

大学双学位教材系列

计算机类

数据库系统概论

岳丽华 丁卫群 编著

 科学出版社

大学双学位教材系列·计算机类

数据库系统概论

岳丽华 丁卫群 编著

科学出版社

2000

内 容 简 介

本书主要介绍数据库系统的功能和结构方面的基本概念和实现技术。全书共分8章。第一章作为开篇,重点介绍基本概念和术语,为全书打下基础。第二章描述关系数据模型的结构及操作,同时还介绍了关系代数型操作语言的结构和内容。第三章介绍关系数据库语言SQL。第四章介绍关系模型的另一种操作语言——演算型语言。第五章介绍数据库系统的实现技术,这一章是全书的难点。第六章介绍数据库设计,主要介绍函数相关和规范化的概念。第七章介绍传统的网状数据模型和层次数据模型。第八章介绍当前较为先进的数据模型(OOD模型)及先进的系统结构(CLIENT/SERVER结构)。

本书主要是作为计算机专业大学本科生和双学位学生的教材,也可作为从事计算机信息处理研究和应用开发人员的参考资料。

科 学 出 版 社 出 版

北京东黄城根北街16号
邮政编码:100717

北 京 双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

*

2000年10月第 一 版 开本:787×1092 1/16

2000年10月第一次印刷 印张:12 3/4

印数:1—4 000 字数:285 000

ISBN 7-03-007395-9/TP·1094

定价:19.00元

(如有印装质量问题,我社负责调换〈环伟〉)

中国科学技术大学计算机类教材
编委会成员名单

主 任

陈国良 教授 计算机科学与技术系

委 员

赵振西 教授 计算机科学与技术系

赵保华 教授 计算机科学与技术系

周学海 讲师 计算机科学与技术系

庄镇泉 教授 电子科学与技术系

王仁华 教授 电子工程与信息科学系

李津生 教授 电子工程与信息科学系

王砚方 教授 近代物理系

刘振安 教授 自动化系

秘 书

李 蓓

序 言

当今科学技术发展的高度综合和相互交融、交叉，要求有更多的人掌握跨学科的知识，兼具承担数种性质、特征差异较大的工作的能力。为此，美国等国家的著名大学通过设立跨学科的专业，开设辅修、双学士学位和第二学士学位等途径培养跨学科的复合型人才。

早在 20 世纪 80 年代中后期，我国一些院校就试点开设辅修专业，学生可在攻读本专业的同时，再修读另一个专业指定课程一定数量的学分，从而获得辅修专业证书。同时，一些院校开始面向社会招收第二学士学位，大学本科毕业或工作一段时间后的工作人员，可以报考第二学士学位。90 年代中期，一些重点院校试点开设双学士学位，学生在学习本专业的同时，利用双休日可以再修读另一个专业学士学位的主要专业基础课程和专业主干课程，从而在大学毕业时可同时获得两个学士学位。从效果来看，尽管这些学生几乎放弃了两年的双休日，但社会人才市场却十分需要这种具有两个学科知识和能力的人才。

今天，计算机几乎应用于每个学科领域，许多学科因计算机的引入而开辟了新的领域。社会越来越多地需要既掌握一门专业知识，又熟悉计算机的人才。因此，无论是学校开设计算机辅修专业，还是计算机双学士学位或第二学士学位，都是学生攻读最多的专业。

随着理工科学生的公共基础课程越来越趋于相同，加上理工科院校都十分重视计算机技术基础知识的教学，这为攻读计算机辅修或双学士学位的学生提供了基础。中国科学技术大学正在进行教学改革，以拓宽本科生的基础知识。我们考虑到学时的限制，将计算机辅修或双学士学位的教学主干课程进行压缩，更加强调计算机科学技术的基础和应用。在此思想指导下，我们针对这些学生的特点，并充分考虑双学士学位的教学目标和模式而编写了本系列教材，希望这套教材在全国高校的双学士学位教学中能发挥其应有的作用。

中国科学技术大学
计算机类教材编委会
2000 年 8 月

前 言

起源于 20 世纪 60 年代的数据库技术,虽然历史不长,但却是计算机领域中发展最快的学科之一。

60 年代,随着计算机技术中文件系统的发展成熟,出现了多种文件组织方式和文件系统,再加上快速直接存取的磁盘存储设备的产生,计算机的应用从科学计算转到了管理工作,数据处理的重点也从大量的数据计算转向了数据管理,从而开始形成数据库的概念。1963 年,C. W. Bachman 设计开发了综合数据存储系统 IDS(Integrated Data Store),为多个 COBOL 程序提供了共享数据库,开创了数据库技术的先河。这一时期的数据库主要是层次数据库系统,初步实现了在同一物理数据上设计不同的逻辑文件,但是,这时的数据库结构还没有一个系统的描述文档,当然也就没有一个数据库设计实现的标准。

70 年代蓬勃发展的网状数据库系统,随着 CODASYL (Conference On Data System Language)给出的一系列数据库任务组(DataBase Task Group)工作报告(DBTG 报告)的发表,澄清了数据库上的许多模糊概念,建立了若干权威性观点,极大地推动了数据库技术的发展。

由 E. F. Codd 提出的关系型数据库系统,为 80 年代数据库技术的发展开创了一个崭新的天地,同时也为数据库的理论研究奠定了基础。IBM 公司 San Jose 研究所在 IBM 370 机上研制的 System R 系统,以及伯克利大学研制的 Ingres 系统,则更为关系型数据库的实现提供了更多的理论与实践经验。1978 年,美国标准化组织发表了数据库系统的最终报告 ANSI/X3/SPARC 建议,规定了数据库系统的总体结构及特征。1986 年 ANSI 通过的 SQL 美国国家标准,以及 1992 年 ISO 组织通过的 SQL 92 国际标准,给出了关系型数据库语言的规范。目前,市场上广为流传的 Oracle, Sybase, Informix, DB2, Ingres, SQL Server 等关系型数据库系统,都是采用 SQL 作为自己的系统语言,即使是那些早期实现的如 dbase, Foxbase 类系统,也都增加了与 SQL 标准的接口。这一系列标准的制定和系统的研制,使得关系型数据库应用正以前所未有的速度和广度发展,在人类社会生活中起着越来越重要的作用。

由于数据库技术既是一门非常实用的技术,同时又是一门涉及面广、研究范围宽的学科,因此它在经历了从第一代层次、网络数据库系统,发展到第二代关系型数据库系统,和第三代以面向对象为主要特征的数据库系统之后,随着计算机技术的发展,数据库技术又开辟了更多的新领域。目前,数据库技术与网络通信技术、面向对象技术、并行计算技术、多媒体技术、人工智能技术等互相渗透,互相结合,成为当前数据库技术发展的主要特征。

自 70 年代以来,数据库系统课程一直是计算机科学技术专业教育的主要课程之一。全国 300 多所设立计算机科学技术专业的大学中,几乎都把数据库课程设置为必修课程,因而有关数据库课程的教材也非常多。但是,随着数据库技术的深入发展,随着国内数据库应用的不断普及与提高,数据库应用系统已经从前几年微机版的 dbase、Foxbase 类转向了 Oracle, Sybase, DB2, SQL Server 等大型数据库系统。于是,数据库用户,特别是数据库

管理员(DBA)就不能仅仅只是了解系统可以为用户的请求做什么,而且还必须能使数据库以最大的效率工作。因此,数据库管理员的地位就越来越显出其重要性,而DBA对诸如逻辑设计、物理模块设计、查询事务处理、索引、并发、恢复等技术都要有较深入的理解。这样一来,对数据库课程的教学也就提出了更高的要求,不仅仅要从数据库的第一代系统转到以介绍第二代关系型数据库系统为主要内容,而且还要对DBMS的内部结构,即它的实现技术做较深入的介绍,还应使学生掌握如何用DBMS中存储的数据去描述现实世界、如何用DBMS中的数据表示用户对企业的查询,同时也还要使学生了解DBMS是如何处理并发存取、如何在系统故障发生时保护数据等知识。本讲义的编写即是基于这样的目标,并结合作者多年来从事数据库应用和数据库教学的经验,对数据库系统做一比较全面的介绍。

本书共分8章。第一章作为开篇,重点介绍基本概念和术语,为全书打下基础。第二章描述关系数据模型的结构及操作,同时还介绍了关系代数型操作语言的结构和内容。本章的重点是关系模型的定义和结构。关系代数作为今后研究数据库的重要内容也应给予必要的重视。第三章介绍关系数据库语言SQL。SQL语言作为关系型数据库的标准语言,不但在数据库研究方面占有重要的地位,而且也是商用数据库系统的标准语言。因此,本章结合具体数据库系统较详细地介绍了该语言的内容,学生学习后还应该在课程实践练习中更多地练习使用该语言编程。本章的难点重点应是视图、完整性限制说明及嵌入SQL结构。第四章介绍关系模型的另一操作语言——演算型语言。本章中,域演算、QBE语言等内容可以作为任选。第五章介绍数据库系统实现技术。这一章是全书的难点,内容包括事务处理、事务的性质、事务的执行调度等概念;并介绍了解决并发控制的锁技术以及解决故障发生的恢复技术;另外,对于备份技术和数据库安全等,也做了相应的介绍。本章所涉及的内容,既是数据库实现技术中的重要方面,也是数据库研究的重要内容。因此,本章虽不可能深入全面地讨论所述问题的方方面面,但是对其基本概念做了详细的介绍,为学生进一步学习打下基础。第六章介绍数据库设计,主要讲述函数相关和规范化的概念。该章的难点是多值相关,及其后的联结相关等概念。第七章介绍传统的数据模型和系统,即网状数据模型和层次数据模型。第八章介绍当前较为先进的数据模型(OOD模型),及先进的系统结构(CLIENT/SERVER结构),这些是为学生进一步学习作一定程度的准备工作。本书内容既可以作为一学期60学时的课程内容(不包括任选内容),也可以作为一学期80学时的课程内容。本书的授课对象是大学本科高年级学生,并假设他们已经学习了相应的先导课程,如数据结构、程序设计等。

本教材得以出版,应感谢中国科学技术大学教务处及学校信息科学教材编委会,是他们为作者提供了出版本书的机会,以及对作者编写本书的鼓励。科学出版社的编辑为此教材的出版做了大量的组织工作,科大计算机系龚育昌教授审阅了全书,在此表示感谢。另外,本书正式出版前,在作为讲义授课时,科大计算机系同学为校正本教材中的各类错误给予了帮助,周昊同学帮助绘制了书中大量的图表,在此一并致以深深的谢意。

由于作者水平有限,书中若有不当之处,敬请广大读者批评指正。

作 者

2000年3月

目 录

前言

第一章 数据库系统的基本概念	1
1.1 引论	1
1.1.1 数据库方法与文件处理方法	2
1.1.2 数据库系统用户	3
1.1.3 数据库方法的优点	5
1.2 数据模型与模式	5
1.2.1 数据模型分类	6
1.2.2 模式与实例	6
1.3 数据库管理系统的体系结构	7
1.3.1 三级模式体系结构	7
1.3.2 数据独立	8
1.4 数据库语言和接口	8
1.4.1 DBMS 语言	9
1.4.2 DBMS 界面	9
1.5 数据库系统环境	10
1.5.1 DBMS 的组成模块	10
1.5.2 数据库系统实用软件	11
1.5.3 通信功能	12
1.6 数据库管理系统分类	12
习题	14
第二章 关系模型与关系代数	16
2.1 关系模型的概念	16
2.1.1 域、元组、属性和关系	16
2.1.2 关系的性质	17
2.1.3 关系模型的符号	18
2.2 关系模型的语义限制	19
2.2.1 域限制	19
2.2.2 关键字限制	19
2.2.3 实体完整性限制	20
2.2.4 外关键字限制	20
2.3 关系的操作	21
2.3.1 插入操作	21
2.3.2 删除操作	21
2.3.3 修改操作	22
2.4 关系代数	22
2.4.1 选择操作	23

2.4.2	投影操作	24
2.4.3	命名操作	25
2.4.4	集合操作	28
2.4.5	连接操作	27
2.4.6	除法操作	27
2.4.7	其它操作	28
2.5	关系代数上的查询操作	31
	习题	31
第三章	关系数据库语言 SQL	35
3.1	SQL 语言数据库的三级体系结构	35
3.2	SQL 的数据定义	36
3.2.1	建表	36
3.2.2	修改表	37
3.2.3	删除表	37
3.2.4	表中的语义限制	37
3.2.5	完整的例子	39
3.3	SQL 的数据查询	42
3.3.1	基本查询语句块	42
3.3.2	基本查询块举例	42
3.3.3	SELECT 语句中的表达式及串操作	44
3.3.4	嵌套查询	45
3.3.5	集合操作查询	46
3.3.6	分组查询	47
3.3.7	SQL 的连接查询	50
3.3.8	SQL 查询中的空值	51
3.4	SQL 中的更新语句	52
3.4.1	插入语句	52
3.4.2	删除语句	53
3.4.3	修改语句	53
3.5	SQL 的视图	54
3.5.1	视图的定义	54
3.5.2	视图的查询	55
3.5.3	视图的更新	56
3.5.4	视图删除	58
3.6	SQL 的数据存取控制	58
3.6.1	自定义的存取控制	59
3.6.2	委托存取控制	61
3.6.3	数据库管理员的责任	62
3.7	索引说明	62
3.8	嵌入的 SQL	64
3.8.1	识别问题	65
3.8.2	通信区	65
3.8.3	数据交换	66

3.8.4 动态 SQL	68
习题	69
第四章 关系演算和 QBE	71
4.1 元组关系演算	71
4.1.1 元组变量及其范围	71
4.1.2 元组演算的形式描述	72
4.1.3 举例	73
4.1.4 全称量词	74
4.1.5 安全表达式	76
4.1.6 SQL 的量词描述	76
4.2 QUEL 语言	76
4.2.1 QUEL 中的数据与存储定义	77
4.2.2 查询结构	78
4.2.3 更新	81
4.2.4 QUEL 与 SQL 的比较	82
4.3 域关系演算	82
4.4 QBE 语言	84
4.4.1 查询结构	85
4.4.2 更新结构	88
4.5 关系数据库语言的比较	89
习题	90
第五章 数据库系统的实现技术	92
5.1 系统目录	92
5.2 查询处理与优化	93
5.3 数据库事务和系统日志	94
5.3.1 事务概念	94
5.3.2 系统日志	96
5.3.3 事务提交点	96
5.3.4 系统日志中的检查点	97
5.3.5 事务的并发执行	97
5.3.6 事务的读和写操作	98
5.3.7 并发失控的影响	98
5.3.8 事务状态和操作	99
5.4 事务执行调度	101
5.5 并发控制的锁技术	102
5.5.1 锁的类型	103
5.5.2 2 阶段锁保证可串行性	106
5.5.3 死锁与活锁	107
5.6 并发控制的 T/O 技术	109
5.6.1 时间戳	109
5.6.2 时间戳算法	110
5.7 基于延迟更新的恢复技术	111
5.7.1 恢复概念	111

5.7.2 恢复处理	112
5.7.3 回退	113
5.7.4 基于延迟更新的恢复技术	114
5.8 基于立即更新的恢复技术	117
5.8.1 单用户环境下基于立即更新的撤消/重做算法	118
5.8.2 与并发执行相关的 UNDO/REDO 立即更新	118
5.9 数据库的备份和恢复	119
5.10 数据库的安全	119
5.10.1 安全的类型	119
5.10.2 数据库管理员与安全	121
5.10.3 用户帐号和数据库审计	121
5.11 基于优先权的自裁定存取控制	122
5.12 多级保密的强制性存取控制	122
习题	125
第六章 数据库设计	127
6.1 关系数据库设计步骤	127
6.2 E-R 模型	128
6.2.1 实体、属性和实体型	128
6.2.2 联系	129
6.3 模式优化	131
6.3.1 关系属性的语义	131
6.3.2 元组及冗余信息	132
6.3.3 空值	133
6.3.4 伪造的元组	134
6.4 函数相关	135
6.5 关系模式的优化	135
6.5.1 实体集上的限制	135
6.5.2 联系集上的限制	136
6.5.3 实体属性的确定	136
6.5.4 实体集的确定	136
6.6 规范化	137
6.6.1 2NF	137
6.6.2 3NF	138
6.6.3 BCNF	138
6.7 多值相关与 4NF	139
6.7.1 多值相关	139
6.7.2 MVD 与 FD	140
6.7.3 4NF	141
6.8 连接相关与 5NF	141
6.9 其它相关性和规范化	142
6.9.1 包含相关	142
6.9.2 Domain-Key NF	142
习题	143

第七章 传统数据库方法	145
7.1 网状数据结构	145
7.1.1 记录和记录型	145
7.1.2 系型及其系值	146
7.1.3 特殊系型	147
7.1.4 系值的存储结构	148
7.1.5 1:1 和 M:N 联系的系结构描述	149
7.2 网状模型的语义限制	150
7.2.1 插入(选择)限制	150
7.2.2 离开(选择)限制	150
7.2.3 系序选择	151
7.3 网状模型定义	151
7.4 网状数据库的数据操作	156
7.4.1 用户工作区 UWA	156
7.4.2 当前指示器	157
7.4.3 状态指示器	158
7.4.4 当前指示器和 UWA 的图示	158
7.4.5 网状数据操作语言(DML)	159
7.5 层次数据库结构	161
7.5.1 层次数据结构	161
7.5.2 层次结构值树	163
7.6 虚拟父子联系	164
7.7 层次模型中的完整性约束	166
7.8 层次模型的数据定义	166
7.9 层次模型中的数据操作	168
7.9.1 HDML 命令的 UWA 和当前值	168
7.9.2 操作命令举例	170
习题	172
第八章 数据库技术进展	175
8.1 先进的数据模型	175
8.2 面向对象数据库基本概念	176
8.3 对象结构和类型	177
8.3.1 对象结构	178
8.3.2 类型构造函数	180
8.4 操作、方法的封装和持久性	181
8.5 类与类型的层次结构及其继承性	184
8.5.1 类型层次与继承	184
8.5.2 类层次	185
8.6 Client/Server 结构	186
习题	188
参考文献	190

第一章 数据库系统的基本概念

1.1 引 论

数据库技术自 20 世纪 60 年代出现以来,虽然历史并不长远,但是其发展相当迅速,已成为计算机科学中发展最快的领域之一。如今,一个企业的成功与否都更多地依赖于她及时获取精确数据的能力,和她操作与分析这些数据的能力。

随着“信息高速公路”在世界各国不断地延伸,随着信息处理已成为可以快速产生亿万富翁的产业,人们越来越认识到数据是企业的财富。当今社会是信息爆炸的时代,信息量增长的速度以及信息量本身的意义都是人们始料不及的。如果没有管理这样大量的数据的能力,不能迅速从如此大量数据中发现与问题有关的信息,那么随着信息量的增加,数据就将不再是企业的财富,而只能是一场灾难。因此,为了能管理和使用如此大量的数据集,用户必须要有数据管理工具。数据管理工具不仅能简化数据管理任务,并且还能及时地从数据中抽取有用的信息。数据库技术就是这种工具之一。

对于数据库,我们可以简单地定义为,它是相关数据的集合,其中的数据必须是可以被记录的、且具有一定意义的事实。例如,人口登记表中的姓名、性别、家庭住址等等。这些数据既可以存放在登记表中,也可以在计算机上用相关的软件(如 Fox 等)记录在磁盘上。由于登记表中这些数据互相相关并具有一定的意义,因此,登记表中这些相关数据的集合就是数据库。

以上对数据库的定义是不严格的。实际上,当我们使用数据库这个术语时,就隐含着数据库应具有如下性质:

(1) 数据库描述了现实世界中的某些方面,构成了现实世界中的一个微小世界。该微小世界的变化在数据库中都能得到反映。

(2) 数据库是一个逻辑上紧密相连的数据集。该数据集中的数据具有某些固有的语义含义。任意数据的什锦拼盘都不能被认为是数据库。

(3) 数据库是为某个特定目标设计、建立和使用的,它拥有确定的用户组和这些用户组感兴趣的预定的应用。

数据库可以大也可以小,复杂度可以高也可以低。例如,只有数百个记录组成的人名地址表库,其记录结构非常简单。而图书馆的图书卡目录库,可以在不同目录下有上百万张卡片记录。更有甚者,一个国家的税收服务则是一个数据量更大、数据结构更复杂的数据库。例如,假设有 1 亿个纳税人,每个纳税人平均需填 2 张报表,每个报表含 200 字节(B),于是就有了一个 $100 \times 10^6 \times 200 \times 2$ 字节的数据库。目前,拥有上百个千兆字节(CB)数据量和上千个用户的数据库是很常见的。

数据库可以由人工维护,也可以由机器维护。计算机化的数据库可以由一组专用的应用程序来建立和维护,也可以由数据库管理系统来建立和维护。

数据库管理系统 DBMS(Database Management System)是计算机程序的集合,它被数据

库用户用来创建和维护数据库,因此,它是一个通用软件系统。数据库管理系统一般应具有的功能包括:定义、创建、操作和维护数据库。定义数据库涉及描述存储在数据库中数据的数据类型、描述这些数据的数据结构和描述这些数据所具有的各种语义限制。创建数据库则是在数据库管理系统的控制下把这些数据存储在某种存储介质上。数据的操作则包括如查询、更新和生成报表等一系列处理功能。

需要指出的是,并不是只有数据库管理系统软件才能实现计算机化的数据库。我们也能够编写自己的程序集,然后使用这些程序集去建立和维护数据库。实际上,这也就是说通过编写一个专用的数据库管理系统软件来创建和维护自己的数据库。对于数据库管理系统而言,不论是专用的,还是通用的,通常都必须使用大量的软件系统来操作数据库和使用数据库本身。所以,我们把数据库和这些软件系统合在一起统称其为数据库系统 DBS(Database System)。

随着越来越多的数据成为可以从计算机网络上获得的在线信息,数据库技术也就越来越重要。今天,该领域正在被更加激动人心的场面所驱动,例如多媒体数据库、数字图书馆、人类基因映射、企业期盼的信息决策、有用信息挖掘等等,使得数据库系统代表着更大更重要的市场,因此,数据库系统的学习是非常重要的。

另一方面,数据库管理系统领域使用的技术跨越了多个领域,包括语言、面向对象和其它程序设计方法、编译、操作系统、并发程序设计、数据结构、算法、理论、并行与分布系统、用户界面、专家系统和人工智能、统计技术、线性规划等等。因此,学习数据库管理系统也要求对计算机科学各个领域知识有一定的基础,学习者也可以从多种角度而非一种角度来学习和研究。

在详细介绍数据库系统之前,我们先对数据库方法与文件方法的处理方式,以及数据库方法的优点和数据库系统涉及的人员等做一初步介绍。

1.1.1 数据库方法与文件处理方法

如上所述,对数据库的管理和维护既可以使用数据库管理系统来组织和处理,也可以编写自己的程序,按照文件方式来组织与处理,那么这两种方式的差别是什么呢?

1. 数据库的自说明性质

数据库系统中,不仅含有数据库本身,而且还有完整的有关数据库的定义描述。定义描述被存放在数据字典(Data Dictionary)之中,其内容包括每个文件的结构、数据项的类型及存储格式、数据项上应具有的语义限制等。存放在数据字典中的数据称为元数据(Metadata)。

数据字典通常被数据库管理系统使用,偶尔也可被需要知道数据库结构信息的用户所使用。由于数据库管理系统的软件并不是为某个特定的数据库服务的,所以当它要存取某个数据库中的数据时,必须先通过引用数据字典来了解该数据库中的文件结构,如存取的数据类型和格式等信息。

传统的文件处理中,数据是应用程序本身的一部分。因此,程序仅仅是对一个特定的数据库工作,其数据结构是在应用程序中说明的,例如,Pascal 程序的记录结构,C++ 程序中的类说明等。因此,文件处理只能存取特定的数据库。数据库管理系统软件则可以通

过引用数据字典来存取各种数据库。

2. 程序与数据的独立

在传统文件处理中,数据文件结构是嵌入在存取程序之中的,对文件结构的任何改变,都需要对存取这些文件的所有程序也做相应修改。因此说,文件处理中程序与数据是紧密关联的。而在数据库系统中,数据库管理系统存取程序是独立于任何特定文件编写的,数据文件结构做为数据字典的一部分被存储,是独立于存取程序的。

允许程序与数据独立、程序与操作独立的性质被称为数据抽象(Data Abstraction)。数据库管理系统为用户提供的数据库抽象包含数据的概念描述(Conceptual Representation)。概念描述中只使用逻辑概念来描述数据,不包括任何有关数据是如何存储的细节

3. 数据的多视图

数据库有很多用户,不同用户又各自有其不同的数据观点,也称为有不同的数据视图。

视图(View)是数据库子集,它可以含有虚拟数据,也就是说,这些数据可以从数据库中推导出来,而不是实际存储在数据库当中。用户通常只关心能否存取所需要的数据,并不关心这些数据是推导得出的还是从实际存储中直接得到的。具有各种应用的多用户数据库管理系统,必须提供可以定义多视图的功能。

4. 数据共享和多用户事务处理

由于数据库中的数据是多个应用的数据综合地维护在一个数据库中,因此多用户数据库管理系统就要允许有多个用户在同一时刻对同一数据进行存取。例如,在订票应用系统中,若在多个订票点都要预订同一个剧场的同一台演出中的同一个座位,数据库管理系统就必须保证那个座位的数据只能在一个订票点为一个顾客访问并修改。这个处理过程通常称为多用户事务处理。数据库管理系统必须提供并发控制软件,保证并发事务正确地处理而不相互影响,以免造成数据库的不一致性。

1.1.2 数据库系统用户

数据库系统的用户很多,具有上千个用户的系统也很常见。使用数据库的用户类型也是多种多样的,现按照其担负的责任和具有的功能分别讨论如下:

1. DBA(Database Administrator)

在数据库环境中,系统资源一是数据库本身,一是数据库管理系统软件。对于这样一个由多个用户使用的共享资源,必须有一个主管人(或组)负责对资源的管理与维护。DBA(数据库管理员)就是这样的一个主管人。

2. 数据库设计者

数据库设计者负责对被存放在数据库中的数据选择合适的结构来描述和存储数据。要完成上述任务,数据库设计者必须与所有用户交流,获取对用户需求的理解,然后给出

一个满足这些需求的设计。另外,数据库设计者不仅要与目前的数据库用户交流,还特别要与潜在的用户组交流,为这些潜在用户设计满足其需要的数据视图,然后将这个视图与其它用户视图综合分析,最终设计出支持所有用户组需求的数据库。

数据库设计者可以是 DBA 的成员,也可以是数据库设计完成之后工作的成员

3. 终端用户

终端用户的工作是为了查询、更新、生成报表而存取数据库的工作人员,根据他们使用数据库的频率和熟悉数据库的程度不同,可以进一步细分为:

(1) 临时用户

该类用户不定期地、偶尔地使用数据库。这一类用户每次需要的信息不同,他们使用比较复杂的数据库查询语言描述需求。通常这类用户是企业中的中层或高层管理人员

(2) 初级用户

这是数据库终端用户的主体。他们的主要工作是使用标准的查询或更新事务(或称其为套装事务(Canned Transaction))经常对数据库查询或更新。例如,银行中的柜台出纳员,订票应用中的售票人员等。

(3) 熟练用户

该类用户包括工程设计人员、科学家、商情分析员等对数据库管理系统的功能很熟悉的人员

4. 系统分析员和应用程序员

系统分析员的职责是对用户,特别是初级用户的需求进行分析,给出这些需求事务处理的说明。应用程序员则是用程序实现系统分析员给出的事务处理说明,并完成这些程序的调试、文档编写、维护等工作。系统分析员和应用程序员都必须对数据库管理系统的功能有较深入的了解。

5. 系统软件工作人员

这一类工作人员不仅仅是对数据库本身有兴趣的人员,而且主要是与数据库管理软件的设计、开发和操作有关的人员。这些人员包括:

(1) 数据库管理系统软件的设计和实现者

数据库管理系统是一个复杂的软件系统,它包含很多有关查询语言、界面接口、数据字典、数据存取和安全等功能的软件模块。另外,数据库管理系统还必须有与其它系统软件如操作系统、各种程序设计语言编译程序的接口软件。数据库管理系统软件的设计和实现者的责任就是编写完成以上功能模块的软件工作者。

(2) 工具开发者

工具开发者负责实现为便于数据库设计和使用、帮助改进数据库系统性能等需要而设计的各种工具软件。

通常这些工具软件都是由一些与 DBMS 开发商独立的软件商完成的。

(3) 操作与维护人员

这类工作人员是指具体负责数据库系统环境中各种软件和硬件实际运转的管理与维

护人员。

虽然系统软件工作人员的工作是使数据库系统对终端用户更有效,但是这些软件人员通常并不使用数据库本身。

1.1.3 数据库方法的优点

人们选择数据库方法来做数据处理,通常是因为数据库管理系统的使用具有如下几个主要优点:

1. 加强标准化

由于是通过 DBA 人员(或工作组)来设计数据库,因此由 DBA 获得了一个组织内不同部门、不同项目以及用户之间对数据处理的交流与掌握,从而可以定义出标准的数据元素的名字、格式、报表结构、术语等,为组织内部统一使用,避免了数据同名不同义、同义不同名等混淆问题。

2. 减少应用开发时间

随着计算机越来越被人们接受,用计算机来实现数据处理应用的需求也就越来越多,于是,减少这类应用的开发时间就变得越来越重要。

从无到有地设计和实现一个新的数据库当然需要很长的时间和经费。但是,一旦该数据库建立起来,使用数据库管理系统技术在该数据库上实现新的应用则只需很短时间就可完成。有人估计,数据库管理系统技术开发应用的时间只是传统文件系统开发时间的 1/4 到 1/6。

3. 灵活性

当数据库建立起来后,随着用户需求的变化,数据库结构也应该跟着变化。例如,当用户需要增加当前数据库中不存在的信息时,由于 DBMS 技术的本身性质,它允许数据库结构变化但不影响已存在的应用程序和已存储的数据。

4. 信息的可用性

数据库中的数据是面向所有用户的,因此当某个用户对数据库做了修改后,所有用户立即就可感知到其已修改后的更新数据,获得了数据的最新可用性。例如,当某个职工要调离时,只要人事部门办理了该职工的调离手续,人事档案中的数据立即为该组织所有应用共知,也就避免了财务部门因人事的修改延迟还继续发放该职工工资的怪事。

1.2 数据模型与模式

当我们描述现实世界时,通常是采用一种抽象的概念模型来描述。例如,我们可以用数学的观点来描述现实世界,出现了某个应用的数学模型。我们也可以用物理学的观念来描述现实世界,从而得到该世界的物理模型。在数据库方法中,我们是用数据的观点来描述现实世界,因此,我们获取了这个世界的数据库模型。