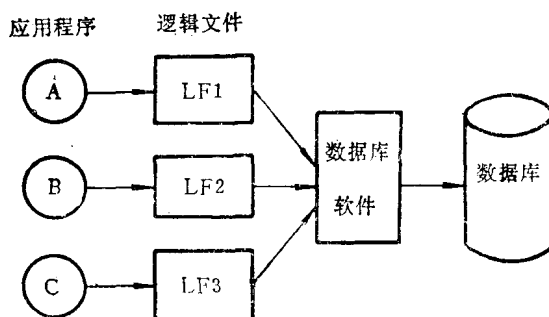


图1-3 电子数据处理的第三阶段——数据库系统

§ 1-2 数据库的定义

什么叫数据库?在计算机科学界中,目前还没有一个统一的、公认的定义。比如,计算机科学家Cohen用下面这段话来定义:“数据库系统是使数据集合并供用户们集体利用的程序系统”(Database system is the program system that makes a community of data available to a community of users)。

上述定义强调了数据库的集成性和共享性,而且十分简洁。其他科学家有不同的定义法,这里不一一引用。为了便于理解,我们可以略为详细地说,数据库是合理地存放在计算机存储设备上的相互关联的数据的集合。它具有如下特点:

- (1) 其中的数据具有最小冗余;
- (2) 其中的数据资源可被各个应用程序共享;
- (3) 其中的数据独立于应用程序;
- (4) 其中的数据由一个软件进行统一管理。

而对一个特定的数据库来说,它是集中统一地保存、管理着的某一单位或某一领域内所有有用信息的系统,这个系统是根据数据之间的自然联系结构而成的,从而能提供一切必需的路径存取其中的每个数据单元以满足所有用户的不同需要。

对上述定义我们可作如下进一步的说明。

1. 数据库是一个单位或一个领域的通用数据系统。“库”的大小视单位或领域的大小及信息量的多少而定。这里所说的“单位”可以是一个工厂,一个学校,一所医院,一个商店,一个政府部门或一个城市等等。“领域”可以是化工物性,钢材品种规格,民法与刑法等等。

2. 这个数据系统是“集成化”的。如同半导体电路工艺从分立元件发展到集成电路是一大进步一样,传统的数据处理方式中互相分立的数据文件发展到数据库就形成集成化的数据系统了,这就大大减少或避免了数据冗余。

3. 数据库中的数据结构基于数据间的自然联系,从而能提供一切必需的存取路径。

对某一数据单元的存取路径实际上是根据它和其它数据之间的联系而来的。比如我们后面将会看到,记录有所谓“双亲”记录和“子女”记录,那末可能利用的存取路径可以从“双亲”到“子女”,也可以从“子女”到“双亲”,或者也可以从一个“子女”到另一个“子女”,如此等等。为此,在数据库中一般都有相当复杂的数据结构。这是数据库和传统的数据文件的基本区别之一。

4. 数据库必须能满足所有用户的不同的需要,而且要有适当的效率。我们这里只说“适当的效率”而不说“高效率”,是因为一切数据库都不可能对所有用户“一视同仁”,都提供高效率的服务。重要的、常用的数据存取起来可能快一点,不大重要或不大常用的数据则可能慢一点。但无论如何,都应该有适当的效率。

5. 由此可见,一个完整的数据库系统应该具有以下三个基本要素:

(一) 数据

这是数据库系统的工作对象。当然,严格说来,数据库系统的工作对象包括数据本身和数据之间的联系二者。但我们后面将会看到,数据之间的联系实际上也是一种数据。因此我们可一般地说数据库的工作对象就是数据。

(二) 物理存储器

这是保存数据的硬件介质。因为数据库的数据量一般都相当大,因此数据库数据通常利用辅助存储器(或叫二级存储器)如磁盘、磁鼓加以保存。

(三) 数据库软件

这是对数据进行定义、描述以及使用和操作的手段,也即指挥、调度、维护数据存储器的工具。数据库软件的核心是数据库管理系统(*data base management system*, 缩写 *DBMS*)。

平时我们说到数据库或数据库系统时,有时指的是用户数据库,有时指的是数据库软件。关于数据库系统的软件配置,我们将在第四章中详细讨论。这里我们给出一个数据库系统的硬件支持,见图1-4。

数据库的物理实体通常是磁盘等直接存取的大容量存储器,有若干个盘组,每个盘组容量可达100MB以上。系统中包括远程终端供数据库用户使用,根据需要可以批处理方式工作,也可以在线方式工作。卡片机和行打用作数据库的输入输出。磁带机用于数据库的转储(*dump*),作为数据保护手段之一,详见十二章。至于计算机主机,在有数据库系统的情况下,要求有较大容量的内存,以容纳数据库软件中起核心作用的部分管理程序和例行程序以及数据库目录、表格等等。对计算机内存容量的最低要求因数据库管理系统大小不同、功能完善程度不同而异,从几十KB到几百KB。如IDS要求内存最小容量为96K字,相当于384KB;DMS-1100为128K字,相当于512KB。国产773小型计算机上配备的DBMS,其控制程序和服务程序的指令部分各占17KW(相当于34KB),数据空间前者要求最少15KW,后者为31KW。

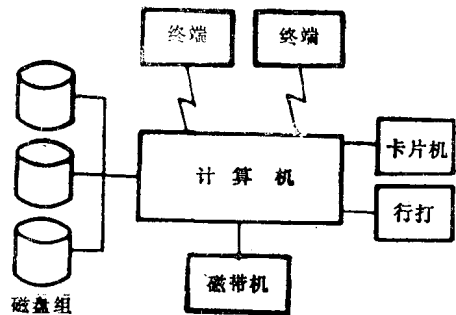


图1-4 一个数据库的硬件支持示意图

§ 1-3 数据库的种类

数据库常以所采用的数据库管理系统来分类。

目前，有文献介绍的数据库管理系统有几百个之多，作为产品由计算机公司提供的通用数据库管理系统也不少。在国际上比较著名和流行的若干数据库管理系统见表1-1。

表中这些由不同公司研制的数据库管理系统互不相同，各有特色。尽管如此，我们仍然可以根据表达数据间联系所采用的方法，把数据库分成以下三大基本类型：

1. 层次式数据库

层次式数据库也叫做树形数据库。这种数据库中的数据，就象一棵树一样，由根（主干）长出若干分支，每一分支又长出若干分支……，如图1-5所示。

在层次结构中，每一节点上有一种类型的数据记录，其下可有若干种其它记录类型。上一层的节点叫“双亲”（parent）节点，下一层次的节点叫“子女”（children）节点。每一双亲可以有几个子女（他们互为兄弟），但每个子女却只能属于唯一的双亲。

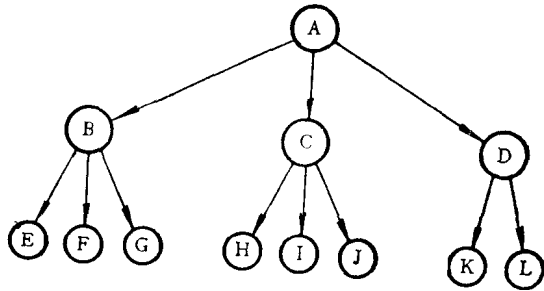


图1-5 数据库类型之一——层次式（树形）结构

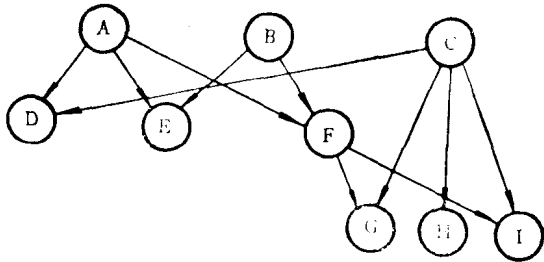


图1-6 数据库类型之二——网状结构

2. 网状数据库

如果层次式数据库中的一个子女可以属于几个双亲，那末其结构就变成网状的了，见图1-6。网状数据库也可叫成网络型数据库。

在网状数据库中，每一节点上的数据记录类型都可与其它任意相联的节点上的记录类型发生直接联系，而在层次结构中，只有相邻两级的父子节点才能直接相连。这是它们二者之间的一个根本区别。

3. 关系式数据库

在关系式数据库中，数据处于一些二维的平面表格之中，如图1-7所示。这种二维表格形式的数据结构是我们最熟悉的，日常生活、生产中用得很多。在数学上，每一个这样的表格就叫一个“关系”。

显然，关系式数据库和层次式及网状数据库有极大的区别。在层次和网状结构中，数据间的联系是通过其间的连线和箭头来表示的；而在关系结构中，各数据间不用任何连线和箭头，数据之间的联系是通过在不同的表格（关系）中相同的数据项（表格中的栏目）来表示的。这样，通过对关系进行各种代数运算（关系代数）就可存取任一数据项。而在层次和网状结构中，表达数据间联系的连线和箭头实际上常常是反映有关数据间的逻辑关系的物理指针，因而可由这些指针提供必需的存取路径。

关系结构远较层次式和网状的结构简单、直观，操作也简便，但早期投入实际应用

表1-1 若干流行的数据库管理系统

简称	全称	开发年代	研制单位	主机	主语言	类型	备注
ADABAS	adaptable data base	1968-1971	Software AG, BRD	IBM360/370 Siemens4004 UNIVAC9400 ICL2900	COBOL FORTRAN PL/I Assembler	网状	非CODASYL型, 单个数据库可包含255 个文件, 内存最少要求 约为110KB
EDMS	extended data management system	1970-1973	Xerox	Sigma 6 Sigma 8 Sigma 9	COBOL FORTRAN Assembler APL	网状	CODASYL型
DMS-1100	data management system-1100	1969-1971	Sperry univac	UNIVAC1100	COBOL FORTRAN PL/I	网状	CODASYL型, 内存要求128KW (每字为36位)
IDMS	integrated data management system	1973-1976	Cullinane Co.	IBM360/370 UNIVAC70, 90 Siemens4004 PDP-11/45 ICL2900	COBOL PL/I FORTRAN	网状	CODASYL型, 内存要求50KB-65KB, 但模式编译需150KB
IMAGE		1974	Hewlett Packard	HP2100 HP3000		网状	
IDS	integrated data store	1964-1966	General Electric/ Honeywell	GE200, 400, 500, H60, 200, 6000 UNIVAC1100	COBOL FORTRAN	层次	1975年发展为CODAS- YL型的IDS-I

简称	全称	开发年代	研制单位	主机	主语言	类型	备注
IMS	information management system	1965-1967	IBM/North American Rockwell	IBM360/370	COBOL PL/I Assembler	层次	内存要求, 220KB
SOCRATE		1970	univ of Grenoble/CII	IBM360/370 IRIS45, 80		网状	
SYSTEM2000		1970	univ of Texas/MRI	IBM360/370 UNIVAC1100 CDC 6000 CYBER70, 170	COBOL FORTRAN PL/I Assembler	层次	内存要求, IBM机-150KB, UNIVAC机-25KW, CDC机-18KW
SYSTEM R		1974-1979	IBM	IBM370	PL/I COBOL	关系	
TOTAL		1969-1971	CINCOM	IBM360/370 CDC6000 CYBER70 H200, 2000 Siemens4004 UNIVAC70, 90 Varian v 70 PDP11	COBOL PL/I FORTRAN Assembler	网状	非CODASYL型; 内存要求8KB-TOTAL4 14KB-TOTAL5/6 35KB-TOTAL7
UDS	universal data system	1975-1978	Siemens	Siemens7.700	COBOL FORTRAN Assembler	网状	CODASYL型