

高等学校教材

数据库原理及应用 (ORACLE)

刘甫迎 邓礼清 蒋锡民 主编



重庆大学出版社

内 容 简 介

数据库的广泛应用,使它在当今信息时代扮演了无可替代的重要角色,越来越多的人希望学习和了解数据库。

本书有十一章和两个附录,详细介绍了数据库的意义,数据库的由来和发展,数据模型,实体联系模型(E-R),网状、层次及面向对象数据库,关系数据库,函数依赖、范式,数据库的设计与维护,数据库的安全性与完整性、并发控制与恢复,分布式数据库等数据库的基本概念、原理和理论。并打破了其它数据库原理的书忌讳写某一具体的数据库的惯例,本书叙述了客户/服务器结构后端大型数据库管理系统的首选——ORACLE,且将之作为上述基本理论的具体例子贯穿全书。特别是书中还著述了ORACLE的SQL、PL/SQL和ORACLE的开发工具Forms、Report、Graphics等内容,使本书同时又成了一本学习ORACLE的教科书。

本书可作为高等院校的教材,也适于从事数据库软件开发和应用的人员参考。

高等学校教材

数据库原理及应用(ORACLE)

刘甫迎 邓礼清 蒋锡民 主编

责任编辑 曾令维

*

重庆大学出版社出版发行

新华书店经销

重庆建筑大学印刷厂印刷

*

开本:787×1092 1/16 印张:18 字数:449千

1998年8月第1版 1998年8月第1次印刷

印数:1—4000

ISBN 7-5624-1826-8/TP·186 定价:26.00元

前 言

诞生于本世纪中叶的计算机科学较之其它现代科学技术的发展更迅速,在世纪之交到来之际,它几乎可以被称为“知识爆炸”了。21 世纪是信息和知识的社会,如何组织和利用这些庞大的信息和知识已成为衡量一个国家科学技术水平高低的重要标志。

早在 60 年代,数据库作为现代信息系统基础的一门软件学科便应运而生了。现在,数据库技术已成为计算机领域中最重要技术之一,它是软件学科中一个独立的分支。它的出现使得计算机应用渗透到工农业生产、商业、行政、教育、科学研究、工程技术和国防军事的各个部门。管理信息系统(MIS)、办公自动化系统(OA)、决策支持系统等都使用了数据库管理系统或数据库技术的计算机应用系统。越来越多的人希望学习和了解数据库。

ORACLE 是世界上第一个以 SQL 语言为基础的关系数据库。从 1979 年它的问世,到 1997 年 ORACLE8 的发表,美国的 ORACLE 公司一直致力于信息管理现代化技术及产品的研究与开发,使 ORACLE 在数据库技术领域创造了无数的第一,它的销量和普及程度名列世界前茅。ORACLE 实际上成了大型数据库管理系统的工业标准。

本书有十一章和两个附录,详细介绍了数据库的意义,数据库的由来和发展,数据模型,实体联系模型(E-R),网状、层次及面向对象数据库,关系数据库,函数依赖、范式,数据库的设计与维护,数据库的安全性、完整性、并发控制与恢复,分布式数据库等数据库的基本概念、原理和理论。并打破了其它数据库原理的书忌讳写某一具体的数据库的惯例,本书叙述了客户/服务器结构后端大型数据库管理系统的首选——ORACLE,且将之作为上述基本理论的具体例子贯穿全书。特别是书中还著述了 ORACLE 的 SQL、PL/SQL 和 ORACLE 的开发工具 Forms、Report、Graphics 等内容(当然前端数据库管理系统也可选 Visual FoxPro,详见重庆大学出版社出版的刘甫迎等主编的《Visual FoxPro 中文版教程》),使本书同时又成了一本学习 ORACLE 的实用教科书。

本书可作为高等院校的教材,也适于从事数据库软件开发和应用的人员参考。

本书由刘甫迎、邓礼清、蒋锡民主编。刘甫迎编著了第五章、第七章、第九章、第十章;邓礼清编著了第二章、第三章、第四章(不含 4.6 节)、第六章、附录 A;蒋锡民编著了第一章、第八章、第十一章、4.6 节、附录 B。全书由刘甫迎统稿。在编著和出版的过程中郭洁、焦根昌、党晋蓉等同志做了不少工作,给予了很大的帮助,在此一并表示感谢!

由于水平有限,错误难免,请指正。

刘甫迎

目 录

第一章 数据库导论.....	1
1.1 数据库的意义	1
1.2 数据库的由来和发展	2
1.3 数据库的体系结构	9
1.4 数据库系统	13
1.5 数据库管理系统	15
习题	18
第二章 实体联系模型 (Entity—Relationship Model)	19
2.1 实体和实体集合	19
2.2 联系和联系集合	20
2.3 属性、映射限制和关键字	21
2.4 实体联系 E-R 图解和将之归纳为表	24
2.5 概括和聚集	29
2.6 E-R 数据库模式设计	30
习题	32
第三章 网状、层次及面向对象数据库	33
3.1 网状数据库 DBTG	33
3.2 层次数据库 IMS	37
3.3 面向对象模型	43
习题	50
第四章 关系数据库	51
4.1 关系模型和基本概念	51
4.2 关系代数	53
4.3 关系演算	58
4.4 关系查询语言: ISBL、QUEL 和 QBE	61
4.5 关系数据库标准语言——SQL	65
4.6 ORACLE 关系数据库系统	68
习题	81
第五章 ORACLE SQL 和 PL/SQL	83
5.1 ORACLE SQL 介绍	83
5.2 定义、修改、删除表 (Table)	85
5.3 模式对象、直接量、函数和表达式	94
5.4 数据操纵语言 (Insert、Delete 和 Update)	103
5.5 视图	104
5.6 ORACLE PL/SQL	107
习题	122
第六章 关系数据库设计理论	125

6.1 引言	125
6.2 函数依赖	126
6.3 范式	128
6.4 多值依赖和4NF	131
习题	133
第七章 数据库的设计与维护	134
7.1 概述	134
7.2 需求分析	136
7.3 概念结构设计	138
7.4 逻辑结构设计	143
7.5 物理设计	146
7.6 实现与维护	148
7.7 用 ORACLE 建立数据库	150
7.8 ORACLE 数据库和实例的启动及关闭(ORACLE 的 DBA)	168
7.9 数据字典(Data Dictionary)	175
习题	177
第八章 数据库保护	178
8.1 数据库的安全性(用户鉴别、特权、角色、审计)	178
8.2 数据完整性(数据库触发器)	188
8.3 并发控制	192
8.4 数据库后备和恢复	196
习题	201
第九章 分布式数据库系统	202
9.1 概述(客户/服务器结构、服务器—服务器结构)	202
9.2 分布式数据库的连接	205
9.3 分布式查询处理及其它	209
9.4 事务管理	210
9.5 多协议信息交换	212
9.6 表快照与复制	220
习题	221
第十章 数据库应用程序开发工具 ORACLE Developer/2000	223
10.1 ORACLE Forms 工具	223
10.2 ORACLE Report	244
10.3 ORACLE Graphics	254
习题	260
第十一章 ORACLE 程序实例——图书管理信息系统	262
习题	271
附录 A 《数据库原理及应用(ORACLE)》教学大纲	272
附录 B 实验指导书	275
主要参考文献	280

第一章 数据库导论

1.1 数据库的意义

诞生于本世纪中叶的计算机科学较之其它现代科学技术的发展更迅速,在世纪之交到来之际,它几乎可以被称为“知识爆炸”了。21 世纪是信息和知识的社会,如何组织和利用这些庞大的信息和知识已成为衡量一个国家科学技术水平高低的重要标志。

早在 60 年代,数据库技术作为现代信息系统基础的一门软件学科便应运而生了。现在,数据库技术已成为计算机领域中最重要技术之一,它是软件学科中一个独立的分支。它的出现使得计算机应用渗透到工农业生产、商业、行政、教育、科学研究、工程技术和国防军事的各个部门。管理信息系统(MIS)、办公自动化系统(OA)、决策支持系统等都使用了数据库管理系统或数据库技术的计算机应用系统。

数据库(Database,即 DB)是存储在一起的相关数据的集合,是存储数据的“仓库”。因此,要理解数据库就需要先了解在数据处理领域中常遇到的两个基本概念:“信息”(information)和“数据”(data)。

信息是关于现实世界事物的存在方式或运动状态的反映的组合。例如,上课用的黑板,它的颜色是黑的,形状是矩形,尺寸是长 3.2m,高 1.4m,材料是木材,这些都是关于黑板的信息,都是关于黑板的存在状态的反映,从不同角度“反映”或“刻画”了黑板这个事物。信息源于物质和能量,一切事物,包括自然和人类都产生信息,信息是物质和能量形态的反映,它不可能脱离物质而存在。信息传递需要物质载体,信息的获取和传递要消耗能量。信息是可以感知的,信息可以存储,并且可以加工、传递和再生。电子计算机是信息处理领域中最先进的工具之一,人类对收集到的信息可以进行取舍整理。几乎和信息同样被广泛使用的另一个概念是“数据”。所谓数据,通常指用符号记录下来的可加以鉴别的信息。例如,为了描述黑板的信息,可以用一组数据“黑色、矩形、3.2m×1.4m”来表示,由于“黑色”、“矩形”、“3.2”、“m”……这些符号已经被人们赋予了特定的语义,所以它们就具有了传递信息功能。

从上面的例子中,可以看到信息和数据之间的固有联系:数据是信息的符号表示或称为载体,信息则是数据的内涵,是对数据的语义解释。但另一方面,某一具体信息与表示它的数据的这种对应关系又因环境而异。同一信息可能有不同的符号表示,同一数据也可能有不同的解释。数据处理领域中的数据概念较之科学计算领域中数据概念已经大大地拓宽了。定义中所说的符号,不仅包含数字符号,而且包含文字、图像和其它符号;而所谓“记录下来”也不仅是指用笔写在纸上,它还包括磁记录、光刻等各种记录形式。

由于信息是现实世界中事物的存在方式和运动状态的反映,而现实世界的事物常常是相互关联的,这就使得人们在了解、掌握事物之间的固有联系和运动规律的基础上,可以从一些已知的信息出发,经过演绎推理,导出新的信息,为人类社会生活的各种需要服务,这就是常说的“信息处理”。例如,根据上述黑板的尺寸数据和木工定额标准,可以推算出制作黑板所需的

木材数量和制作费用。

一般,人们将原始信息表示成源数据,然后对这些源数据进行综合推导加工,得出新的数据。这些结果数据表示了新的信息,可以作为某种决策的依据(或用于新的推导加工)。这个过程通常称为“数据处理”。

电子计算机使大规模的数据处理成为可能,它和通信、网络技术的发展一起,进一步推动了信息处理和利用社会化,极大地增强了人类社会信息处理能力。

数据库系统就是实现有组织地、动态地存储大量相关数据,方便用户访问的计算机软硬件资源组成的系统。而数据库技术是研究数据库的结构、存储、设计和使用的一门软件学科。因此,数据库技术主要是研究如何存储、使用和管理数据。在计算机应用中,数据处理占的比重最大,而数据库系统是数据处理的核心机构,所以它的效能往往决定了整个计算机应用的经济效益。

数据库离不开数据模型。数据模型是对现实世界客观事物及其联系的描述,它反映数据项之间和记录之间的联系,在数据库技术中使用模型的概念描述数据库的结构与语义。常用的三种数据模型是:层次模型(hierarchical model)、网状模型(network model)和关系模型(relational model)。此外,还有面向对象模型(object-oriented model)等(数据模型,在后面章节将详细叙述)。

数据库这门学科与其它基础软件、系统软件、应用软件有着密切的联系。例如操作系统,数据库技术是在操作系统的文件系统的基础上发展起来的,而且数据库系统本身就是在操作系统(例如 Windows)支持下才能工作。数据库与网络技术和多媒体技术的关系也很密切,分布式数据库要用上网络,数据库甚至可在国际互联网(Internet)上交换海外的数据、声音、图像、图片等多媒体信息。数据库技术与数据结构的关系也是不可分的,数据库技术不仅要用到数据结构的知识,而且丰富了数据结构的内容。程序设计是使用数据库系统的最基本方式,因为数据库中大量的应用程序过程大都是用高级语言加上数据库的操纵语言编写的。集合论、数理逻辑是关系数据库的理论基础,其很多概念、术语、思想都直接用到关系数据库中。

1.2 数据库的由来和发展

数据库这个名词起源于 50 年代,当时美国为了战争的需要,把各种情报集中在一起,存入计算机,称为 Information Base 或 Database。1963 年美国 Honeywell 公司的 IDS(Integrated Data Store)系统投入运行,揭开了数据库技术的序幕。1965 年美国利用数据库帮助设计了阿波罗登月火箭,推动了数据库技术的产生。当时社会上产生了许多形形色色的 Database 或 Databank,但基本上都是文件系统的扩充。1968 年美国 IBM 公司推出了层次模型的 IMS 数据库系统,并于 1969 年形成产品。1969 年,提出了 COBOL 语言的美国 CODASYL(Conference on Data System Language,数据系统语言协会)组织的数据库任务组(DBTG)发表了网状数据库系统的标准文本(1971 年正式通过)。1970 年初,IBM 公司的高级研究员 E.F.Codd 发表论文提出了关系模型,奠定了关系数据库的理论基础。

70 年代是数据库蓬勃发展的年代。网状系统和层次系统占领了市场,关系系统开始处于实验阶段,IBM 公司研制出了原型关系语言 System R。1979 年关系软件(Relational Software)公司推出了第一个基于 SQL 的商用关系数据库产品 ORACLE。

80年代起,关系数据库产品已相当成熟,取代了网状系统和层次系统的市场。同时关系数据库理论也日趋完善,走向更高级的阶段,有了分布式数据库系统(Distributed database systems)等。后来,从不同的计算机应用领域提出了许多数据库的非传统应用课题,诸如多媒体数据、空间数据、时序数据、科学数据、复杂对象、知识、超文本管理等。为了适应这类应用的需要,提出了不少新的概念、新的数据模型和系统结构。经过几年的研究和实践,逐步形成了面向对象数据库系统(Object Oriented Database Systems)、主动数据库系统(Active Database Systems)、大型知识库系统(Large Knowledge Base Systems)、数据库中的知识发现(Knowledge Discovery in Database)、科学数据库(Science Database)等热点。下个世纪数据库技术必将获得更加长足的发展。了解数据库的由来和发展的历史以及数据管理技术各阶段的特点,对学习好数据库显然十分必要。

综观数据管理技术的发展可知:它与硬件(主要是外部存储器)、软件、计算机应用的范围有密切关系。数据管理技术的发展大致经过以下四个阶段:人工管理阶段、文件系统阶段、数据库阶段和高级数据库阶段。

一、人工管理阶段(50年代中期以前)

这一阶段的计算机主要用于科学计算。硬件中的外存只有卡片、纸带等。软件只有汇编语言,没有数据管理方面的软件。数据处理的方式基本上是批处理。这个时期的数据管理特点如下:

1. 数据不保存

进行某一课题计算时将原始数据随程序一起输入主存,运算处理后将结果数据输出。任务完成后,数据空间同程序空间一起释放。

2. 没有专用软件对数据进行管理

每个应用程序要包括存储结构、存取方法、输入输出方式等,数据与程序不具有独立性,一旦存储结构改变,就必须由应用程序员修改程序。由于程序直接面向存储结构,因此不存在逻辑结构与物理结构的区别。

3. 只有程序(PROGRAM)的概念、没有文件(FILE)的概念

即使有文件,也大多是顺序文件,其它组织方式必须由程序员自行设计与安排。

4. 数据面向应用

即一组数据对应于一个程序。由于各应用程序处理的数据不会毫无联系,程序之间会有重复。

5. 对数据的存取以记录为单位

其灵活性差。

二、文件系统阶段(50年代后期至60年代中后期)

这一阶段的计算机不仅用于科学计算,还大量用于信息管理。外存已有磁盘、磁鼓等直接存储设备。软件方面出现了高级语言和操作系统。操作系统中的文件系统(有时也称为“信息处理模块”)是专门处理外存的数据管理软件。处理数据方式有批处理,也有联机实时处理。这一阶段数据管理情况如下:

1. 特点

(1) 数据可长期保存在外存的磁盘上。用户随时通过程序对文件进行查询、修改和删除等处理。由于计算转向管理,数据处理的工作量增大。

(2) 数据的物理结构与逻辑结构有区别,但较简单。程序与设备之间有设备独立性(程序只需用文件名与数据打交道,不必关心数据的物理位置),由操作系统的文件系统提供存取方法(读/写)。由存取方法实现数据的逻辑结构与物理结构之间的转换。

(3) 文件的形式已多样化,有索引文件、链接文件和直接存取等,因而对文件的记录可顺序访问,也可随机访问。但文件之间是独立的,联系要通过程序去构造,文件的共享性差。

(4) 有了存储文件以后,数据不再仅仅属于某个特定的程序,而可以重复使用。但文件结构的设计仍然是基于特定的用途,程序仍然是基于特定的物理结构和存取方法编制的,因此,数据结构与程序之间的依赖关系并未根本改变。

(5) 对数据的存取基本上还是以记录为单位。

2. 缺陷

在文件系统中,改变存储设备,不必改变应用程序。虽然文件系统提供了存取方法,但这只是初级的数据管理。这种文件系统,还未能彻底体现用户观点下的数据逻辑结构独立于数据在外存的物理结构要求。因此,数据的物理结构修改时,仍然需要修改用户的应用程序。

文件系统有三大缺陷:

(1) 数据冗余性(redundancy),由于文件之间缺乏联系,造成每个应用程序都有对应的文件,有可能同样的数据在多个文件中重复存储;

(2) 不一致性(inconsistency),这往往是由数据冗余造成的,在进行更新操作时,稍不谨慎,就可能同样的数据在不同的文件中不一样;

(3) 数据联系弱(poor data relationship),这是文件之间独立、缺乏联系造成的。

由于这些原因,促使人们研究一种新的数据管理技术,这就是 60 年代末产生的数据库技术。

三、数据库阶段(60 年代末开始)

60 年代末,磁盘技术取得了重大进展,大容量(数百兆字节以上)和快速存取的磁盘陆续进入市场,成本有了很大的下降,为数据库技术的实现提供了物质条件。

60 年代中期,出现的大多数系统(database 或 databank)还不能称为真正的数据库系统。数据管理技术进入数据库阶段的标志是前面讲到的 60 年代后期的三大事件:即 1969 年 IBM 公司推出的 IMS 产品(层次数据库系统)和 CODASYL 研究与建议的 DBTG 系统(网状数据库系统),以及 1970 年起,IBM 公司 E. F. Codd 连续发表一系列论文,奠定了关系数据库理论基础。

70 年代以来,数据库技术得到迅速发展,并投入使用。数据库系统阶段根据其支持的数据模型(data model),到目前为止经历了三代:从上面讲到的层次网状代,到关系模型代和后面高级数据库阶段谈到的面向对象代。关系模型代从 70 年代初 E. F. Codd 奠定了关系数据库理论基础后,70 年代末推出了一些试验系统,80 年代初出现了一批商品化的关系数据库系统,如 ORACLE、SQL/DS、DB2、INGRES、INFORMIX、UNIFY 和 dBASE 等。SQL 语言在 1986 年被美 ANSI 和国际标准化组织(ISO)采纳为关系数据库语言的国际标准。

与文件系统相比,数据库系统克服了文件系统的缺陷,提供了对数据更高级更有效的管

理。概括起来,数据库技术的管理方式具有以下特点:

1. 采用复杂的数据模型(结构)

数据模型不仅要描述数据本身的特点,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的根本区别。这样数据不再面向特定的某个或多个应用,而是面向整个应用系统。数据冗余明显减少,实现了数据共享。

2. 有较高的数据独立性

在数据库系统中,系统提供映像的功能,确保应用程序对数据结构和存取方法有较高的独立性。数据的物理结构和逻辑结构差别可以很大。用户以简单的逻辑结构操作数据而无需考虑数据的物理结构。数据库结构分成用户的逻辑结构、整体逻辑结构和物理结构(图 1.1)。用户(应用程序或终端用户)的数据和外存中的数据之间的转换由数据库管理系统实现。为提高效率、减少冗余或增加新的数据,常需改变数据结构。在改变物理结构时,不影响整体逻辑结构、用户的逻辑结构以及应用程序,这样就认为数据库达到了物理数据独立性。在改变整体逻辑时,不影响用户的逻辑结构以及应用程序,这样就认为数据库达到了逻辑数据独立性。

3. 数据库系统为用户提供了方便的用户接口

用户可使用查询语言或简单的终端命令操作数据库,也可以使用程序方式(用高级语言如 C、FORTRAN 等语言和数据库操纵语言编制的程序)操作数据库。

4. 提供下述四方面的数据控制功能

(1)数据完整性:保证数据库始终包含正确的数据。用户可设计一些完整性规则以确保数据值的正确性。例如可把数据值限制在某个范围内,并对数据值之间的联系进行各种检验。

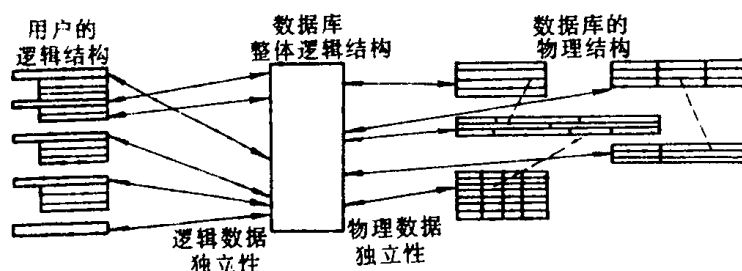


图 1.1 数据库系统的结构

(2)数据安全性:保证数据的安全和机密,防止数据丢失或被窃取。

(3)数据库的并发控制:避免并发程序之间的相互干扰,防止数据库数据被破坏,杜绝提供给用户不正确的数据。

(4)数据的恢复:在数据库被破坏时或数据不可靠时,系统有能力把数据库恢复到最近某个时刻的正确状态。

5. 对数据库的操作除了以记录为单位外,还可以数据项为单位

数据库阶段的程序和数据的联系可用图 1.2 表示。

综上所述,数据库可以定义为:一个存储起来为某个特定组织的多种应用服务并具有尽可能小的冗余度的互相关联的数据集合。其数据结构独立于使用数据的程序,对数据的增添、删

除、修改和检索,由系统进行统一的控制,而且数据模型也有利于将来应用的发展。

从文件系统发展到数据库技术是信息处理领域的一个重大变化。在文件系统阶段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;而在数据库方式下,数据开始占据了中心位置,数据的结构设计成为信息系统首先关心的问题,而利用这些程序的应用程序设计则退居到以既定的数据结构为基础的外围地位。

目前在国内外数据库应用已相当普及,各行业都建立了以数据库技术为基础的大型计算机网络系统,并在国际互联网(Internet)的基础上建立了国际性联机检索系统,其应用深入到人类社会生活的各个领域,甚至家庭。

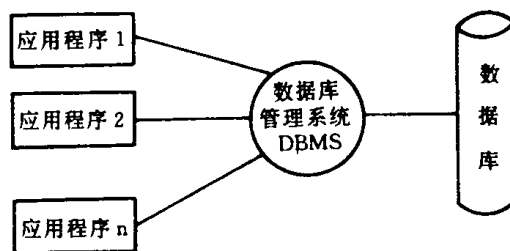


图 1.2 程序和数据的联系

四、高级数据库阶段(70 年代后期开始)

这一阶段的主要标志是分布式数据库系统、面向对象数据库、智能数据库系统的出现。

1. 分布数据库系统(distributed database systems)

集中式数据库把数据集中在一个数据库中,集中管理,减少了数据冗余和不一致性,而且数据联系比文件系统强得多。但集中式系统也有弱点,如其系统庞大、通信拥挤等。为了真正实现对分布在不同地方的数据资源共享,早在 70 年代就开始了分布式数据库的研究。经过 10 多年的努力,到 1986 年在软件市场上开始出现了分布式数据库产品。当时,关系技术公司(现为 INGRES 公司)宣布了称为 INGRES/STAR 的分布式版本。其后不久,ORACLE 公司也宣布了称为 SQL * STAR 的 ORACLE 分布式版本。这些年来,分布式数据库的研究和应用有了很大的进展。

(1) 多数据库系统

多数据库系统(MBS)是为在多个数据库之间实现互操作,解决数据资源共享的一种技术途径。多数据库系统是由若干数据库组成的一个集合,其中每个数据库称为分数据库。分数据库系统可以是集中式的,或是分布式的。它们都受各自的、可能是不同的 DBMS(数据库管理系统)管理。如果所有的分数据库系统都是相同的 DBMS,则称其为同构型的 MBS,否则称为异构型 MBS。MBS 不考虑全局模式,所有分数据库系统都是自治的,即数据定义自治,数据复制自治,以及在逻辑和物理级上的数据重构自治等。对 MBS 来说,MBS 语言很重要,它应当提供数据库语言和允许分数据库实现互操作的全部功能,支持用户非过程的操作。SYBASE 和 ORACLE 都提供分布式处理的功能,支持分布多数据库系统。

SYBASE 采用客户/服务器的体系结构,支持在网络环境下应用的分布计算模式。客户部分包括 Client 应用程序开发工具和 Client 接口。Client 工具为应用软件开发的各个阶段提供支持,Client 接口实现 Client 与 Server 间的连接,以及 OPEN Client 的开放互连。Server 分为 SQL Server 和 Open Server 两部分。SQL Server 完成分布式 RDBMS 的功能,Open Server 完成与其它非 SQL Server 管理的数据源和各种事务处理应用系统的互连。

ORACLE 主要通过 SQL * Net 和它的 RDBMS 实现分布式处理。它实现是多点分布式

查询,即一个查询可以涉及多个结点的数据库。ORACLE 对多种平台、多种网络、多种操作系统的支持能力很强,特别在 ORACLE 环境内,它提供的 4GL 工具具有比较强的分布查询能力。1992 年 ORACLE 公司发表了版本 ORACLE7。该产品的特点是采用协调服务器技术,提高分布检索功能和数据复制功能。

(2) 联邦数据库系统

联邦数据库系统是多个协作数据库系统的集合,其中每个数据库系统是自治的,可能是异构的。组成的各个数据库系统可以在不同程度上集成。控制和协调操纵组成数据库系统的软件称为联邦数据库管理系统(FDBMS)。联邦数据库系统的特点是分布、异构和自治。联邦数据库系统的体系结构一般采用五级模式:

- 局域模式
- 组成模式
- 联邦模式
- 输出模式
- 外模式

联邦数据已愈来愈引起数据库界的重视。它可以利用分布式数据库的技术,但与分布式数据库又不完全相同。经过多年研究,分布式数据库在学术上已很完美,但实现技术复杂,开销太大,商品化比较困难。在实际系统和应用需求方面,人们对分布环境下各结点之间的联系并不要求像分布式数据库那样紧密,逻辑上成为一个整体,而联邦式的联系更加实际可行。

2. 主动数据库系统

主动数据库是最近出现的一个重要而活跃的研究领域,属于智能数据库的一种。主动数据库因有自动触发执行一些系统或用户预定义操作序列的功能而得名。相应地传统的数据库系统可称为被动数据库系统。目前大多数工作是旨在将产生式规则嵌入数据库系统,并加入自动根据情景触发执行动作的机制。目前,一些原型系统正在被建造之中。在这些原形系统中,加入数据库的规则一般具有如下形式:

```
WHEN<事件>
IF<条件>
THEN<动作>{,IF<条件>THEN<动作>}
```

表示当<事件>发生时,将触发执行后面跟着的一条(或一组)规则。IF—THEN 规则(或规则组)的执行则按产生式系统的控制方式进行。可见,这种规则仅比一般产生式规则多了一个触发事件,用以控制何时触发执行。

由于产生式规则提供了一种表示触发、警报、断言、完整性约束、存取约束、诱导数据以及按快照(snapshots)等的统一机制,所以上述嵌入了产生式规则的这种主动数据库系统可以有十分广泛的应用。例如,可用它来实现传统的完整性约束、一致性存取约束等,还可以用它来实现诸如“例外处理”、错误自动处置,以及各种自动警报功能。当仓库存货低于某阈限时就自动发出适量进货的报告。此外,规则还可用来支持数据库的演绎推理功能,以及使大系统开发的合作更加方便等,因为可实现用户定义的资源共享策略、继承机制、版本控制、配置控制和工作流控制等。

主动数据库管理系统的特点是:

- (1) 规则是通过事件或数据触发执行,而不是识别一执行周期;

- (2) 采用规则作为完整性、存取控制、视图变换、触发等的统一机制；
- (3) 在数据库模式中定义规则存储在数据库中，由数据库管理系统执行，采用查询优化的技术；
- (4) 采用数据库管理系统中并发控制和恢复。

目前建成的模型系统有 ETM、HiPAC、Postgres、CPLEX、Starbust、RUBIS、O2、ATM、ODE、Alert、SAMOS 等。

3. 面向对象数据库

面向对象程序设计是软件工程的重要发展方向之一，它可以提高程序设计的生产率、重用性以及可扩展性。为了满足各种新的数据应用的需要，诸如多媒体数据、空间数据、复杂对象、超文本、知识、时序数据等的管理，面向对象数据库引起了大家的广泛兴趣。

目前，面向对象数据库核心系统的一些技术问题已基本解决，诸如数据模型、数据语言、查询处理、对象管理、消息管理、版本管理、事务管理、物理结构等都有了具体的解决办法。当前有关论文主要讨论面向对象数据库的数据模型、程序设计语言和提高性能。从 1987 年以来已出现许多商品化的面向对象数据库系统，例如 Servio 公司的 GemStone, Object Design 公司的 ObjectStore, Objectivity 公司的 Objectivity/DB, Versant Object Technology 公司的 Versant, ONTOS 公司的 ONTOS, Itasca Systems 公司的 Itasca, O2 Technology 公司的 O2。这些产品都支持面向对象数据模型。具体地说，它们允许用户建立新的属性和方法类型，可以从父类继承属性和方法，建立具有唯一对象标识的例示，可以单个或集体地检索例示、装载和运行方法。

这里，以 O2 为例说明面向对象数据库的结构。O2 是一个开放的系统，用 C 和 C++ 编写，在 Unix 环境下运行。它的接口工具用 C++ 编写，在 X11 和 OSF/Motif 下运行。图 1.3 给出了 O2 的系统框图。

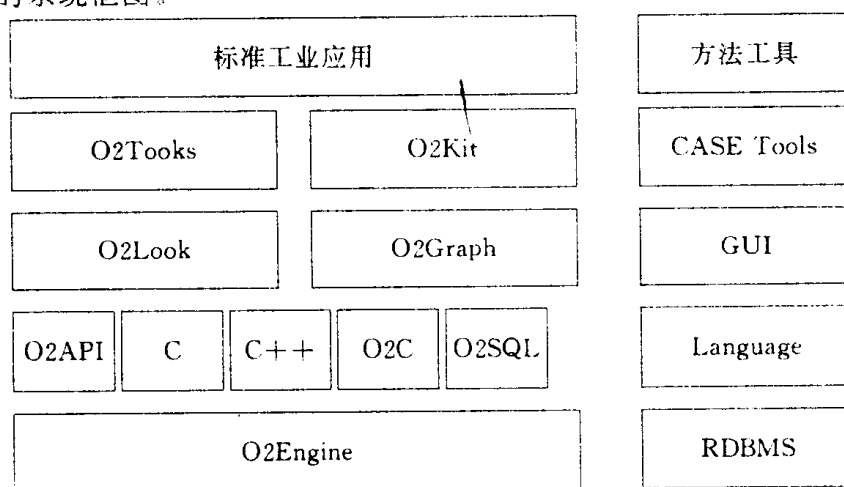


图 1.3 O2 系统框图

作为 O2 系统核心的 O2 Engine 具有下列特点：

- (1) 完全的面向对象模型；
- (2) 客户/服务器体系结构；
- (3) 支持多种数据库；
- (4) 支持多种模式；
- (5) 数据库中的方法；

- (6) 递增式无用单元收集器；
- (7) 提供安全保护；
- (8) 提供索引。

O2 Engine 的框图如图 1.4 所示。

O2 API 是应用程序与 O2 Engine 的接口,它可以直接访问 O2 Engine,对数据管理提供紧控制,具有高的性能。用 O2 编程非常方便。

意大利阿巴莫 (A. Albano) 等人的论文“一种强类型面向对象数据库程序设计语言的关系机制”主要提出一种对象关系数据模型。这种模型的特点是:

- (1) 将联系表示成为类之间 n 元对称关系;
- (2) 联系可以被组成专门的层次结构;
- (3) 参考完整性约束实际上是强迫的;
- (4) 联系的一些约束,诸如基数、映射、依赖性和非易变性可说明性定义;
- (5) 系统要求实现联系,参考约束可以在合理的代价下实现。

在这个模型中,典型的面向对象的聚集机理仍然可以用来建立对象之间的关系,用对象类型定义对象属性。类是相同类型各种元素的一个有序集合。联系是满足固定特征的各种联结的有序集合。类和联系都是语言的第一类值,它们的结构可以描述为:

Class(Element Type)Key Keylist...Key Keylistn

Assoc(Signature)Key Keylist...Key Keylistn

面向对象数据库原理在本书 3.3 节中还要详述,这里的 O2 等可作为面向对象模型的实例。

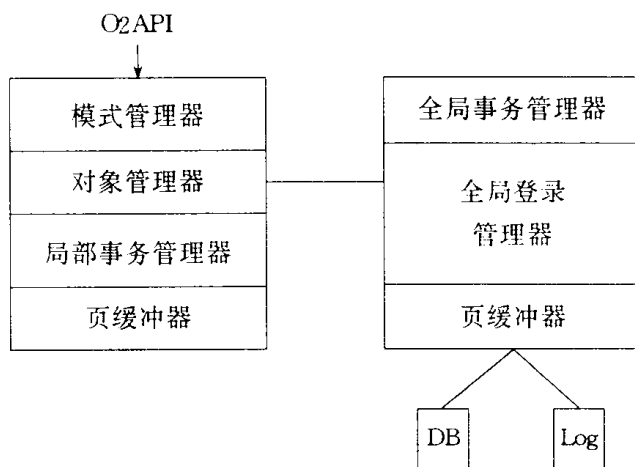


图1.4 O2 Engine体系结构

1.3 数据库的体系结构

1. 数据库的三级体系结构

数据库的体系结构分成三级:内部级(internal)、概念级(conceptual)和外部级(external)。图 1.5 所示的结构是 1975 年 2 月美国 ANSI/X3/SPARC(国家标准化委员会/关于计算机和信息处理/系统设计和要求委员会)提出的。虽然现在数据库管理系统的产品多种多样,在不同的操作系统支持下工作,但是绝大多数系统在总的数据库体系结构上仍具有三级结构的特征。

从某个角度看到的数据特性称为“数据视图”。外部级最接近用户,是单个用户所能看到的数据特征。单个用户使用的数据库视图称为“外模型”。概念级是涉及到所有用户的数据的定义,

也就是全局的数据视图,称为“概念模型”。内部级最接近于物理存储设备,涉及到实际数据存储的方式。物理存储的数据视图称为“内模型”。这些模型有两种表达方式,一种是用图示形式,即数据结构图;另一种是用数据定义语言描述。这三种模型用数据定义语言描述分别得到外模式(亦称为子模式)、概念模式(亦称为模式)和内模式(亦称为存储模式)。

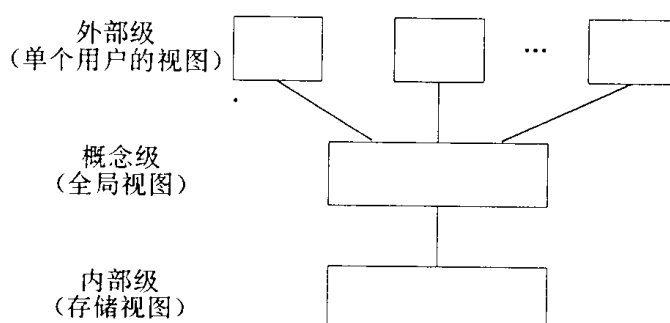


图1.5 三级结构

数据库的三级体系结构是数据的三个抽象级别,它把数据的具体组织留给数据库管理系统管理,使用户能抽象地处理数据,而不必关心数据在计算机中的表示和存储。这三级结构之间往往差别很大,为实现这三个抽象级别的转换,数据库管理系统在这三级结构之间提供了两层映像(mappings):外模式/模式映像,模式/内模式映像。

例 1.1: 图 1.6 是一个数据库的三级结构的例子。这个数据库的体系结构有一个内模式,一个概念模式,两个外模式。在概念级,数据库有一个记录类型 EMPLOYEE,它有三个字段,标明了相应的数据类型和长度。在内部级,记录类型名改为 STORED—EMP,内含一个指针 PREFIX,三个字段的命名也与概念级不一样。在外部级,一个外模式是用面向 PL/I 语言的数据定义语言描述,供 PL/I 应用程序使用;另一个外模式是用面向 COBOL 语言的数据定义语言描述,供 COBOL 应用程序使用。外模式中用到的记录名、字段名也与概念级不同(一般也可以用相同的命名)。在这三级描述中还应该说明这些命名的变化(图中未写出)。

External(PL/I)	External(COBOL)
DCL 1 EMPP	01 EMPC
2 EMP#CHAR(6)	02 EMPNO PIC X(6)
2 SAL FIXED BIN(31);	02 DEPTNO PIC X(4)
Conceptual	
EMPLOYEE	
EMPLOYEE-NUMBER	CHARACTER(6)
DEPARTMENT-NUMBER	CHARACTER(4)
SALARY	NUMERIC(5)
Internal	
STORED-EMP	LENGTH=18
PREFIX	TYPE=BYTE(6),OFFSET=0
EMP#	TYPE=BYTE(6),OFFSET=6,INDEX=EMPX
DEPT#	TYPE=BYTE(4),OFFSET=12
DAY	TYPE=FULLWORD,OFFSET=16

图1.6 数据库三级结构的例子

下面详细考察这个三级结构(图1.6)。

2. 概念模式(conceptual schema)

数据库的全局逻辑视图称为视图,由若干个概念记录类型组成。概念视图用 DDL (Data Definition Language, 数据定义语言) 描述后得到的是概念模式 (简称为模式)。描述概念视图的 DDL 称为“模式 DDL”。模式是所有概念记录类型的定义,因此它是数据库中全部数据逻辑结构的描述,即数据库中所有记录类型的整体描述。实际上,模式就是数据模型的描述。

模式还要描述记录之间的联系、所允许的操作、数据的一致性、有效验证、安全和其它管理控制方面的要求。数据按外模式的描述提供给用户,数据按内模式的描述存储在磁盘中。而概念模式提供了一种约束其它两级的相对稳定的中间观点,它使得这两级的任何一级的改变都不受另一级的牵制。

在模式中还必须达到数据独立性。模式描述中必须不涉及到存储结构、访问技术等细节,这样,模式描述中就不会遇到存储字段的表达、记录的物理顺序、索引方式等存储/访问细节问题。只有这样,模式才算做到了数据独立,而在模式基础上定义的外模式才能做到数据独立。

3. 外模式(external schema)

外模式是用户与数据库系统的接口。

单个用户的视图称为外部视图。一个用户往往只用到数据库的一部分。外部视图由若干外部记录类型组成,这些外部记录类型和概念记录类型、内部记录类型可能不一样。用户使用数据操纵语言语句对数据库进行操作,实际上是对外部视图的外部记录进行操作。例如 GET 操作是读一个外部记录值(实际上是逻辑记录值),而不是数据库的内部记录值。每个外部视图用 DDL 描述后得到的是外模式。描述外部视图的 DDL 称为“外模式 DDL”。外模式是外部视图中每个外部记录类型的定义,因此它是面向一个或多个用户需要的那部分数据的描述。在外模式中还必须定义外模式和概念模式间数据结构的差异。

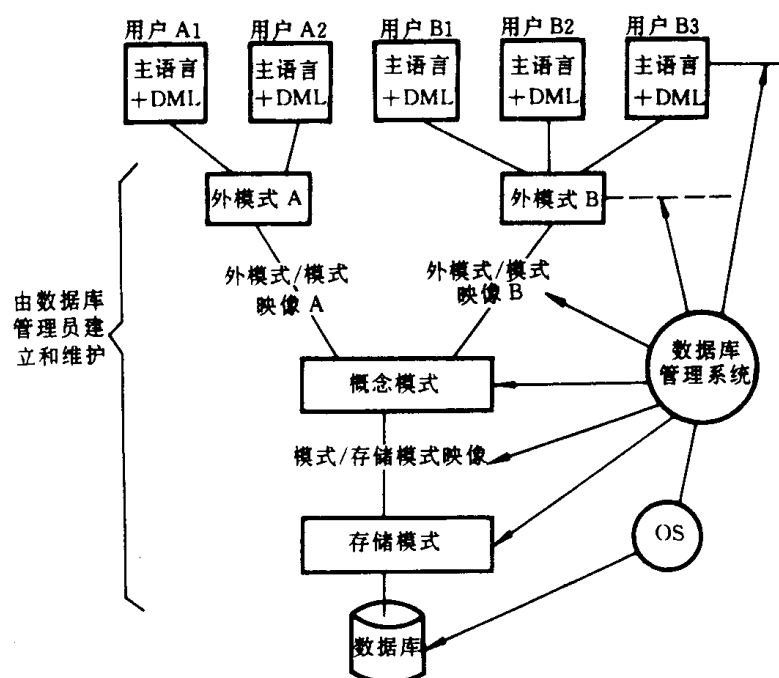


图1.7 数据库系统的数据体系结构

外模式与概念模式的分离,使数据库系统的灵活性大大增加。外模式是从模式导出的,它必须是导出它的模式的逻辑子集。但在外模式里,不仅那些与用户无关的数据可以略去,而且数据项可以按照用户使用的习惯重新命名,记录类型也可以重新组合和命名。应用程序员不必再关心整个数据库的全局逻辑结构,他只与外模式发生直接联系,按照外模式存储和操纵数据。

对于一个数据库系统而言,任何一个应用程序都必须使用而且只能使用一个外模式,才能对数据库中的数据进行操作。一个模式可以支持若干个外模式,但每一个外模式只能属于一个模式。

在网状系统中,外模式的描述方法和应用程序所使用的程序设计语言(通常称为宿主语言)有关,例如,COBOL 环境下外模式的描述与 FORTRAN 环境下的描述就不同,但它们完全可能对应同一个外模式。

4. 内模式(internal schema)

内部视图是数据库结构中最低一级的逻辑表达,它由若干内部记录类型组成。内部记录也称为存储记录。内部视图用 DDL 描述后得到的是内模式,描述内部视图的 DDL 称为“内模式 DDL”。内模式要定义所有的内部记录类型,定义一些索引、数据在存储器的安排以及安全性、恢复和其它管理方面的细节。所以内模式是数据在物理存储结构方面的描述。

由于内部记录不涉及物理记录和物理块,也不涉及设备的约束,因此内部级与物理级是不同的。比内模式更接近于物理存储的那些软件机制是属于操作系统的一部分(即文件系统),例如从磁盘上读一部分数据或写一部分数据到磁盘上等操作。

外模式、模式、内模式都有“源”与“目标”之分。用 DDL 书写的都是源模式,系统不能直接使用,必须由数据库管理系统提供编译程序,将源模式编译成二进制的目标码,成为机器可使用的模式,即目标模式,放在目标库(一般称为数据字典)中,以供系统随时调用。

5. 模式/内模式映像

这个映像存在于概念级和内部级之间,用于定义模式和内模式间的对应性。由于这两级的数据结构可能不一致,即记录类型和字段类型的组成可能不一样,因此需要说明概念记录和字段怎样对应到内部记录和字段。如果内模式要做修改,即数据库的存储设备和存储方法有所变化,那么模式/内模式映像也要做出相应的修改,但模式很可能仍然保持不变。也就是对内模式的修改尽量不要涉及到模式,当然对于外模式 and 应用程序的影响就更小,这样就可以说数据库达到了物理数据独立性。

模式/内模式映像一般是在内模式中描述的,但有的数据库系统把映像的部分内容放在模式中描述。

6. 外模式/模式映像

这个映像存在于外部级和概念级之间,用于定义外模式和模式之间的对应性。由于这两级的数据结构可能不一致,因此需要说明外部记录和字段怎样对应到概念记录和字段。一个模式可能有多个外模式,每个用户只使用一个外模式,但不同的用户可共享同一个外模式,不同的外模式可以重叠。如果数据库的整体逻辑结构(即模式)要做修改,那么外模式/模式映像也要做相应的修改,但外模式很可能仍然保持不变。也就是对模式的修改尽量不要涉及到外部级的外模式,当然对于应用程序的影响就更小,这样就说明数据库达到了逻辑数据独立性。外模式/模式映像都是在外模式中描述。