

高等学校教材

# 数据库系统原理

李战怀 主编



西北工业大学出版社

数据库系统原理

西北

13-43

H/1

出版社

**(陕)新登字 009 号**

**【内容简介】** 本书系统地介绍了关系数据库系统的基本概念、基本原理、设计技术及开发工具等。全书共分八章：第一章. 绪论；第二章. 数据模型；第三章. SQL 语言；第四章. 数据库的安全性与完整性；第五章. 关系数据库设计理论；第六章. 事务管理与并发控制；第七章. 数据库设计方法；第八章. 其它几种重要的数据库。

本书主要作为高等院校计算机系本科生和非计算机专业研究生的数据库教材，也可供有关数据库软件人员参考。

高等学校教材  
**数据库系统原理**  
李战怀 主编  
责任编辑 何格夫  
责任校对 钱伟峰

©1999 西北工业大学出版社出版

(邮编: 710072 西安市友谊西路 127 号 电话: 8491147)

全国各地新华书店发行

陕西西安市长安县印务总厂印装

ISBN 7-5612-1100-7/TP·159(课)

\*

开本: 787 毫米 × 1 092 毫米 1/16 印张: 10.125 字数: 240 千字

1999 年 6 月第 1 版

1999 年 6 月第 1 次印刷

印数: 1—2 000 册

定价: 12.50 元

---

购买本社出版的图书，如有缺页、错页的，本社发行部负责调换。

# 前 言

当今世界正处在信息爆炸的时代,普遍地使用计算机来处理各种信息和数据是信息时代最显著的一大特点,而在信息处理方面的应用业已成为计算机应用的主要领域。随着管理信息系统(MIS),决策支持系统(DSS),办公室自动化系统(OAS),计算机集成制造系统(CIMS)及信息高速公路等的飞速发展,数据库已成为这些信息系统的数据存储核心形式。千百万个各种类型的数据库管理系统配置在千百万台各种类型的计算机上,夜以继日地活跃在数据处理和信息产业领域内,有效地管理和组织着各类庞大的信息及数据,为人类开发信息资源做出了无法替代的贡献。

技术的发展和实际应用的需求使得数据库系统课程已成为计算机专业本科生和研究生的必修课程。

根据国内外数据库技术的发展和应用的现状,结合本科生的实际情况,我们编写了这本教材。其主要宗旨是:鉴于目前国内外数据库应用领域中关系数据库系统处于绝对的统治地位,本教材的安排和组织上就围绕着关系数据库而进行。期望在学完这门课程后,学生们能对关系数据库系统的基本原理、设计和应用有一个较好的理解,以便在走上工作岗位后能较快地承担实际工作。同时,也能对数据库技术的过去和发展有一个基本的了解。

本书由李战怀教授任主编。具体分工如下:李战怀负责全书结构和章节的安排,并参与了第五章、第六章和第八章的编写工作;李红燕编写了第三章和第四章,并为第一章、第七章之外的其余章节配备了习题;李宗良编写了第七章;蒋泽军和张民慧合编了第一章和第二章。

本书由西安交通大学沈君毅教授主审,他仔细审阅了全书,并提出了许多很好的意见,在此我们表示深深的谢意!

在本书的编写过程中,徐秋元教授、胡正国教授和张延园教授都给予了大力的支持、帮助和指导,在此向他们致以衷心的感谢!在资料的整理过程中,张阳、邓江等做了大量的工作,在此也向他们表示感谢!

本书是航空工业总公司“九五”统编教材,主要作为计算机科学与工程系本科生和大专生的教材,同时也可作为有关人员自学和开发数据库应用时的参考书。

虽然作者多年来从事数据库领域的研究、开发和教学工作,但由于学识所限,加之时间仓促,不足之处在所难免,热忱希望各位前辈、同行和读者对书中的错误或不当之处提出宝贵意见。

编 者

1998年9月于西安

# 目 录

第一章 绪论	1
1.1 什么是数据库	1
1.1.1 信息、数据、数据库	1
1.1.2 数据库系统与文件系统的区别	2
1.2 数据库管理系统	4
1.2.1 DBMS 功能	5
1.2.2 DBMS 结构	5
1.2.3 数据描述语言	7
1.3 数据库系统结构	8
1.3.1 集中式系统	8
1.3.2 个人计算机系统	8
1.3.3 分布式系统	9
1.3.4 客户/服务器系统	9
1.4 数据库技术的研究动态	9
1.4.1 数据库技术的现状	9
1.4.2 数据库技术的发展趋势	10
习题	11
第二章 数据模型	12
2.1 实体-联系模型	12
2.2 层次模型	14
2.3 网状模型	16
2.3.1 网状模型的数据结构	16
2.3.2 网状模型的数据操纵	18
2.4 关系模型	18
2.4.1 关系模型的数据结构	19
2.4.2 关系模型的完整性	21
2.4.3 关系模型的数据操纵	21
习题	26
第三章 SQL 语言	27
3.1 SQL 语言的主要特点	27

3.2 SQL 语言的数据定义功能 .....	28
3.2.1 DDL 对表的创建、修改及撤销 .....	28
3.2.2 DDL 对索引的创建 .....	29
3.2.3 DDL 对视图的创建 .....	30
3.3 SQL 语言的模式操纵功能 .....	31
3.3.1 表的修改 .....	31
3.3.2 表的撤销 .....	32
3.4 SQL 语言的数据查询功能 .....	32
3.4.1 简单查询 .....	33
3.4.2 条件查询 .....	33
3.4.3 联结查询 .....	33
3.4.4 带聚集函数的查询 .....	34
3.4.5 嵌套查询 .....	34
3.4.6 对查询结果的分组 .....	36
3.4.7 对查询结果的排序 .....	36
3.4.8 对查询结果的集合运算 .....	37
3.5 SQL 语言的数据操纵功能 .....	37
3.5.1 INSERT 语句 .....	38
3.5.2 UPDATE 语句 .....	39
3.5.3 DELETE 语句 .....	40
3.5.4 对视图的操作 .....	40
3.6 SQL 语言的数据控制功能 .....	42
3.6.1 完整性保护 .....	42
3.6.2 安全性保护 .....	43
3.6.3 事务管理 .....	45
3.6.4 封锁机制 .....	46
3.7 嵌入式 SQL 语言 .....	46
3.7.1 游标机制 .....	47
3.7.2 动态 SQL 语言 .....	49
习题 .....	52
<b>第四章 数据库的安全性及完整性 .....</b>	<b>53</b>
4.1 数据库系统的安全性 .....	53
4.1.1 用户身份鉴别(User Identification) .....	54
4.1.2 视图机制(View Mechanism) .....	55
4.1.3 存取控制(Access Control) .....	57
4.1.4 审计(Auditing) .....	61
4.1.5 数据加密(Data Encryption) .....	62
4.2 数据库系统的完整性 .....	62

4.2.1	语义完整性约束	65
4.2.2	触发器机制	66
4.2.3	规则机制	69
习题		72
<b>第五章</b>	<b>关系数据库设计理论</b>	<b>73</b>
5.1	引论	73
5.2	依赖理论	74
5.2.1	函数依赖	74
5.2.2	函数依赖的公理系统	76
5.3	规范化理论	80
5.3.1	什么是规范化	80
5.3.2	第一范式	81
5.3.3	第二范式	81
5.3.4	第三范式	82
5.3.5	Boyce-Codd 范式	82
5.3.6	第四范式	83
5.3.7	第五范式	83
5.3.8	各范式之间的关系	83
5.4	模式设计的方法学	84
5.4.1	模式设计的标准	84
5.4.2	关系模式的分解	84
5.4.3	把关系模式无损联结地分解成 BCNF	89
5.4.4	把关系模式无损联结及保持依赖地分解成第三范式	91
习题		92
<b>第六章</b>	<b>事务管理与并发控制</b>	<b>93</b>
6.1	事务	93
6.2	封锁机制	96
6.2.1	锁模式	96
6.2.2	两阶段封锁协议	97
6.2.3	死锁	97
6.3	乐观的并发控制	98
6.4	多粒度锁	99
6.5	恢复	101
6.6	工程设计事务的要求	103
6.7	在客户/服务器体系结构中的并发控制	104
6.8	组事务	106
习题		107

<b>第七章 数据库设计方法</b> .....	109
7.1 数据库设计步骤及方法 .....	109
7.1.1 ER 设计方法 .....	110
7.1.2 IDEF1X 设计方法 .....	111
7.1.3 ORM 设计方法 .....	114
7.2 计算机辅助数据库设计——结构设计 .....	115
7.2.1 Erwin .....	115
7.2.2 SmartER .....	116
7.2.3 InfoModeler .....	121
7.3 计算机辅助数据库设计——行为设计 .....	122
7.4 数据库性能分析及优化 .....	122
7.5 数据库试运行与维护 .....	123
7.5.1 数据库试运行 .....	123
7.5.2 数据库维护 .....	124
习题.....	124
<b>第八章 其它几种重要的数据库</b> .....	125
8.1 分布式数据库 .....	125
8.1.1 分布式数据库的定义 .....	126
8.1.2 分布式与集中式数据库的特性 .....	128
8.1.3 为什么要设计分布式数据库 .....	131
8.1.4 分布式数据库管理系统 .....	132
8.1.5 客户/服务器体系结构.....	134
8.2 面向对象数据库 .....	135
8.2.1 面向对象数据库与传统数据库的主要区别 .....	136
8.2.2 面向对象的基本概念 .....	137
8.2.3 在数据库中使用抽象数据类型和对象 .....	139
8.3 多媒体数据库 .....	143
8.3.1 多媒体数据库的背景 .....	144
8.3.2 多媒体数据的逻辑及物理建模 .....	145
8.3.3 表示特征 .....	147
8.3.4 有格式和无格式数据库的集成 .....	149
习题.....	152
<b>参考文献</b> .....	153

# 第一章 绪 论

数据库技术从产生(60年代末)到现在虽然只有30年的时间,然而数据库系统的理论、技术与方法迅速发展并日益完善,同时数据库技术与人工智能、并行计算、网络通信及面向对象设计等技术的结合,使计算机的应用水平和应用范围都得到了很大的提高,数据库系统已成为计算机应用中不可缺少的组成部分。因此,学习和掌握数据库系统的理论、技术与方法是全面认识计算机系统的一个重要环节。

## 1.1 什么是数据库

### 1.1.1 信息、数据、数据库

众所周知,现在已进入了信息社会。那么到底什么是信息呢?信息与自然界中存在的物质一样,也是一种资源。信息伴随物质而存在,并随物质的变化而变化,物质是信息的基础,信息是物质的描述。那么信息与数据又有什么关系呢?我们先来看一个例子,一台冰箱就是客观存在的物质,从它得到的诸如使用寿命长、省电等描述就是关于这台冰箱的信息。然而这样的信息给人的感觉非常模糊,不能给人以准确而深刻的印象。如果将上述信息用具体的数据来描述:这台冰箱的使用寿命至少为30年,每小时的耗电量仅为 $0.68 \text{ kW} \cdot \text{h}$ 。这样一来,这条信息给人的印象就非常清晰而深刻了。由此可见,信息和数据有一定的区别。但真正有用的信息往往是通过数据来体现的,所以有时很难将信息与数据区分开来。信息处理与数据处理常常是指同一个概念,计算机之间交换数据也可以说成是交换信息。

自然界中的信息千差万别,来源、种类、数量和用途也是千变万化。原始信息(数据)有些可直接被应用,而有些则需经过一定的分类、加工和处理,才能变得更加有用,得到更加广泛和充分的应用。随着人类文明(包括物质文明与精神文明)的不断进步,人类社会对数据处理的要求也越来越高。由于计算机硬件技术的飞速发展,处理器运算速度大幅度提高,存储器(包括内存和外存)的容量也迅速增加。计算机软件技术也提高很快,其中数据处理技术以惊人的速度发展着,从而使得计算机的绝大部分应用不是在数值计算方面,而是在数据处理方面。尤其在进入信息时代以后,数据处理方面的应用价值和意义是无法估量的。数据处理经过低级的手工管理阶段,文件系统阶段,发展到高级的数据库系统阶段,尤其是数据库技术与网络通信技术的结合,使数据处理的能力达到了前所未有的水平。它对于社会各领域中管理信息系统的建立提供了极为有力的软件技术支持。

至此,可以回答究竟什么是数据库了。

一个数据库就是一个可共享的数据集合,用于满足各种不同的信息需求,并且集合中的数据彼此之间有相互联系。

数据库有两个重要特性:数据结构化和数据共享。结构化实质将原本不同性质的数据文件



经过符合逻辑的编组,从而达到减少冗余,简化、方便数据访问的目的。共享是指所有具有访问权限的用户均可访问相同的数据,然而却可以用于不同的目的。

前面曾经提到,数据处理经过了由低级到高级,由简单到复杂的三个发展阶段:

- (1) 手工管理阶段。
- (2) 文件系统阶段。
- (3) 数据库系统阶段。

下面就介绍文件系统与数据库系统在数据处理方面有哪些不同。

### 1.1.2 数据库系统与文件系统的区别

数据处理的中心问题是数据管理。数据管理是指对数据进行分类、组织、编码、储存、检索和维护。我们在介绍数据库系统与文件系统有什么区别之前,先来看一看数据处理在初期,即人工管理阶段是怎样进行的。

50年代中期以前,计算机主要用于科学计算,少量用于数据处理。因此,对于数据保存的需求并不迫切,一般是在算题时,将原始数据与程序一起输入主存,运算处理后将结果数据输出。随着计算任务的完成,用户作业退出计算机系统,数据空间随同程序空间一起释放。由于没有软件系统对数据进行管理,程序员不仅要规定数据的逻辑结构,而且还要在程序中设计物理结构,包括存储结构、存取方法、输入输出方式等。这样,程序中存取数据的子程序随着存储的改变而改变,即数据与程序不具有独立性。程序员必须花费许多精力在数据的物理布置上,并且一旦数据在存储上产生一些改变,就必须修改程序以适应这些改变,否则程序就不能正确运行。这一阶段数据的组织方式必须由程序员手工设计。

那么50年代后期到60年代中期出现的文件系统,与60年代末期出现的数据库系统之间在数据管理方面有什么本质区别呢?

#### 1. 文件系统

信息系统设计的传统处理方法,集中在满足一个组织内各部门对数据处理的需求上。信息系统相应于不同用户数据查询的方式,是开发不同的计算机程序。例如,一个企业的订货、销售、人事和工资等管理,都有各自不同的程序。开发的每个应用程序或系统都是面向具体过程的,并且他们的设计是为了满足各个特定部门或用户组。也就是说,没有全面的计划或模型来指导应用的发展。每个新的计算机应用都有它自己的数据文件集合,可是这些新文件中的许多数据可能已经存在于其他应用的数据文件中。

所以,尽管文件系统的出现极大地推动了计算机应用的发展,但在数据管理方面仍然存在着以下问题。

(1) 数据冗余度大。在文件系统中,每个应用都有它自己的文件。这种方法必然导致很高的数据冗余度。将相同数据项保存在多个文件中有若干缺点:① 浪费了存储空间;② 同一数据项可能不得不输入多次,以修改它所有的出现;③ 数据不一致性。存储数据的不一致性是计算机应用中最常见的错误来源之一,由此导致文档和报告的不一致,并削弱了用户的信心。例如,一个用户的客户地址发生了变化,该客户在用户不同部门的旧地址有的改了,有的却没改。这样过时的地址就可能导致该客户尚未付款的账单寄往错误的地址。结果账单可能被退回,客户的付款推迟或丢失。

(2) 数据共享性差。用文件系统管理数据时,每个应用拥有其私有的文件,其它应用很少

有机会共享这些“私有”文件中的数据项,如图 1.1 所示。

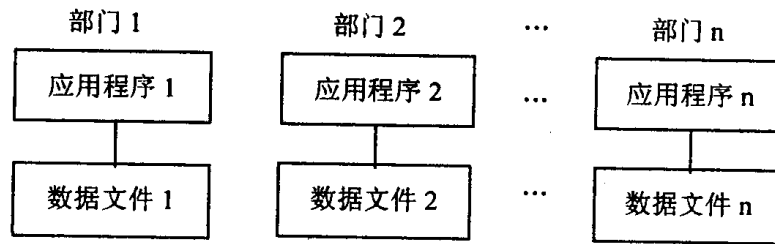


图 1.1 文件系统中应用与数据的关系

在图 1.1 中,部门 1 的应用程序就不能共享部门 2 的数据文件,数据文件 1 只能由部门 1 的应用程序独享。

(3) 数据独立性差。在文件系统中,文件是为某一特定应用服务的,数据结构与程序是合为一体的。一旦数据结构发生很小的变化,与之相关的应用程序都得改变,有时甚至会导致整个应用程序全部重写。

(4) 数据无集中管理。各应用程序私有的数据文件之间是彼此孤立的、无联系的,应用设计者不能面向应用全局来设计数据结构,而只能面向应用局部来设计数据文件。这样,文件之间不能反映现实世界事物之间的内在联系。

(5) 程序维护困难。在文件系统中,文件、记录和数据项的描述都嵌入在各个应用程序中。因此,对数据文件的任何修改(例如改变数据项的名称、格式或访问方式)都要求与之相联系的应用程序作相应的修改。

修改现存应用程序的过程称为程序维护。在文件系统阶段,由于数据文件的改变而引起的程序维护的开支是很大的。

从上述问题可以看出,用文件系统管理数据存在许多缺点,使应用程序的开发效率低、成本高。在这种情况下,迫切需要有一种新的技术、新的方法、新的理论来取代文件系统,以便更好地管理数据。

## 2. 数据库系统

随着 60 年代末数据库系统的出现和发展,数据处理进入了一个崭新的阶段。从前面所述的数据库的定义可以看出,在数据库系统中,数据可以为尽可能多的应用服务。

与传统文件系统相比,数据库系统在数据处理方面有以下优势:

(1) 数据冗余度小。在数据库系统中,以前分散和重复的数据文件被集中于一个单独的逻辑结构中,并且每一个数据项仅在数据库的一个地方出现。数据不再是面向某一特定局部应用,而是面向整个应用系统的。这样就大大减小了数据冗余度,既节约存储空间,减少存取时间,又可避免数据之间的不一致性。

不过,虽然数据冗余度大了不好,但也不是说一定要完全取消数据冗余。有时根据需要(例如,为了提高访问效率;数据合法性检查;在网络环境中,减少传输时间等)有意存储同一数据项的多个拷贝。尽管如此,数据库系统中的数据冗余度是受到控制的。在数据库系统中允许一定的数据冗余度,是为了改善系统的性能或提供其它方便,并且系统是(或应该)知道数据冗余的存在。

(2) 数据一致性好。数据不一致性是指数据的矛盾性、不相容性。通过消除(或控制)数据

冗余,数据库系统便极大地减小了数据不一致的机会。

(3) 数据集成性好。在一个数据库中,数据组织在一个单一的逻辑结构中,在彼此有联系的数据实体之间定义了逻辑关系。这样,用户可以容易地从一个数据项联系到另一个相关的数据项。数据管理软件执行这一功能,而不考虑数据项的物理组织或在数据库中的位置。

(4) 数据共享性好。数据库的建立,就是为了让所有具有访问权限的用户共享数据。任意多个应用可以同时使用相同的数据文件、记录、数据项值,并且可以开发新的应用而不用增加新的存储空间。

数据共享是数据库系统的主要特性之一。数据共享示意图如图 1.2 所示。

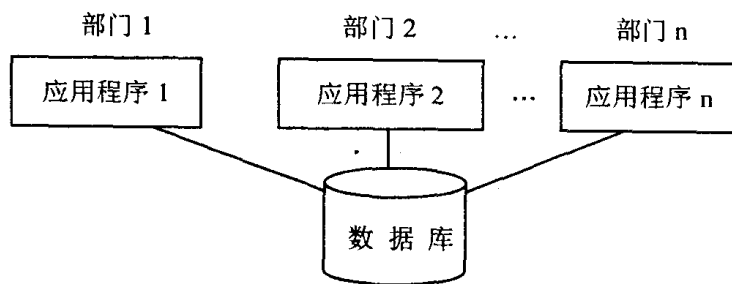


图 1.2 数据库系统中应用与数据的关系

(5) 数据结构化。数据库系统不仅描述数据实体本身,更重要的是还要描述数据实体之间存在的联系,这是数据库系统与文件系统之间的本质区别。后面将在第二章详细介绍数据库系统如何描述数据实体之间的联系。

数据的结构化也是数据库系统的主要特性之一。

(6) 具有一定的逻辑数据独立性。数据库系统的一个重要目标就是要使数据与程序彼此分离,使它们可以独立地发展。

数据库系统实现了以下两方面的数据独立性:

● 物理独立性。当数据的存储结构(或物理结构)改变时,数据的逻辑结构可以不变,从而应用程序也不必改变。这就是数据和程序的物理独立性,简称数据的物理独立性;

● 逻辑独立性。当系统总体逻辑结构改变时,通过对映象的相应改变而保持局部逻辑结构不变。应用程序是根据局部逻辑结构编写的,因而应用程序也就不必改变。这就是数据和程序的逻辑独立性,简称数据的逻辑独立性。

以上是数据库系统在数据处理方面的许多长处。除此之外,数据库系统还有其它一些特性,如数据安全性、数据完整性、并发控制、故障恢复、数据字典等,将在下一节中介绍。

## 1.2 数据库管理系统

数据库系统中用于全面管理数据的计算机软件就是数据库管理系统 DBMS(DataBase Management System)。

一个数据库管理系统就是一个通用软件系统,它管理数据库,提供组织、访问和控制等方面的工具。所谓“通用”是指 DBMS 独立于单个的应用,因而可被任何需要访问数据库的用户使用。

### 1.2.1 DBMS 功能

DBMS 是数据库的核心软件,是数据处理技术各种先进思想的汇集。它的主要功能有:

#### 1. 数据的存储、获取和修改

一个数据库可被许多用户共享。因此,DBMS 必须提供多个用户视图,允许用户方便、有效地存储、获取和修改他们的数据。

#### 2. 数据字典

数据字典是与一个组织相关数据的所有信息的储藏所。数据字典通常包含数据库完整性定义,安全保密定义,存取路径的定义等。DBMS 必须维护一个用户可访问的数据字典,这一服务可由 DBMS 自身的一个子集提供,或由一个独立的软件包提供。

#### 3. 安全机制

数据库中的数据必须受到保护,免遭由于意外或误用而引起的破坏。DBMS 的安全机制用于控制数据的访问(如用户口令、身份标识、用户级别、存取权限等),并且定义允许用户执行什么样的操作(例如只读或可读可写)。

#### 4. 数据完整性

数据完整性就是数据的正确性、真实性、客观性。一旦数据失真,其完整性就受到了破坏。DBMS 必须提供工具协助用户维护他们的数据的完整性。在 DBMS 及其软件接口中可设计多种编辑检查和完整性约束。在后续章节中将对完整性作详细介绍。

#### 5. 并发控制

由于一个数据库是由多个用户共享的,两个或更多的用户可能同时试图访问相同数据。如果两个用户并发地试图修改同一数据记录,就有可能产生错误结果,因为两个事务处理会相互干扰。DBMS 中必须有防护措施来避免或克服这种干扰。

DBMS 实现并发控制的基本方法是将被操作的数据进行封锁,封锁单位的大小将会影响到并行度的高低。封锁的单位越大,开销越小,但并行度就越低;反之封锁的单位越小,开销越大,但并行度却越高。因此,在决定封锁单位的大小时,应考虑到开销和并发度这两方面的因素。

#### 6. 故障恢复

数据库恢复是 DBMS 的重要任务之一。由各种各样的系统失效造成的数据库损坏或丢失是不可避免的,其原因是多方面的,例如:用户操作失误,硬件出错,不正确或不合法的数据,程序出错,以及自然灾害等。由于数据库在现代管理信息系统中有着举足轻重的地位,DBMS 必须提供机制,在数据库丢失或损坏之后迅速而准确地恢复数据库。

DBMS 应提供以下工具来恢复数据库:

- (1) 建立副本,提供整个数据库的周期性副本拷贝。
- (2) 建立日志,提供用于日后审查的有关事务处理和数据库变更的依据。
- (3) 设置检查点,DBMS 定期挂起所有处理,并使数据库的文件和日志同步。
- (4) 恢复或重启动工具,DBMS 将数据库恢复到一个正确的状态,并重新开始处理事务。

### 1.2.2 DBMS 结构

由于在文件系统中,数据和程序没有独立性,相互依赖,因此出现了许多问题。数据库管理

系统 DBMS 的出现,就是为了解决程序和数据的依赖问题。并且,它减轻了应用程序员的负担,可使应用程序员将精力集中于应用程序本身,用 DBMS 提供的工具来处理数据模型和数据语言的支持。

一个 DBMS 就是一个庞大的软件系统,它包含:

- 一个数据模型;
- 一个数据语言,用于定义和操纵数据;
- 一些工具,用于约束和实现安全性、完整性和并发性等,即数据库管理例程序。

用户与数据库打交道时,除了逻辑数据结构(即数据模型)外,还用数据描述语言 DDL (Data Description Language) 和数据操纵语言 DML (Data Manipulation Language) 作为交流的工具。模式这个概念包含了两层意思:数据结构及其表达工具。模式分为外模式和内模式,外模式包含所有已知的数据模型,内模式在下面介绍。图 1.3 描述了 DBMS 的结构。

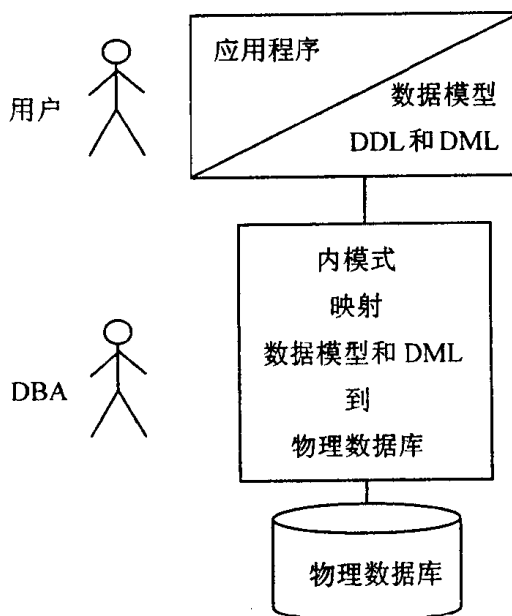


图 1.3 DBMS 结构

从图 1.3 中可以看出,内模式将逻辑数据结构转换为物理数据结构,将 DML 操作转换为操纵物理数据的文件操作。DBMS 就是通过这种方式获得数据独立性。与文件系统不同,在 DBMS 中,物理数据在不同用户之间不重复存储,它们是可共享的公共资源。它们之所以能实现共享,是因为内模式对所有用户提供了一个通用接口。在这样的环境下,需要一个数据库管理员 DBA (Data Base Administrator),

一方面帮助用户解决碰到的问题,另一方面管理数据库使之为用户最佳服务。这样,DBA 的责任包括数据库的创建和维护,保证数据库的安全性和完整性,系统检测,以及性能调整。为了达到这些,DBA 应负责设计、实现和控制数据库环境,并负责数据库的安全性和完整性。

一个 DBMS 必须符合一定的标准,如数据独立性,数据一致性等,因为一个 DBMS 可被共享,多个用户可并发地对共享数据进行修改。如果这种操作不受约束的话,就有可能使某些修改无效,表现为数据的不一致性。航空公司代理处的远程终端可能售出多于实际座位数量的飞机票;库存量已为零的某种货物仍被出售并为其产生了一个账单等。

由 DBMS 支持的数据语言,必须在通用的意义上满足数学运算及其工具特性的所有意义上和实践上的要求。这些语言应可从两方面来使用,一方面作为独立的查询语言,另一方面作为数据子语言 DSL (Data SubLanguage)。一个 DSL 就是一个 DDL 或 DML,只不过它被嵌入在一个一般的程序设计语言中,例如 Cobol, PL/1, C, 等等。一般程序设计语言因此被称为宿主语言。一个嵌入系统,即一个 DBMS 与一个 DSL,在应用程序中很有用。而一个独立查询语言则主要用来处理一些对数据库的特别请求,即想不到的、意外的、事先不可预知的请求等。图 1.4 描绘了一个嵌入式 DBMS 环境。

从图 1.4 中可看出,与数据库有关的代码段(用 DSL 语句来表示)以对 DBMS 的调用来实现,DBMS 既返回结果又返回状态(即成功、失败或数据结尾),并放入宿主程序的数据缓冲

区中,即图 1.4 中的“用户工作空间”。

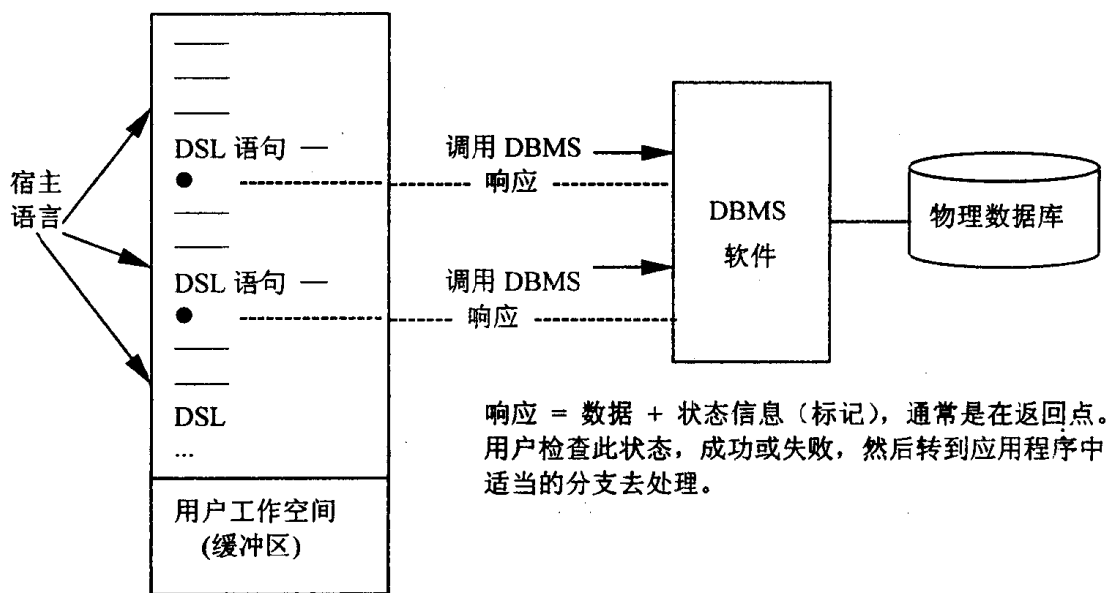


图 1.4 嵌入式 DML(DSL)和 DBMS 交互作用

### 1.2.3 数据描述语言

当用户使用数据库时,和用户首先见面的是 DDL。通常 DDL 分三种:模式 DDL,子模式 DDL,物理 DDL。数据描述语言一方面要方便用户,同时还要与整个系统协调,尤其是与 DML 协调配合。所以,它不是简单地定义数据,更重要的是反映系统内部的处理风格。在 DDL 中要尽量使用户给出的信息最少。用户只需给出必须给的信息,凡能通过计算、推导可获得的信息均由系统自动推出。这样既方便了用户,也提高了信息的准确度。

#### 1. 模式 DDL

模式描述定义面向应用对象的全局,它要顾及所有的应用要求,因此要尽可能保持中立,不偏向任何具体应用,不是对某个应用最优,而是对整体最优。因此它的信息要完整,尽可能少重复。

DBA 用模式 DDL 来定义数据库的整体逻辑结构。通常包括定义文件名、记录类,数据项名、种类、长度,文件间的联系,完整性,安全性约束等。

#### 2. 子模式 DDL

子模式定义是面向具体应用的。它要求按照用户的意图裁剪数据库,采用分割子集、归并、变换数据元素或关系导出子模式,所以它是模式的一个子集。在数据库技术中,由于有了子模式概念,才实现了数据库共享、安全保密、数据独立性等一系列的功能。

在子模式中,仅仅描述一个用户所感兴趣并且允许他进行操纵的数据,屏蔽了其它与用户无关的数据,这使系统具有数据独立性和保密性。子模式可以在数据库生命周期内任何时候定义而动态生成,用户可用它定义所需要的局部逻辑数据结构。子模式包括的内容与模式基本相同,只是各项命名、数据项的个数和次序等可不同于模式。

多个应用程序可共享同一子模式。

用户在使用模式与子模式描述语言时有两种方式：

● 程序语言的方式，即按系统的文法规定，由用户事先将模式及子模式用程序编出，再经编译生成目标；

● 交互式语言的方式，即系统提问，用户根据提问输入各种相应的信息。

这两种方式比较起来，交互式使用起来更方便一些。用户按问回答，一般不会发生语法错误；另外，许多保留字由系统自动显示在屏幕上，不需用户键入，从而也节省了人力和时间。

## 1.3 数据库系统结构

计算机体系结构是指计算机系统各组成部分之间的相互关系，它是一个综合概念，包括硬件、软件、算法和语言等。它研究的是硬件、软件的功能分配，以及机器界面的确立。关于计算机体系结构，人们过去主要强调的是计算机硬件结构的组成和功能实现，随着软件规模的日益庞大，结构日趋复杂，人们越来越认识到软件体系结构的重要性。

运行数据库的计算机系统大体上可以划分为四类，即集中式系统，个人计算机系统，分布式系统和客户/服务器系统。数据库系统的体系结构也相应地划分为这四类。当今的客户/服务器结构是数据库系统中最为常用的一种结构，它的数据也是分布的，因而广义地理解，它也是一种分布式系统。

### 1.3.1 集中式系统

在集中式系统中，所有的程序，包括 DBMS，应用程序以及与用户终端进行通信的软件等都运行在一台宿主计算机上，所有的数据处理都是在宿主计算机中进行。宿主计算机一般是大型机、中型机或小型机。应用程序和 DBMS 之间通过操作系统管理的共享内存或应用任务区来进行通信，DBMS 利用操作系统提供的服务来将数据写入磁盘，或从磁盘中读出数据。

用户通过本地终端或远程终端来访问数据库。终端通常是非智能的，本身没有处理能力。近年来，微处理器的出现引起了智能化终端的发展，这种终端可以完成某些用户的输入、输出处理。

集中式系统的主要优点是：具有集中的安全控制机制，以及处理大量数据和支持大量并发用户的能力。集中式系统的主要缺点是：购买和维持这样的系统一次性投资太大。

### 1.3.2 个人计算机系统

当 DBMS 在个人计算机(PC)上运行时，PC 机起到宿主机的作用，同时也起到了终端的作用。与大型系统不同，通常所谓的微机 DBMS 的功能和数据库应用功能是结合在一个应用程序中的，这使得 DBMS 的功能更加灵活，运行速度更高，但这是以降低数据库的安全性和完整性为代价的。

近年来，很多 PC 连入了局域网，在局域网中，数据驻留在文件服务器上。文件服务器负责管理局域网中用户对数据的共享访问。虽然基于 PC 的数据库用户通过局域网能够共享公共的数据文件，但文件服务器的作用只不过是在它的磁盘上检索用户需要的数据文件，并把数据文件传送到用户的 PC 上，然后由运行在用户 PC 上的 DBMS 对数据进行处理。如果数据稍作修改，用户 PC 又必须把整个数据文件传回文件服务器，并保存在磁盘上。这种系统的明显缺

点是:不管文件服务器速度多快,性能多强,数据库的性能都会受到实际运行 DBMS 的那个 PC 的性能限制。而且,网络上传输的是整个文件,数据传输量很大,这就大大降低了整个系统的性能。

### 1.3.3 分布式系统

一个分布式数据库系统由一个逻辑数据库组成,整个逻辑数据库的数据,存储在分布于网络中的多个结点上的物理数据库中。

在分布式数据库中,由于数据分布在网络的多个结点上,因此与集中式数据库相比,存在一些特殊的问题,例如,应用程序的透明性、结点自治性、分布式查询、分布式更新处理等,这就增加了系统实现的复杂性。

较早的分布式数据库是由多个宿主系统构成的,数据在各个宿主系统之间共享。在当今的客户/服务器结构的数据库系统中,服务器的数目可以是一个,也可以是多个。当系统中存在多个数据库服务器时,也就形成了一类分布式系统。

### 1.3.4 客户/服务器系统

在客户/服务器结构的数据库系统中,数据处理任务被划分为两部分:一部分运行在客户端,另一部分运行在服务器端。划分的方案可以有多种。一种常用的方案是:客户端负责应用的处理,数据库服务器端负责数据访问和事务处理。

在客户/服务器结构中,客户端软件和服务器端软件可以运行在一台计算机上,但大多数情况下是分别运行在网络中不同的计算机上。客户端软件一般运行在 PC 上,服务器端软件可以运行在从 PC 机到大型机等各类计算机上。与文件服务器相比,数据库服务器有明显的优越性。它把数据处理任务分开在客户端和服务器上运行,这有利于充分利用服务器的高性能数据处理能力以及客户端灵活的数据表示能力,同时还大大减少了网络上的数据传输量。通常从客户端发往数据库服务器的只是查询请求,从数据库服务器传回给客户端的只是查询的结果,而不像文件服务器那样,需要传送整个文件。

客户/服务器结构的数据库系统,也采用数据复制等技术,来解决分布数据处理的问题,以提高整个系统的效率。

客户/服务器结构的数据库系统是当前最为流行的和发展最为迅速的一种结构。

## 1.4 数据库技术的研究动态

### 1.4.1 数据库技术的现状

数据库技术从出现到现在不过 30 年,但它的飞速发展极大地推动了计算机在人类社会各领域中的应用,已成为不可替代的计算机支柱系统之一。

计算机技术从初期只能适应最简单的一些单一应用环境,发展到现在与其它高级技术如人工智能、分布式计算、网络通信等结合,形成了具有综合处理能力的数据库系统。例如:专家数据库系统,工程数据库系统,分布式数据库系统等。下面分别简要地介绍它们。



### 1. 专家数据库

人工智能与数据库技术是计算机领域中两个重要的分支。人工智能是研究计算机模拟人的大脑思维的一门科学,它在逻辑推理和知识处理方面有其特殊的本领,但对于信息检索则效率低,从而使它很难进入实用。而数据库技术是数据处理的最先进的技术,对于数据的存储、管理、检索等有其独特的优势,但对逻辑推理和知识处理却无能为力。专家数据库正是人工智能与数据库技术相结合的产物,它发扬了两者各自的长处而克服了它们的短处,使计算机可以更好地为人类服务。

### 2. 工程数据库

工程数据库最主要的应用就是在 CAD/CAM 领域。80 年代后期兴起的计算机集成制造系统 CIMS(Computer Integrated Management System)把计算机辅助技术的应用水平推向一个更新的里程碑。在 CIMS 中,信息的集成与管理是通过产品数据管理 PDM(Product Data Management)系统实现的,而 PDM 系统中必然包括有工程数据库及其管理系统(工程 DBMS)。

对一个生产企业来说,所需集成的信息种类是多种多样的,大体上可分为与 CAD/CAM 直接有关的技术信息和非直接相关的生产管理信息(或称事务类信息)。前一类信息的特点是数据类型多,数据结构复杂,对这类数据的操作要求也相应提高。这样,通常只适用于一般事务管理的数据库和数据库管理系统是无法胜任的。由于环境、要求的不同,必须要有适应于工程领域的数据库和数据库管理系统。不仅如此,工程数据库是 CAD/CAM 系统的信息源,它将 CAD/CAM 系统中各个软件模块如应用、方法库及图形处理系统联成一个整体。可以说,工程数据库及其数据库管理系统在 CAD/CAM 系统中有着举足轻重的地位。

### 3. 分布式数据库

分布式数据库是一个逻辑上属于同一整体而物理上分布存放在计算机网络结点上的数据集合。

纵观几十年来计算机应用的发展史,不难发现这样一个规律,即社会需求激励和推动着新的计算机应用技术的产生和发展。从人类社会的组织分布来看,很多部门的职能机构在逻辑上是一个整体,但在物理上,即地理位置上,却经常分散在不同的建筑物中,甚至不同的城市里。集中式数据库远远不能满足他们的需要。随着信息时代的到来,分布式数据库越来越显示出它的优越性。而个人 PC 机的普及和网络通信技术的发展,为分布式数据库的实现提供了物质基础。分布式数据库不仅从地理位置上满足了同一部门的分散机构,而且分布式数据库比集中式数据库有更大的可靠性,因为一个结点的故障不会影响整个系统的正常运行。到目前为止,分布式数据库技术还不是十分成熟,有待于进一步发展和完善,但它仍然是今后数据库的发展趋势之一。

#### 1.4.2 数据库技术的发展趋势

在当今的数据库领域中,有许多新的潮流,它们在不同的方向代表了新的发展趋势。它们之中的某些新技术(如面向对象程序设计,多媒体数据,国际互联网,等等),正在改变着传统数据库和数据库管理系统的性质。其它一些潮流则包含了新的应用,例如,数据仓库或信息集成。下面概要介绍几个数据库系统的未来发展趋势。