

面向对象 数据库技术

王意洁 编著



電子工業出版社

Publishing House of Electronics Industry
<http://www.phei.com.cn>

内 容 简 介

本书全面系统地论述了面向对象数据库的基本理论、国际标准、设计原理、实现技术、应用,以及这一领域的最新进展。全书共分 10 章,内容包括面向对象数据库的标准化、体系结构、数据建模、存储管理、查询处理、并行查询处理、事务管理、模式演变与版本管理及安全等,并介绍新一代面向对象数据库系统的代表产品——Objectivity/DB。

本书内容全面,取材新颖,基本上覆盖了面向对象数据库技术的全部内容,反映了国际上的最新研究水平,具有很好的系统性、完整性、先进性和实用性。本书适合从事数据库研究及开发人员阅读,也可以作为高等院校计算机、电子技术、自动控制和系统工程等有关专业的研究生和高年级本科生的教材或参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

面向对象的数据库技术 / 王意洁编著. —北京: 电子工业出版社, 2003.3

ISBN 7-5053-8481-3

I.面... II.王... III.关系数据库-数据库管理系统 IV.TP311.13

中国版本图书馆 CIP 数据核字(2003)第 007354 号

责任编辑: 赵 平

特约编辑: 孙学瑛

印 刷: 北京东光印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1 092 1/16 印张: 18 字数: 461 千字

版 次: 2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

印 数: 4000 册 定价: 35.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话: (010) 68279077

前 言

随着计算机技术的发展和应用需求的推动,面向对象的数据库技术受到了广泛的关注。面向对象的数据库对现实世界中复杂对象具有很强的描述能力以及高效率开发应用系统和实现软件复用的能力。它在传统数据库技术难以有效支持的一些应用领域(如 CAD, CIMS 等)中最先获得成功的应用。近年来,随着计算机软硬件技术和网络技术的飞速发展,尤其随着各种电子文档、多媒体文件、复杂的数据格式等非传统海量数据的大量涌现,人们发现面向对象数据库技术有了愈来愈多的应用领域,如 CASE、多媒体应用、文档管理、工作流、地理信息系统、各行各业的电子商务和各种 Internet/Intranet/Extranet 的应用等。面向对象的数据库技术继续扮演着有效管理各类复杂数据的重要角色,为推动国民经济的发展起到重要的作用。另一方面,面向对象的数据库技术对关系数据库产品产生了深刻的影响,引出了对象关系数据库的概念,促进了传统关系数据库的发展。面向对象数据库和关系数据库将在各自专长的领域内发挥重要的作用。它们将长期处于并存状态,彼此影响,相互渗透。因此,以发展的眼光看,面向对象数据库技术的研究具有重要的学术价值和应用前景。凡是关心数据库发展和面向对象技术的计算机工作者,都很有必要了解面向对象数据库的技术发展。

本书参考了国内外有关面向对象数据库技术的经典著作和最新文献,也融会了作者在该领域的研究成果。本书内容涵盖了面向对象数据库的基本理论、关键技术和相关国际标准。其关键技术包括数据模型、体系结构、存储管理、查询、事务管理、安全与版本管理及模式演变等。本书还介绍了作者在事务处理和查询处理方面的研究成果,内容翔实全面,取材新颖,充分展示了面向对象数据库技术的最新研究动态和发展趋势,具有很好的系统性、完整性、先进性和实用性。本书共分 10 章。第 1 章为引论;第 2 章介绍面向对象数据库系统的体系结构;第 3 章介绍面向对象数据模型及对象定义语言;第 4 章介绍对象存储技术;第 5、第 6 章介绍查询语言、查询处理及其并行化技术;第 7、第 8、第 9 章分别介绍面向对象数据库中的事务处理、模式演变与版本管理及安全技术;第 10 章介绍新一代面向对象数据库系统的代表产品——Objectivity/DB。

由于作者本身水平有限,书中难免存在一些缺点和错误,恳请专家、读者批评指正。

作 者

2002 年 8 月 24 日

目 录

第 1 章 引论	1
1.1 数据库技术的发展	1
1.1.1 网状数据库	1
1.1.2 层次数据库	2
1.1.3 关系数据库	2
1.1.4 新一代数据库技术	2
1.2 面向对象数据库	8
1.2.1 面向对象数据库的基本特征	8
1.2.2 面向对象数据库的标准化	9
1.2.3 面向对象数据库的应用	11
1.2.4 现有的面向对象数据库系统	19
1.3 并行面向对象数据库的发展	21
第 2 章 面向对象数据库系统的体系结构	24
2.1 面向对象数据库系统的客户/服务器结构	24
2.2 面向对象数据库系统的页服务器体系结构	26
2.2.1 提高性能的关键技术	26
2.2.2 变体	27
2.3 客户/服务器面向对象数据库系统 KDOODB 的体系结构	29
2.3.1 总体结构	29
2.3.2 客户/服务器结构	32
第 3 章 数据建模	35
3.1 传统的数据模型	35
3.1.1 网状数据模型	35
3.1.2 层次数据模型	36
3.1.3 关系数据模型	36
3.1.4 评价	37
3.2 面向对象数据模型	37
3.2.1 类型	38
3.2.2 对象	41
3.2.3 文字	53
3.2.4 类型层次	55
3.2.5 性质	55
3.2.6 操作	58
3.2.7 元数据	58
3.2.8 并发控制	61
3.2.9 事务模型	63
3.2.10 数据库	64

3.3	对象定义语言	65
3.3.1	类型特性	66
3.3.2	实例性质	67
3.3.3	操作	68
3.3.4	举例	69
3.4	对象交换格式	72
3.4.1	初始化属性值	72
3.4.2	初始化关系	75
3.5	C++联编	75
3.5.1	将 ODMG 对象模型映射到 C++	76
3.5.2	C++ODL	77
3.5.3	C++OML	85
3.5.4	应用举例	101
第 4 章	存储管理	104
4.1	对象存储	104
4.1.1	大对象	105
4.1.2	磁盘空间管理	107
4.2	聚簇	108
4.2.1	基于排序的聚簇策略	110
4.2.2	预分	111
4.2.3	遍历	113
4.3	索引	116
4.3.1	一维索引	117
4.3.2	多维索引	123
4.4	指针转换	127
4.4.1	就地转换与复制转换	128
4.4.2	主动转换与懒惰转换	130
4.4.3	直接转换与间接转换	130
第 5 章	查询语言	132
5.1	对象查询语言	132
5.1.1	语言定义	132
5.1.2	OQL 的巴克斯范式 (BNF)	142
5.1.3	C++OQL	148
5.1.4	应用举例	150
5.2	SQL3	152
5.2.1	行类型	153
5.2.2	抽象数据类型	156
5.3	比较 ODMG 标准与 SQL3 标准	157
第 6 章	查询处理	159
6.1	查询优化	159

6.2	循环查询	162
6.2.1	循环查询的定义	162
6.2.2	“分而治之”的循环查询处理策略	163
6.3	并行查询处理	170
6.3.1	查询处理的分阶段执行策略	174
6.3.2	基于对象类的混合式数据放置策略	175
6.3.3	基于合格标记的并行数据操作算法	178
6.3.4	多元连接查询的并行执行	191
6.3.5	并行查询优化	194
6.3.6	循环查询的并行处理	198
第 7 章	事务处理	202
7.1	事务	202
7.2	事务处理的核心	203
7.2.1	正确性准则	203
7.2.2	事务模型	207
7.2.3	评述	211
7.3	并发控制	212
7.3.1	传统并发控制	212
7.3.2	客户/服务器结构中的并发控制	216
7.3.3	在并发控制中利用版本信息	218
7.4	恢复	219
7.4.1	基于页的日志序号	220
7.4.2	WAL_P 的数据结构	220
7.4.3	事务的正常执行	222
7.4.4	系统重启	226
7.4.5	介质故障恢复	232
7.5	事务标识符分配	232
7.5.1	事务处理对事务标识符的需求	233
7.5.2	事务标识符的分配策略	234
第 8 章	模式演变与版本管理	239
8.1	模式演变	239
8.1.1	模式	239
8.1.2	模式一致性	242
8.1.3	模式演变分类	243
8.2	版本管理	244
8.2.1	版本集	244
8.2.2	引用机制与配置管理	245
8.2.3	版本层次结构	247
8.2.4	变动传播与通告	248
8.2.5	分布式环境中的版本管理	249

8.2.6 实现.....	250
第 9 章 安全.....	253
9.1 身份验证.....	253
9.1.1 口令.....	255
9.1.2 密码智能卡.....	255
9.1.3 生物信息.....	255
9.2 访问控制.....	256
9.2.1 访问控制矩阵.....	256
9.2.2 访问控制策略.....	259
9.2.3 授权管理.....	263
9.3 审计与入侵检测.....	263
9.3.1 入侵检测系统.....	263
9.3.2 审计控制.....	266
9.4 基本授权模型.....	267
第 10 章 新一代面向对象数据库系统 Objectivity/DB.....	272
10.1 Objectivity/DB 核心数据库.....	272
10.2 Objectivity/DB 容错选项.....	273
10.3 Objectivity/DB 数据复制选项.....	274
10.4 Objectivity 开放文件系统.....	275
10.5 Objectivity C++接口.....	276
10.6 Objectivity XML 接口.....	277
10.7 Rose Objectivity 连接器.....	278
参考文献.....	279

第1章 引 论

什么是数据库？数据库就是一个数据集合，是一个长期存在的信息集合。它由数据库管理系统进行管理。数据库管理系统通常也称为数据库系统，具有如下特点。

(1) 允许用户利用数据定义语言说明数据的逻辑结构(即模式)，并根据模式创建新的数据库。

(2) 允许用户利用查询语言查询和修改数据。

(3) 能够存储大量的数据，防止故障和非授权用户对数据的破坏，并可以为查询和修改提供有效的支持。

(4) 能够对多用户的并发数据访问进行有效控制，防止用户之间相互影响和并发访问对数据的破坏。

1.1 数据库技术的发展

数据库技术出现于 20 世纪 60 年代。在 40 年的发展过程中，数据库技术主要经历了网状数据库、层次数据库、关系数据库、新一代数据库技术等几个阶段。

1.1.1 网状数据库

网状数据库中数据的基本存储单位是记录，一个记录可以包含若干数据项。数据项可以是多值的或者复合的数据，前者称为向量，后者称为重复组。每个记录有一个惟一的标识——数据库码(DataBase Key, 简称 DBK)，它是在一个记录存入数据库时由系统自动赋予的。DBK 可以看成是记录的逻辑地址，可以用做记录的替身，或用于查找记录。在网状数据模型中，数据间的联系用系表示。网状数据库是系的集合，其存储结构归结为系的实现方法。最常用的方法是链式实现方式。网状数据库是一种导航式的数据库，用户在执行具体操作时不但需要说明做什么，而且还需要说明怎么做。例如，在查找语句中不但要指明查找对象，而且还要规定存取路径。

第一个网状数据库管理系统是由美国通用电气公司的 Bachman 等人在 1964 年开发成功的 IDS(Integrated Data Store)系统。这也是世界上第一个数据库管理系统，它奠定了网状数据库的基础，并在当时得到了广泛的认可和应用。1971 年，美国数据系统委员会(CODASYL)提出了著名的 DBTG 报告，此报告对网状数据模型和语言进行了定义，并在 1978 年和 1981 年又进行了修改和补充，因此，网状数据模型又称为 CODASYL 模型或 DBTG 模型。1984 年，美国国家标准协会(ANSI)提出了一个网状定义语言(Network Definition Language, 简称 NDL)的推荐标准。20 世纪 70 年代先后出现过大量的网状数据库管理系统的产品，比较著名的有 Honeywell 公司的 IDSII, Univac 公司(后来并入 Unisys 公司)的 DMS1100, HP 公司的 IMAGE 等。网状数据模型对于层次和非层次结构的事务都能够比较自然地模拟，在关系数据库出现之前，网状数据库管理系统比层次数据库管理系统应用得更为普遍。在数据库技术的发展史上，网状数据库占有重要的地位。

1.1.2 层次数据库

现实世界中的很多事物是按照层次关系组织起来的,所以紧随网状数据库之后出现的是层次型数据库系统。层次数据模型模拟了现实世界的层次组织关系,按照层次存取数据。其中最基本的数据关系是层次关系,它代表两个记录之间一对多的关系,也叫做双亲子女关系。一个数据库系统中有且仅有一个记录无双亲,称为根结点,其他记录有且仅有一个双亲。在层次模型中,从一个结点到其双亲的映射是惟一的,所以对于每一个记录(根结点除外)来说,只需指出它的双亲,就可以表示出层次模型的树状整体结构。

比较著名的层次数据库系统是 IBM 公司于 20 世纪 60 年代末推出的信息管理系统,即 IMS (Information Management System),它是 IBM 公司研制的最早的大型数据库产品。目前,它已经发展到了 IMS V6 版本,可以提供对集群、N 路数据共享、消息队列共享等先进特性的支持,并在 3W 连接、商务智能等应用中扮演着新的角色。

1.1.3 关系数据库

网状数据库和层次数据库解决了数据的集中和共享问题,但在数据独立性和抽象级别上仍有很大欠缺。用户在对这两种数据库进行存取操作时,必须明确数据的存储结构,具体指明存取路径。为了弥补这些不足之处,IBM 公司的研究员 Codd 博士于 1970 年提出了关系数据模型,以关系作为描述数据的基础。在之后的几年中,Codd 又发表了一系列文章,奠定了关系数据库的理论基础。

关系数据模型有严格的数学基础,抽象级别较高,而且简单清晰,便于理解和使用。关系数据模型是以集合论中的关系概念为基础发展起来的。在关系数据模型中,无论实体还是实体间的联系均由单一的结构类型——关系来表示。在实际的关系数据库中,关系也称为表,一个关系数据库由若干张表组成。

为了准确形象地表示关系数据库的准则,1974 年 IBM 公司的 Rey Boyce 和 Don Chamberlin 制订了简单的关键字语法,用以说明符合关系数据库准则的数据定义,即结构化查询语言——SQL (Structured Query Language)。SQL 语言是一种高度非过程化的语言,其功能包括查询、操作、定义和控制等,它只要求用户指出做什么而无需指出怎么做。SQL 语言作为数据库发展史上的一个里程碑,自诞生之日起便成了检验关系数据库的试金石,其每一次的突破都引导着关系数据库技术和产品不断地向前发展。特别是 20 世纪 80 年代以来,SQL 更成为了关系数据库语言的国际标准。

SQL 语言共有三个版本,包括基本 SQL 定义即 SQL-89、增强型标准 SQL-92 和目前最新的且正在完善中的 SQL3 标准。其中,SQL-89 标准定义了模式、数据操作和事务处理;SQL-92 标准又增加了模式操作、动态创建和执行以及网络环境支持等功能。SQL3 的主要优势在于对抽象数据类型的支持,为新一代的对象关系数据库提供标准。

SQL 的出现使得关系数据库产品成为当前数据库市场的主流,自 20 世纪 70 年代以来,先后共有 200 多种产品投放市场,占据 90% 以上的市场份额。其中较为著名的数据库厂商包括 IBM, Oracle, Sybase, Informix, CA 以及 Microsoft 等。

1.1.4 新一代数据库技术

关系数据库以其完备的理论基础、简洁的数据模型、透明的查询语言和方便的操作方法

等优点得到了众多用户的一致好评。但是,随着数据库系统的日益普及以及人们要求的不断提高,关系数据库也暴露出了一些局限性。首先,关系模型过于简单,不利于表达复杂的数据结构;其次,关系模型支持的数据类型有限,无法包容更多的数据类型。于是,关系数据库受到了来自诸多方面的严峻挑战,它已经无法适应现代信息系统应用开发的要求,这和当年它出现时为应用所带来的巨大方便和深远影响形成了鲜明的对比。如果说过去是数据库技术的发展带动了应用发展的话,那么今天则是应用反过来推动了数据库技术的进一步变革。毫无疑问,这些挑战来自面向对象技术、并行与分布处理技术、人工智能、网络技术、多媒体技术以及移动计算技术的飞速发展。

1.1.4.1 面向对象数据库

面向对象是一种认识、描述事物的方法论,它起源于程序设计语言,并迅速渗透到计算机科学的各个领域。面向对象方法以客观世界中存在的实体对象为基本元素,通过类和继承来表达事物之间具有的共性和关系,并采用一种比较直接的映射方式很好地实现了抽象、封装、复杂性控制、信息隐蔽等机制。

作为面向对象方法在数据库领域中的实现和应用,面向对象数据库技术现在仍处于不断发展和创新的阶段,在概念、原理和技术上都还没有达成广泛的共识。但是,人们普遍认为面向对象数据库的基本特征应该包括对象、类、继承、封装等。纯面向对象数据库系统可以完全实现面向对象。目前已经有一些成功的纯面向对象的数据库产品,比如 CA 的 Jasmine。对象关系数据库系统一般也支持 SQL 环境中的基本类型扩展、复杂对象处理、对象类属的继承和产生式规则的应用。另外,许多关系数据库产品也在其新版本中加入了面对面向对象的支持。

目前,数据库开发工具大体可以分为两类:通用工具(如 PowerBuilder)和专用工具(如 Oracle 的 Developer2000 和 Designer2000)。工具的使用大大提高了系统开发效率,并在一定程度上规范了人们的开发行为。这些工具大多都支持面向对象的开发方法和模式,这也是面向对象技术和关系技术的一种结合。

1.1.4.2 并行数据库

近年来,随着应用领域数据库规模的急剧膨胀,数据库工作的负荷日益加重,对数据库性能的要求也越来越高。目前,并行处理技术的迅速发展以及数据模型中潜在的并行性,使得利用并行处理技术提高数据库系统的性能成为可能。并行数据库就是数据库技术与并行处理技术相结合的产物,是在大规模并行计算机系统(简称 MPP)和机群并行计算环境的基础上建立的数据库系统。

目前,并行数据库系统的并行结构主要有:

- 全共享结构(Shared-Everything)(如图 1.1 所示),即处理机共享主存储器和磁盘存储器结构。处理机、共享主存储器和磁盘存储器由高速通信网络连接;处理机之间的通信可以通过主存储器间接实现,也可以通过高速通信网络直接进行。
- 共享主存储器结构(Shared-Memory)(如图 1.2 所示),即多处理机共享主存储器结构。每个处理机具有独立的磁盘存储器;多处理机和共享主存储器由高速通信网络连接;处理机之间的通信可以通过主存储器间接实现,也可以通过高速通信网络直接进行。
- 共享磁盘结构(Shared-Disk)(如图 1.3 所示),即多处理机共享所有磁盘存储器的结构。每个处理机具有独立的主存储器;处理机之间的通信可以通过磁盘存储器间接实现,

也可以通过高速通信网络直接进行。

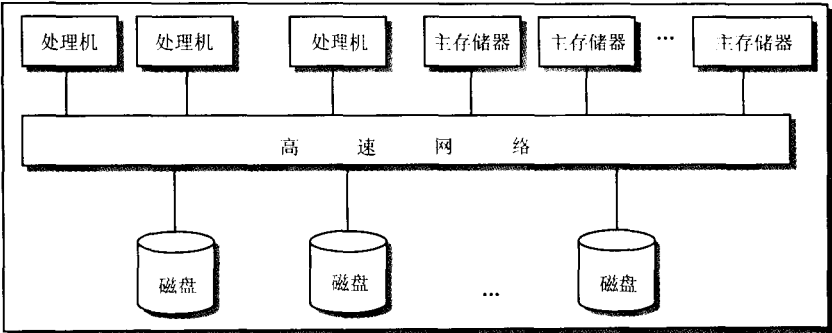


图 1.1 全共享结构

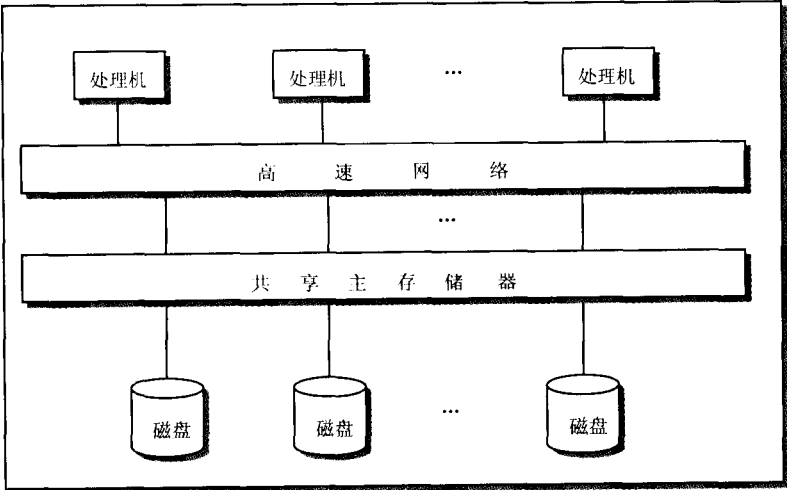


图 1.2 共享主存储器结构

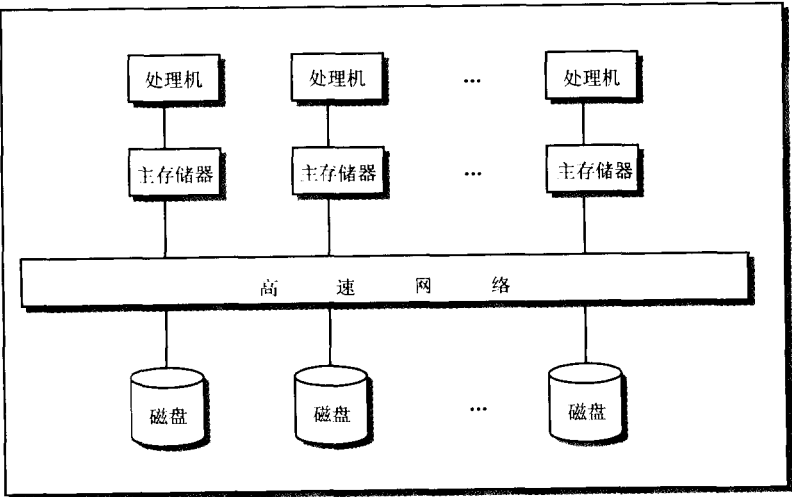


图 1.3 共享磁盘结构

• 无共享结构 (Shared-Nothing) (如图 1.4 所示), 即没有任何共享硬件资源的结构。每个处理机都具有独立的主存储器和磁盘存储器, 处理机之间的通信由高速通信网络实现。

• 分层并行结构 (如图 1.5 所示), 即融合了全共享结构和无共享结构的优点的一种结构。它有许多“超级结点 (Supernodes)” ; 这些超级结点由高速互联网连接; 每个超级结点内有多处理机共享主存储器和独立的磁盘存储器系统。

并行数据库系统是新兴的数据库研究领域, 尚无真正的并行数据库系统投入运行。目前并行数据库的原型系统有 ARBRE, BUBBA, GAMMA, XPRS 等。

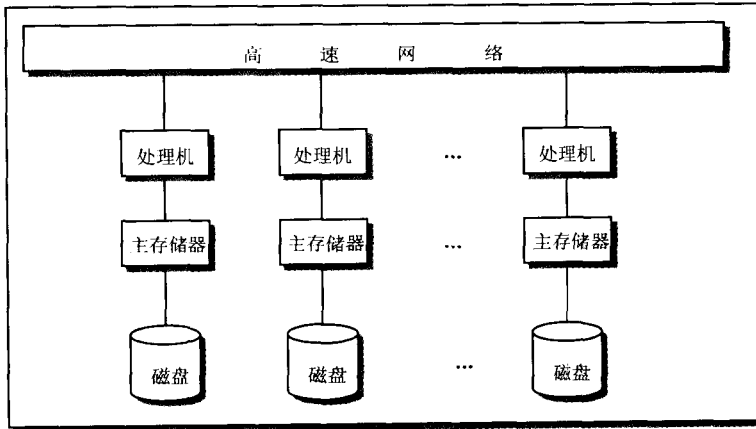


图 1.4 无共享结构

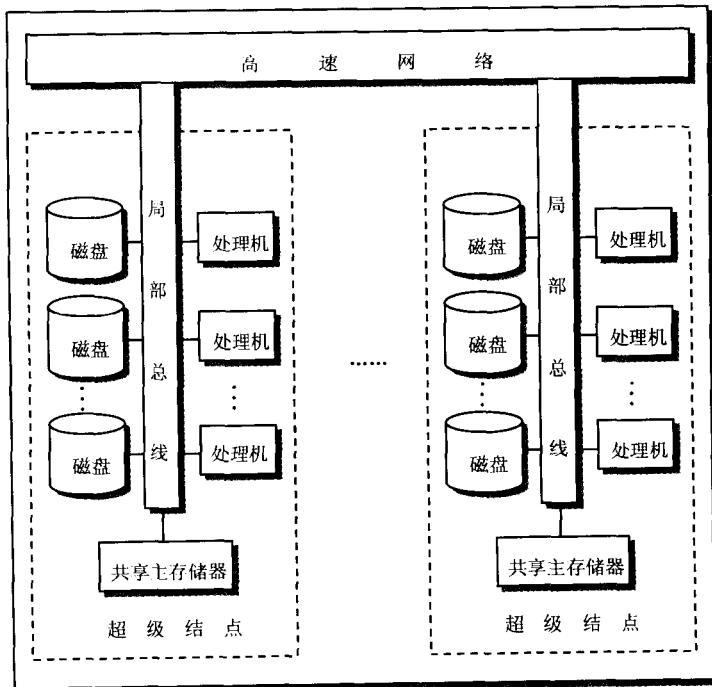


图 1.5 分层并行结构

1.1.4.3 分布式数据库

数据库技术与分布处理技术的结合, 导致出现分布式数据库。通俗地说, 分布式数据库系统是一组数据集, 它们逻辑上属于同一系统, 物理上分布在用网络连接的多个结点上。分布式数据库具有如下特点:

- 数据的物理分布性, 即数据库中的数据不集中存储在一台计算机上, 而是分别存储在多台计算机上;
- 数据的逻辑整体性, 即数据库虽然在物理上是分散的, 但数据不是互不相关的, 它们在逻辑上是相互联系的整体;
- 数据的分布透明性, 即在用户看来, 数据库仍然是一个集中的数据库, 分布的实现由系统完成;
- 结点的自主性, 即系统中的结点具有独立性, 能够执行局部的应用请求, 并通过网络处理全局的应用请求;
- 数据的冗余性, 即通过适当冗余来提高系统的效率和可靠性。

20 世纪 80 年代, 陆续出现了许多分布式数据库的原型系统, 比如, 美国国防部委托 CCA 公司设计并研制的 SDD-1 系统, 美国加利福尼亚大学伯克利分校研制的分布式 INGRES 系统, IBM 圣约瑟研究实验室设计开发的 R* 系统, 前西德斯图加特大学设计并研制的 POREL 系统等。从 20 世纪 90 年代开始, 主要的数据库厂商 (如 Oracle, Sybase, Informix 等) 对集中式数据库管理系统的核心加以改造, 逐步加入分布处理功能, 向分布式数据库管理系统发展。目前, 分布式数据库开始进入实用阶段。

1.1.4.4 主动数据库

在许多实际的数据库应用领域中, 如计算机集成制造、管理信息系统、办公自动化等应用, 常常希望数据库系统在紧急情况下能够根据数据库的当前状态, 主动适时地做出反应, 执行某些操作, 向用户提供有关信息。但是, 传统的数据库系统是被动的系统, 它只能被动地按照用户给出的明确指示执行相应的数据库操作, 很难适应这些应用需要主动的要求。因此, 把数据库技术与人工智能技术和面向对象技术相结合, 就出现了主动数据库。

主动数据库的主要目标是提供对紧急情况及时反应的能力, 同时提高数据库管理系统的模块化程度。主动数据库通常采用的方法是在传统数据库系统中嵌入事件—条件—动作 (简称 ECA) 规则, 在某一事件发生时引发数据库管理系统去检测数据库当前状态, 判断是否满足设定的条件, 如果条件满足, 则触发规定动作的执行。

主动数据库的实现途径主要有以下三种。

- 在原有数据库管理系统的基础上进行改造, 增加事件监视器。
 - 将一般程序设计语言改造成主动程序设计语言, 并将数据库操作嵌入其中。
 - 重新设计主动数据库程序设计语言, 将数据管理和操作与应用程序彻底融合。
- 上述三种实现途径, 系统效率依次提高, 但实现难度依次增大。

1.1.4.5 Internet 数据库

Internet 的迅猛发展对数据库技术产生了巨大而深远的影响。数据库技术通过与网络互相结合进而可以支持电子商务, 已经成为各大厂商竞争的焦点。目前几乎所有的大型数据库厂

家都可提供 Internet 环境下的数据库应用,甚至专门开发与 Internet 无缝衔接的版本,Oracle8i 就是一个非常典型的例子。以数据库技术和网络技术为支撑的电子商务,很有可能建立起新型的虚拟商业,乃至虚拟工业。对企业而言,电子商务不应仅仅是在 Internet 上创建本企业的网站,进行企业形象宣传或通过网络进行常规的商业操作,更应包括通过电子商务来改变企业传统的运作模式,加强企业信息的有效利用以及与客户和合作伙伴之间的沟通,进而辅助管理与决策。

1.1.4.6 多媒体数据库

多媒体数据库是以数字化和数据压缩为基础的多媒体技术与数据库技术相结合的产物,主要用于管理多媒体数据。多媒体数据具有对象复杂,存储分散和时空同步等特点。多媒体数据库管理系统的基本功能主要包括:

- 能够表示和处理多种媒体的数据。多媒体数据在计算机内的表示方法取决于各种媒体数据所固有的特性和关联。常规的格式化数据可以使用常规的字段表示;非格式化数据要根据系统需求确定表示方法。多媒体数据的管理方式和处理方法取决于数据的表示方法。
- 能够反映和管理各种媒体数据的特性,或各种媒体数据之间的空间或时间的关联。多媒体数据库管理系统除了要对多媒体数据的内容与结构建模之外,还应提供对各种媒体数据的特性和集成机制的时空关联的组织和管理方法。
- 满足物理数据独立性、逻辑数据独立性和媒体数据独立性。物理数据独立性是指当物理数据组织(存储模式)改变时,不影响概念数据组织(逻辑模式)。逻辑数据独立性是指概念数据组织改变时,不影响用户程序使用的视图(外模式)。媒体数据独立性是指在多媒体数据库管理系统的设计和实现时,要求系统能保持各种媒体的独立性和透明性,即用户的操作可最大限度地忽视各种媒体的差别,不受具体媒体的影响和约束,并要求系统不受媒体变换的影响,实现复杂数据的统一管理。
- 不仅具有传统数据库的数据操作功能,而且具有多媒体数据的搜索功能、浏览功能等。
- 具有网络功能,开放功能,以及事务和版本管理功能。

1.1.4.7 移动数据库

移动计算的兴起引发了数据库技术的又一次突破,移动数据库应运而生。简单地说,移动数据库就是支持移动计算环境的分布式数据库。它的目的就是有效地支持移动计算环境中的各种数据应用,满足人们能够在任意地点、任意时刻访问任意数据的需求。归纳起来,移动数据库主要能够实现以下四个目标。

- 可用性与可伸缩性:在避免系统不稳定性的同时,提供高可用性和可伸缩性;
- 移动性:允许移动计算机在和网络断开的情况下访问和更新数据库;
- 可串性:支持满足可串性的并发事务执行;
- 收敛性:使系统总能收敛于一致状态,从而避免出现混乱。

移动数据库对移动环境中的许多重要应用,诸如战场移动指挥、移动办公系统、公共信息发布等都具有重要的意义。目前的移动数据库一般利用掌上设备,通过有线或无线通信手段下载主机数据,再由自身的数据管理和处理系统来实现特定的应用,以满足移动办公、人员流动以及特殊环境的需求。

除了上述各种数据库以外,数据库技术还深入到了许多其他的应用领域,并与该领域中的技术互相结合,互相促进,比如与知识处理技术相结合形成了知识库技术,与模糊处理技术相结合形成了模糊数据库技术,与时序逻辑技术相结合形成了时序数据库技术,与辅助设计技术相结合形成了工程数据库技术,与统计学应用相结合形成了统计数据库技术,与图形学应用相结合形成了图形数据库以及空间数据库技术等。从某种意义上说,信息化社会就是数字化社会。它离不开对数字的控制和处理,当然也就离不开对数据库技术和产品的依赖,同时它也将驱动着数据库技术继续向前迈进。可以预见,未来的数据库应用在广度和深度上都将超过现有水平。

1.2 面向对象数据库

数据模型是现实世界的对象或实体,以及对象的约束和对象间联系的逻辑组织。使用了面向对象概念的数据模型称为面向对象数据模型。面向对象数据库(Object Oriented DataBase, 简称 OODB)是其行为和状态以及关系都是根据面向对象数据模型定义的对象对象的汇集。面向对象数据库系统(Object Oriented DataBase System, 简称 OODBS)是支持定义和操作 OODB 的数据库系统。面向对象的概念已经包括了传统数据库语言中的某些数据模型概念,比如把对象组合成类(对应于将记录组合成记录类型,将元组组合成关系),对象间的聚集关系(嵌套对象),对象类之间的泛化关系等。因此,面向对象的概念是新一代数据库应用所需数据模型的坚实基础。面向对象数据模型包含聚集和概括关系,意味着面向对象数据库系统为定义和操作对象间的联系提供了用户接口。应用程序员无需显式编程和管理这些联系。此外,面向对象数据模型能够有效地表示复杂对象,通过封装和信息隐藏概念可提供模块化机制,通过封装和继承概念可提供软件重用机制,通过滞后联编等概念可提供系统扩充能力。

1.2.1 面向对象数据库的基本特征

首先,面向对象数据库系统是一个数据库管理系统;其次,它是一个面向对象的系统,并与当前流行的面向对象程序设计语言一致。与传统数据库相比,以面向对象数据模型为核心的面向对象数据库具有以下基本特征。

(1) 对象

面向对象数据模型将客观世界模拟成由各个相互作用的称为对象的单元组成的复杂系统。对象的定义包括状态和行为两方面,状态由一组属性值组成,行为由一组方法组成。系统中所有对象的状态共同构成系统的状态。对象之间的相互作用可以改变对象的状态。对象之间的相互作用通过发送消息来实现。通常,消息由三个部分组成:接收对象、方法和方法参数。对象作为接收者可以执行的全部方法称为对象的接口(Interface)。接口描述了对象的行为特征,接口中只有方法的定义,没有方法的实现。接口是对象提供给外界的惟一的可见部分,只能通过接口对对象进行访问,这种性质称为封装性(Encapsulation),它实现了对象的内部细节对应用程序隐蔽的目的。

(2) 对象类

具有相同属性和方法的对象组成对象类。对象只属于某一个类并作为该类的实例。类是元类(Metaclass)的实例。面向对象数据模型允许同一方法有多个实现并存,称为多态(Polymorphism)。

(3) 继承

在面向对象数据模型中,定义子类时只要定义不同于超类的属性和方法,子类对象可以使用超类的属性和方法,这称为子类对超类的继承(Inheritance)。一般来说,子类继承超类的方法的定义和实现。但是,也可以只继承方法的定义,子类自己实现该方法,这称为重载(Overloading)。面向对象数据库系统有单继承和多继承之分。每个类至多只有一个直接超类的称为单继承系统,直接超类数多于一个的称为多继承系统。单继承系统中所有类的继承关系图呈树形,多继承系统中所有类的继承关系图是有向无环图。

(4) 持久性、对象标识

在大多数程序设计语言中,对象是临时的,随着进程的终止而消失。但是,在面向对象数据库中,对象在被删除前是永久存在的。一个对象的对象标识在系统中是惟一的,它独立于对象的状态和行为,在对象生成时由系统赋予,在整个生存期内是不变的。

(5) 并发控制

事务是对数据库进行读和写的一个序列。事务处理的目的是确保事务的原子性和可串行性。事务处理主要包括两个方面:并发控制和恢复。与关系数据库系统相比,面向对象数据库系统中的并发控制具有如下特点。

① 对面向对象数据库进行存取的最小逻辑单位是对象而不是类,所以面向对象数据库系统中的并发控制是对象级的。

② 类不仅是一个实例集,而且保持这些实例的各种聚合特性。所以,从加锁的角度来看,一个类对象应分裂成两个类实例对象:实例集对象和实例聚合对象,数据库系统对其中的一个对象加锁或对两个对象加锁均取决于用户请求。

③ 子类可以继承超类的属性和方法,超类的属性和方法的改变也会影响到其子类的属性和方法。因此,当一个事务存取一个类的实例时,另一个事务不能修改该类或其超类的任何定义。

(6) 阻抗失配

数据查询语言是由系统自选查询路径的非过程性语言。非过程性语言的面向集合的操作方式与高级程序设计语言的面向单个数据的操作方式之间会产生不协调现象,俗称阻抗失配。阻抗失配的根本原因在于数据库的数据模型和程序设计语言的不一致,因而对所有嵌入式数据库语言来说,阻抗失配是不可避免的。面向对象数据模型的有关对象的概念来自面向对象高级程序设计语言。因此,如果选择一种面向对象高级程序设计语言,扩充它的数据库功能,使其成为面向对象数据库语言,则从根本上解决了阻抗失配问题。

1.2.2 面向对象数据库的标准化

1.2.2.1 标准化工作回顾

20世纪80年代中期以来,许多研究机构、大学和软件厂商都在进行面向对象数据库系统的研制工作。20世纪90年代初,出现了几个面向对象数据库系统的研究原型,发展至今已经有许多商业性的面向对象数据库系统了。由于缺乏统一的标准,它们应用的推广受到了影响,从而阻碍了面向对象数据库的发展。近年来,面向对象数据库的用户和厂商都强烈地意识到面向对象数据库标准的重要性。为此,国际标准化组织以及一些联合体开始了面向对象数据库的标准化工作,并已出台了几个国际标准。

• OODBTG 最终报告

1989年1月,美国ANSI所属的ASC/X3/SPARC数据库系统研究组(简称DBSSG)成立了面向对象数据库任务组(简称OODBTG)。到1991年8月,该任务组完成了“OODBTG最终报告”,提出了对象信息管理(简称OIM)标准的建议,其中一个重要的内容是“对象数据管理(简称ODM)参考模型”。ODM参考模型是一个继承了对象、数据库和系统三者特征的功能框架结构模型。它描述了对象模型的一般特征,数据管理的特征,以及对象数据管理应具备的系统特征,从而勾画出一个面向对象数据库管理系统的基本功能规范。

• SQL3

在SQL92(即SQL语言标准的92文本)之后,ANSI/X3H2与ISO/IEC/JTC1/SC21/WG3于1993年共同研究提出了用以替换SQL92标准的工作文本SQL3。SQL3的一个很重要的特征是能定义抽象数据类型(简称ADT),引入了对象数据的处理能力。SQL3支持值抽象数据类型、对象抽象数据类型、子类和超类、数据类型生成器以及类型模板。目前,正在研究SQL3所能提供的对象数据处理的能力。毫无疑问,SQL语言要增加支持对象数据处理的功能。

• 对象数据库标准: ODMG

对象管理集团(简称OMG)是美国企业社团为促进研究和应用面向对象的概念和方法的实体。ODMG系列标准是OMG所属的对象数据库管理组(简称ODMG)承担的对象数据库管理系统(简称ODBMS)标准研究的工作成果,其目标是制订访问对象数据的统一标准。ODMG于1993年、1997年和2000年相继推出了ODMG 1.0标准、ODMG 2.0标准和ODMG 3.0标准。ODMG标准定义的ODBMS集成了数据库功能和面向对象程序设计语言功能。它规定了对象模型、对象定义语言、对象查询语言,以及与C++, Smalltalk语言的联编。目前参与ODMG标准的活动的厂商有: SunSoft, Object Design, Ontos, O2 Technology, Versant, Objectivity, Hewlett-Packard, Poet Software, Itasca, Intellitic, Digital Equipment Corporation, Servio, Texas Instruments等。

1.2.2.2 ODMG 1.0

ODMG 1.0标准的目的是为了let ODBMS的用户编写的可移植的应用,能运行在多个ODBMS的产品上,即数据模型、程序设计语言联编、数据操作和查询语言必须是可移植的。它定义的ODBMS可以使数据库对象作为程序设计语言对象出现在一个或多个现有的程序设计语言中。ODBMS用透明的持久数据、并发控制、数据恢复、关联查询和数据库的其他功能扩展了语言。

ODMG 1.0规定的ODBMS的主要组成包括以下几个部分。

• 对象模型

它是ODBMS支持的公用数据模型,以OMG的对象模型为基础。在ODMG对象模型中,基本的建模原语是对象,对象可以从属于各种类型(Type),同一类型的所有对象呈现出共同的行为和状态;对象的行为通过操作集定义,对象的状态通过作为它的性质(Property)集合的值来定义,其性质可以是对象本身的属性(Attribute),也可以是该对象与一个或多个其他对象之间的关系(Relationship)。类型有一个接口(Interface)和一个或多个实现,类(Class)有一个接口和一个实现。

• 对象定义语言 ODL

ODL用于定义符合ODMG对象模型的对象类型的接口。ODMG提出的开发ODL的指