



21世纪高校计算机系列规划教材

数据库原理

陈利平 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

前 言

在当今信息化社会中，数据库是组织、管理和利用信息的最有效的方法，特别是互联网技术的应用与普及，更使数据库技术成为了大众化技术。因此，数据库技术不仅是计算机专业的重要课程，也是非计算机专业的必修课或选修课。该课程的主要任务是研究如何存储、使用和管理数据，目的是使学生掌握数据库的基本原理、方法和应用技术，能有效使用现有的数据库管理系统和软件开发工具，掌握数据库结构设计和数据库应用系统的开发方式。

数据库技术是计算机软件领域的一个重要分支。数据库技术由应用而产生，随着应用的普及和深入而蓬勃发展。因此，本书的宗旨是在系统介绍数据库技术的基本原理和方法的同时，突出其面向应用的特征。

本书在编排上，特别注意理论与应用的关系。因此，本书在突出数据库技术的理论、操作和应用的同时，还加强了对数据库设计这一实践性很强的内容的介绍，并且对数据库的新技术也进行了介绍，还配有习题和上机实验指导。具体内容如下：

第1章介绍数据库技术的历史，关系数据库的基本知识、基本术语。

第2章介绍关系数据模型的运算理论：关系代数和关系演算。

第3章介绍 SQL 语句的基本内容。

第4章介绍关系数据库的规范化理论：函数依赖、分解特性和范式等内容。

第5章介绍数据库设计的任务和特点、设计方法和步骤、数据库设计使用的辅助工具。

第6章介绍数据库的安全性、完整性、并发控制和恢复技术。

第7章介绍几种常见的数据库访问技术。

第8章介绍数据库新技术的部分内容。

本书还给出了上机实验指导，以便学生循序渐进地提高实际应用能力。

本书是集体劳动的成果，由陈利平担任主编，高金华、周光宇、柳佳刚、彭代文、李泽军、担任副主编，黎红、邓会敏、赵巾帼、桂友武、任长安、秦辉参与本书的编写工作。

由于水平有限，本书难免会存在疏漏和不足之处，敬请读者批评指正。

编 者

2007年1月

目 录

第 1 章 数据库系统概述	1
1.1 数据库技术的发展.....	1
1.1.1 人工管理阶段.....	1
1.1.2 文件系统阶段.....	1
1.1.3 数据库系统阶段.....	3
1.1.4 高级数据库阶段.....	4
1.2 数据库的基本概念.....	6
1.2.1 数据.....	6
1.2.2 数据库.....	6
1.2.3 数据库管理系统.....	7
1.2.4 数据库系统.....	7
1.2.5 用户.....	8
1.3 数据模型.....	8
1.3.1 概念模型.....	9
1.3.2 数据模型.....	13
1.3.3 层次模型.....	14
1.3.4 网状模型.....	16
1.3.5 关系模型.....	17
1.4 数据库系统的结构.....	19
1.4.1 数据库系统的三级模式结构.....	19
1.4.2 数据库系统的二级映像.....	21
1.4.3 面向用户的数据库体系结构.....	22
1.5 数据库管理系统.....	23
1.5.1 数据库管理系统的工作模式.....	23
1.5.2 数据库管理系统的主要功能.....	25
1.5.3 DBMS 的组成.....	25
1.6 小结.....	26
习题.....	27
第 2 章 关系数据模型	28
2.1 关系数据模型的基本概念.....	28
2.1.1 关系、元组、属性、域、分量、关系模式.....	28
2.1.2 关键字.....	29
2.1.3 关系数据模型的集合论定义.....	30
2.1.4 关系数据模型的完整性约束.....	32

2.2	关系的运算.....	33
2.2.1	关系代数.....	34
2.2.2	关系代数运算实例.....	41
2.2.3	关系演算.....	44
2.3	查询优化.....	46
2.3.1	查询优化实例.....	47
2.3.2	查询优化的一般准则.....	48
2.3.3	关系代数等价变换规则.....	48
2.3.4	关系代数表达式优化的算法.....	50
2.4	小结.....	53
	习题.....	54
第3章	关系数据库的标准语言 SQL	56
3.1	SQL 概述.....	56
3.1.1	SQL 的特点.....	56
3.1.2	SQL 语言的基本概念.....	57
3.2	SQL 的数据定义.....	58
3.2.1	基本表的定义.....	58
3.2.2	基本表的修改与删除.....	59
3.2.3	索引的建立和维护.....	60
3.3	SQL 的数据操纵.....	62
3.3.1	数据查询.....	62
3.3.2	插入数据.....	79
3.3.3	删除数据.....	79
3.3.4	修改数据.....	80
3.4	视图.....	80
3.4.1	建立视图.....	80
3.4.2	删除视图.....	81
3.4.3	查询视图.....	81
3.4.4	更新视图.....	82
3.4.5	视图的优点.....	83
3.5	触发器.....	83
3.5.1	触发器结构.....	83
3.5.2	创建触发器.....	83
3.6	小结.....	84
	习题.....	84
第4章	关系数据库规范化理论	87
4.1	关系规范化的作用.....	87
4.1.1	问题的提出.....	87

4.1.2	解决的方法.....	88
4.1.3	关系模式规范化.....	90
4.2	函数依赖.....	90
4.2.1	关系模式的简化表示.....	91
4.2.2	函数依赖的基本概念.....	91
4.2.3	码的函数依赖表示.....	92
4.2.4	函数依赖和码的唯一性.....	93
4.3	函数依赖的公理系统.....	93
4.3.1	函数依赖集的完备性.....	93
4.3.2	函数依赖的推理规则.....	94
4.3.3	属性的闭包与 F 逻辑蕴含的充要条件.....	96
4.3.4	最小函数依赖集 F_{min}	97
4.4	关系模式的规范化.....	98
4.4.1	规范化的含义.....	99
4.4.2	第一范式.....	99
4.4.3	第二范式.....	100
4.4.4	第三范式.....	101
4.4.5	BCNF 范式.....	102
4.5	多值依赖与第四范式.....	103
4.5.1	问题的引入.....	103
4.5.2	多值依赖的基本概念.....	104
4.5.3	第四范式.....	106
4.6	关系模式的分解.....	106
4.6.1	无损分解.....	107
4.6.2	保持函数依赖.....	110
4.7	连接依赖与第五范式.....	112
4.7.1	连接依赖.....	112
4.7.2	第五范式.....	114
4.8	关系模式规范化步骤.....	114
4.9	小结.....	115
	习题.....	116
第 5 章	数据库设计	118
5.1	数据库设计概述.....	118
5.1.1	数据库设计的任务.....	118
5.1.2	数据库设计的特点.....	119
5.1.3	数据库设计的方法.....	119
5.1.4	数据库设计的工具.....	120
5.1.5	数据库设计的步骤.....	121

5.2	需求分析.....	122
5.2.1	需求分析的任务.....	122
5.2.2	需求收集.....	123
5.2.3	需求分析.....	124
5.2.4	需求分析的结果.....	126
5.3	概念结构设计.....	126
5.3.1	概念结构设计概述.....	126
5.3.2	概念结构设计的方法.....	127
5.3.3	局部 E-R 图设计.....	128
5.3.4	全局 E-R 图设计.....	130
5.4	逻辑结构设计.....	131
5.4.1	概念模型转换为关系数据模型.....	131
5.4.2	关系模型的优化.....	132
5.4.3	设计用户子模式.....	133
5.5	物理结构设计.....	134
5.5.1	物理结构设计概述.....	134
5.5.2	存取方法选择.....	135
5.5.3	存储结构的确定.....	136
5.6	数据库的实施和维护.....	136
5.6.1	数据库的实施.....	136
5.6.2	数据库的维护.....	137
5.7	小结.....	138
	习题.....	138
第 6 章	数据库的管理.....	140
6.1	数据库的安全性.....	140
6.1.1	数据库安全性概述.....	140
6.1.2	用户标识和鉴别.....	141
6.1.3	存储控制.....	142
6.1.4	审计管理.....	144
6.1.5	SQL Server 安全管理.....	144
6.2	数据库的完整性.....	150
6.2.1	数据库完整性概述.....	150
6.2.2	完整性约束条件.....	151
6.2.3	完整性控制.....	152
6.2.4	SQL Server 的完整性.....	154
6.3	数据库的并发控制.....	156
6.3.1	事务的基本概念.....	157
6.3.2	并发控制概述.....	158

6.3.3	封锁.....	159
6.3.4	不加锁的并发控制.....	165
6.3.5	SQL Server 的并发控制.....	166
6.4	数据库的恢复技术.....	169
6.4.1	数据库恢复概述.....	169
6.4.2	故障种类.....	169
6.4.3	恢复基本策略.....	170
6.4.4	需要备份的数据.....	171
6.4.5	SQL Server 的恢复技术.....	173
6.5	小结.....	175
	习题.....	176
第 7 章	数据库系统的访问	177
7.1	数据库的访问接口.....	177
7.1.1	固有调用.....	177
7.1.2	ODBC.....	177
7.1.3	JDBC.....	182
7.2	客户机/服务器模式的数据库系统.....	186
7.3	浏览器/服务器模式的数据库系统.....	188
7.3.1	Web 数据库的体系结构.....	188
7.3.2	技术实现.....	189
7.4	Java 访问数据库的技术.....	190
7.4.1	JDBC 技术.....	190
7.4.2	JSP 的数据库访问技术.....	191
7.5	数据库系统的多层体系结构.....	192
7.6	小结.....	193
	习题.....	194
第 8 章	数据库技术的发展	195
8.1	数据库新技术的分类.....	195
8.2	面向对象数据库系统.....	196
8.2.1	面向对象程序设计方法.....	196
8.2.2	面向对象数据模型.....	197
8.2.3	面向对象数据库语言.....	200
8.2.4	面向对象数据库模式的演进.....	201
8.2.5	对象-关系数据库.....	202
8.3	分布式数据库系统.....	203
8.3.1	分布式数据库系统的特点与目标.....	203
8.3.2	分布式数据库系统的体系结构.....	206
8.4	网络环境下的数据库体系.....	207

8.4.1	客户机/服务器系统.....	207
8.4.2	开放式数据库的互连技术.....	210
8.5	数据仓库与数据挖掘.....	213
8.5.1	数据仓库.....	213
8.5.2	数据仓库设计实例.....	220
8.5.3	数据挖掘技术概述.....	223
8.6	其他新型的数据库系统.....	225
8.6.1	多媒体数据库系统.....	225
8.6.2	主动数据库系统.....	226
8.6.3	演绎数据库系统.....	226
8.6.4	实时数据库系统.....	228
8.7	小结.....	228
	习题.....	228
附录 A	上机实验指导.....	229
参考文献	239

第 1 章 数据库系统概述

现在数据库已是每一项业务的基础。数据库被应用于维护商业内部记录，在万维网上为顾客和客户显示数据，以及支持很多其他商业处理。数据库同样出现在很多科学研究的核心中，天文学家、地理学家及其他很多科学家搜集的数据也是用数据库表示的。数据库也用在企业、行政部门。因此，数据库技术已成为当今计算机信息系统的核心技术，是计算机技术和应用发展的基础。经过 40 多年的发展，数据库已形成了较为完整的理论体系和实用技术。

本章主要介绍数据库技术的发展和数据库系统涉及的最基本、最重要的概念，包括数据模型、数据库管理系统和数据库系统的组成。

1.1 数据库技术的发展

随着计算机硬件和软件的发展，计算机数据管理方法至今大致经历了四个阶段：人工管理阶段、文件系统阶段、数据库系统阶段和高级数据库阶段。

1.1.1 人工管理阶段

在人工管理阶段（20 世纪 50 年代中期以前），计算机主要用于科学计算，其他工作还没有展开。外部存储器只有磁带、卡片和纸带等，还没有磁盘等字节存取存储设备。软件只有汇编语言，没有操作系统和管理数据的软件。数据处理的方式基本上是批处理。这个阶段数据管理的特点如下：

（1）数据不保存：因为该阶段计算机主要应用于科学计算，对于数据保存的需求尚不迫切，只是在计算某一课题时将数据输入，完成后得到结果，因此无需保存数据。

（2）系统没有专用的软件对数据进行管理：数据需要由应用程序自己管理，没有相应的软件系统负责数据的管理工作。因此，每个应用程序不仅要规定数据的逻辑结构，而且要设计物理结构，包括存储结构、存取方法、输入方式等，因此程序员负担很重。

（3）数据不共享：数据是面向程序的，一组数据只能对应一个程序。多个应用程序涉及某些相同的数据时，也必须各自定义，因此程序之间有大量的冗余数据。

（4）数据不具有独立性：程序依赖于数据，如果数据的类型、格式或输入、输出方式等逻辑结构或物理结构发生变化，则必须对应用程序做出相应的修改。

在人工管理阶段，程序与数据之间的关系可用图 1-1 表示。

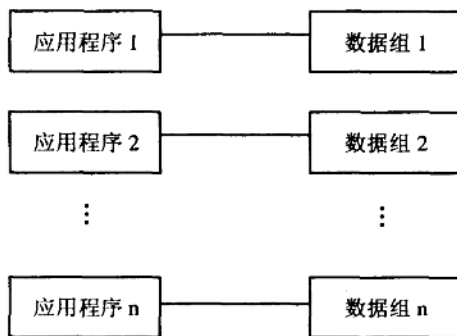


图 1-1 人工管理阶段

1.1.2 文件系统阶段

从 20 世纪 50 年代后期到 20 世纪 60 年代中期，计算机不仅用于科学计算，还大量应用于信息管理。大量的数据存储、检索和维护成为紧迫的需求。在硬件方面，有了磁盘、磁鼓

等直接存储设备；在软件方面，出现了高级语言和操作系统，且操作系统中有了专门管理数据的软件，一般称为文件系统；在处理方式方面，不仅有批处理，还有联机实时处理。

用文件系统管理数据的特点如下：

(1) 数据可以文件形式长期保存下来：由于计算机大量用于数据处理，数据需要长期保存在存储器上，以使用户可随时对文件进行查询、修改和增加、删除等处理。

(2) 文件系统可对数据的存取进行管理：有专门的软件即文件系统进行数据管理，文件系统把数据组织成相互独立的数据文件，利用“按名访问，按记录存取”的管理技术，对文件进行修改、插入和删除的操作。因此，程序员只与文件名打交道，不必明确数据的物理存储，大大减轻了程序员的负担。

(3) 文件组织多样化：有顺序文件、链接文件和索引文件等，因而对文件的记录可顺序访问，也可随机访问，更便于存储和查找数据。但文件之间相互独立、缺乏联系。数据之间的联系要通过程序去构造。

(4) 程序与数据之间有一定独立性：由专门的软件即文件系统进行数据管理，程序和数据之间由软件提供的存取方法进行转换，数据存储发生变化不一定影响程序的运行，既可大大节省维护的工作量，又可减轻程序员的负担。

与人工管理阶段相比，文件系统阶段对数据的管理有了很大的进步，但一些根本性问题仍没有彻底解决，主要表现在以下三个方面：

(1) 数据冗余度大：由于数据的基本存取单位是记录，因此，程序员之间很难明白他人数据文件中的数据的逻辑结构。理论上，一个用户可通过文件管理系统访问很多数据文件，然而实际上，一个数据文件只能对应于同一程序员的一个或几个程序，不能共享，即文件仍然是面向应用的。当不同的应用程序具有部分相同的数据时，也必须建立各自的文件，而不能共享相同的数据，因此数据的冗余度大，浪费存储空间。

(2) 数据独立性差：文件系统文件是为某一特定应用服务的，文件的逻辑结构对该应用程序来说是优化的，若要对现有的数据增加一些新的应用会很困难，系统不容易扩充。数据和程序相互依赖，一旦改变数据的逻辑结构，则必须修改相应的应用程序。而应用程序发生变化，如改用另一种程序设计语言来编写程序，也需修改数据结构。因此，数据和程序之间缺乏独立性。

(3) 数据一致性差：由于相同数据要重复存储、各自管理，在进行更新操作时，容易造成数据的不一致性。

在文件系统阶段，程序与数据之间的关系可用图 1-2 表示。

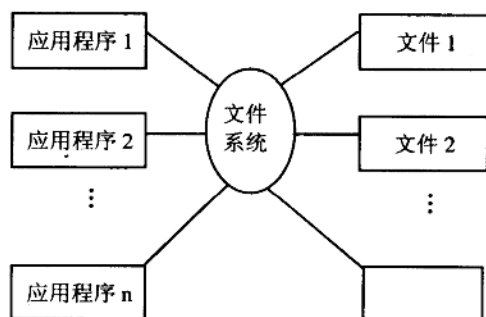


图 1-2 文件系统阶段

【例 1.1】 某学校利用计算机对教职工的基本情况进行管理，各部门分别建立三个文件：职工档案文件、职工工资文件和职工保险文件。每一职工的电话号码在这三个文件中重复出现，这就是“数据冗余”。若某职工的电话号码需要修改，就要修改这三个文件中的数据，否

则会导致同一数据在三个文件中不一样；产生的原因主要是三个文件中的数据没有联系。

若在职工档案文件中存放电话号码值，而其他文件中不存放电话号码值，并将存放档案文件中电话号码值的位置作为“指针”，则可消除文件系统中的三个缺点。

1.1.3 数据库系统阶段

20世纪60年代后期，计算机硬件、软件有了进一步的发展。计算机应用于管理的规模更加庞大，数据量急剧增加；硬件方面出现了大容量磁盘，使计算机联机存取大量数据成为可能；硬件价格下降，而软件价格上升，使开发和维护系统软件的成本增加。文件系统的数据管理方法已无法适应开发应用系统的需要。为解决多用户、多个应用程序共享数据的需求，出现了统一管理数据的专门软件系统，即数据库管理系统。用数据库系统来管理数据比用文件系统具有明显的优点，从文件系统到数据库系统，标志着数据管理技术的飞跃。

用数据库系统管理数据的特点如下：

(1) 数据结构化：数据结构化是数据库与文件系统的根本区别。

有了数据库管理系统后，数据库中的任何数据都不属于任何应用。数据是公共的，结构是全面的。它是在对整个组织的各种应用（包括将来可能的应用）进行全局考虑后建立起来的总的数据库结构。它是按照某种数据模型，将全组织的各种数据组织到一个结构化的数据库中，整个组织的数据不是一盘散沙，可表示出数据之间的有机联系。

【例 1.2】要建立学生成绩管理系统，系统包含学生（学号、姓名、性别、系别、年龄）、课程（课程号、课程名）、成绩（学号、课程号、成绩）等数据，分别对应三个文件。

若采用文件处理方式，因为文件系统只表示记录内部的联系，而不涉及不同文件记录之间的联系，要想查找某个学生的学号、姓名、所选课程的名称和成绩，必须编写一段较复杂的程序来实现。

而采用数据库方式，数据库系统不仅描述数据本身，还描述数据之间的联系，上述查询可以非常容易地实现。

(2) 数据共享性高、冗余少、易扩充：数据库系统从全局角度看待和描述数据，数据不再面向某个应用程序，而是面向整个系统，因此数据可以被多个用户、多个应用共享使用。这样便减少了不必要的数据库冗余，节约了存储空间，同时也避免了数据之间的不相容性与不一致性。

由于数据面向整个系统，是有结构的数据，不仅可被多个应用共享使用，而且容易增加新的应用，这就使得数据库系统弹性大，易于扩充，可以适应各种用户的要求。

(3) 数据独立性高：数据的独立性是指数据的逻辑独立性和数据的物理独立性。

数据的逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的，即当数据的总体逻辑结构改变时，数据的局部逻辑结构不变，由于应用程序是依据数据的局部逻辑结构编写的，因而不必修改应用程序，从而保证了数据与程序间的逻辑独立性。

例如，在原有的记录类型之间增加新的联系，或在某些记录类型中增加新的数据项，均可确保数据的逻辑独立性。

数据的物理独立性是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的，即当数据的存储结构改变时，数据的逻辑结构不变，从而也不必修改应用程序。

例如，改变存储设备和增加新的存储设备，或改变数据的存储组织方式，均可确保数据的物理独立性。

(4) 有统一的数据控制功能：数据库为多个用户和应用程序所共享，对数据的存取往往是并发的，即多个用户可以同时存取数据库中的数据，甚至可以同时存取数据库中的同一个数据，为确保数据库数据的正确有效和数据库系统的有效运行，数据库管理系统提供下述四方面的数据控制功能。

① 数据的安全性 (security) 控制

数据的安全性是指保护数据以防止不合法使用数据造成数据的泄露和破坏，保证数据的安全和机密。使每个用户只能按规定，对某些数据以某些方式进行使用和处理。

例如，系统提供口令检查或其他手段来验证用户身份，防止非法用户使用系统；也可以对数据的存取权限进行限制，只有通过检查后才能执行相应的操作。

② 数据的完整性 (integrity) 控制

数据的完整性是指系统通过设置一些完整性规则以确保数据的有效性、正确性和相容性。完整性控制将数据控制在有效的范围内，或保证数据之间满足一定的关系。

有效性是指数据是否在其定义的有效范围，如月份只能用 1~12 之间的正整数表示。

正确性是指数据的合法性，如年龄属于数值型数据，只能含 0, 1, ..., 9, 不能含字母或特殊符号。

相容性是指表示同一事实的两个数据应相同，否则就不相容，如一个人不能有两个性别。

③ 并发 (concurrency) 控制

多用户同时存取或修改数据库时，可能会发生相互干扰而提供给用户不正确的数据，并使数据库的完整性受到破坏，因此必须对多用户的并发操作加以控制和协调。

④ 数据恢复 (recovery)

计算机系统出现各种故障是很正常的，数据库中的数据被破坏、丢失也是可能的。当数据库被破坏或数据不可靠时，系统有能力将数据库从错误状态恢复到最近某一时刻的正确状态。

在数据库系统阶段，程序与数据之间的关系可用图 1-3 表示。

从文件系统管理发展到数据库系统管理是信息处理领域的一个重大变化。

在文件系统阶段，人们关注的是系统功能的设计，因此程序设计处于主导地位，数据服从于程序设计；而在数据库系统阶段，数据的结构设计成为信息系统首先关心的问题。

数据库技术经历了以上三个阶段的发展，已有比较成熟的数据库技术，但随着计算机软、硬件的发展，数据库技术仍需不断向前发展。

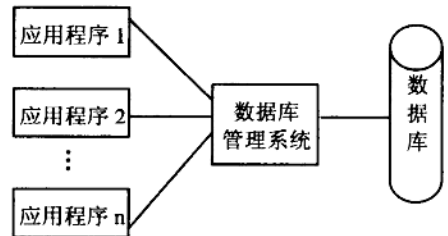


图 1-3 数据库系统阶段

1.1.4 高级数据库阶段

20 世纪 70 年代，层次、网状、关系三大数据库系统奠定了数据库技术的概念、原理和方法。自从 20 世纪 80 年代以来，数据库技术在商业领域的巨大成功刺激了其他领域对数据库技术需求的迅速增长。一方面，这些新的领域为数据库应用开辟了新的天地；另一方面，

在应用中提出的一些新的数据管理的需求也直接推动了数据库技术的研究和发展,尤其是面向对象数据库系统。另外,数据库技术不断与其他计算机分支结合,向高一级的数据库技术发展。例如,数据库技术与分布处理技术相结合,出现了分布式数据库系统;数据库技术与并行处理技术相结合,出现了并行数据库系统。

1. 分布式数据库技术

随着地理上分散的用户对数据共享的要求日益增强,以及计算机网络技术的发展,在传统的集中式数据库系统的基础上产生和发展了分布式数据库系统。

分布式数据库系统不是简单地把集中式数据库安装在不同场地,用网络连接起来便实现了,而是具有自己的性质和特征。

分布式数据库系统主要有以下特点:

(1) 数据的物理分布性和逻辑整体性。数据库的数据物理上分布在各个场地,但逻辑上是一个相互联系的整体。

(2) 场地自治和协调。系统中的每个结点都具有独立性,可以执行局部应用请求(访问本地数据库);每个结点又是整个系统的一部分,可通过网络处理全局的应用请求,即可以执行全局应用(访问异地数据库)。

(3) 各地的计算机通过数据通信网络相联系。本地计算机单独不能胜任的处理任务,可以通过通信网络取得其他数据库和计算机的支持。

(4) 数据的分布透明性。在用户看来,整个数据库仍然是一个集中的数据库,用户不必关心数据的分片,不必关心数据物理位置分布的细节,不必关心数据副本的一致性,分布的实现完全由分布式数据库管理系统来完成。

(5) 适合分布处理的特点,提高系统处理效率和可靠性。因此,数据复制技术是分布式数据库的重要技术。然而,分布式数据库中的这种数据冗余对用户是透明的,即用户不必知道冗余数据的存在,维护各副本的一致性也由系统来负责。

分布式数据库系统兼顾了集中管理和分布处理两个方面,因而具有良好的性能,具体结构如图1-4所示。

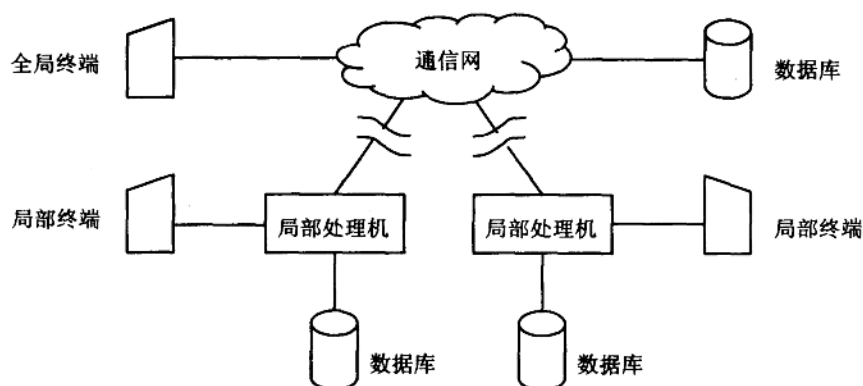


图 1-4 分布式数据库

2. 面向对象数据库技术

在数据处理领域,关系数据库的使用已相当普遍,然而现实世界存在着许多具有更复杂数据结构的实际应用领域,而层次、网状和关系三种模型在这些应用领域显得力不从心。例如,多媒体数据、多维表格数据、CAD 数据等应用问题,都需要更高级的数据库技术来表达,以便于管理、构造与维护大容量的持久数据,并使它们能与大型复杂程序紧密结合,而面向对象数据库正是适应这种形势发展起来的,它是面向对象的程序设计技术与数据库技术相结合的产物。

对象数据库系统的主要特点如下:

(1) 对象数据模型能完整地描述现实世界的数据结构,能表达数据间嵌套、递归的联系。

(2) 具有面向对象技术的封装性(把数据与操作定义在一起)和继承性(继承数据结构和操作)的特点,提高了软件的可重用性。

3. 面向应用领域的数据库技术

数据库技术是计算机软件领域的一个重要分支,经过 30 多年的发展,已形成相当规模的理论体系和实用技术。为了适应数据库应用多元化的要求,在传统数据库基础上,结合各个应用领域的特点,研究适合该应用领域的数据库技术,如数据仓库、工程数据库、统计数据库、科学数据库、空间数据库、地理数据库等。

1.2 数据库的基本概念

下面先介绍一些数据库最常用的术语和基本概念。

1.2.1 数据

数据(Data)是描述事物的符号记录。数字、文字、声音、图形、图像等都是数据。用数据描述的现实世界中的对象可以是实在的事物,如描述一个学生的情况可用学号、姓名、性别、年龄、系别、入学时间等,如下:

(80102, 赵文, 男, 22, 计算机系, 2003)

这里的学生记录就是数据。对于这条记录,若了解其含义,将得到如下信息:赵文是个大学生,男,今年 22 岁,2003 年考入计算机系;若不了解其含义,则无法得到上述信息。可见,数据的形式本身还不能完全表达其内容,需要经过语义解释。因此数据和关于数据的解释是不可分的,数据的解释是对数据含义的说明,数据的含义称为数据的语义,数据与其语义是不可分的。

1.2.2 数据库

数据库(DataBase, DB)是长期存储在计算机内有组织的、共享的数据的集合。数据库中的数据按一定的数据模型组织、描述和存储。它可以供用户共享,具有尽可能小的冗余度和较高的数据独立性,使得数据存储最优,数据最容易操作,并且具有完善的自我保护能力和数据恢复能力。

数据库的特点如下:

(1) 集成性:把某特定应用环境中的各种应用相关的数据及数据之间的联系全部集中地并按照一定的结构形式进行存储,或者说,把数据库看成若干个单个性质的数据文件的联合和统一的数据整体。

(2) 共享性: 数据库的不同块中的数据可为多个不同的用户所共享, 即多个不同的用户, 使用多种不同的语言, 为了不同的应用目的, 同时存取数据库, 甚至同时存取同一块数据, 即多用户系统。

1.2.3 数据库管理系统

数据库管理系统 (DataBase Management System, DBMS) 是位于用户与操作系统之间的一层数据管理软件, 它是数据库系统的核心组成部分, 用户在数据库系统中的一切操作, 包括数据定义、查询、更新及各种控制, 都是通过 DBMS 进行的。DBMS 就是实现把用户意义下的抽象的逻辑数据处理转换成计算机中的具体的物理数据的处理软件, 这给用户带来了很大的方便。它的主要功能包括以下几个方面:

1. 数据定义

DBMS 提供数据定义语言 DDL (Data Define Language), 用户通过它可以方面地对数据库中的数据对象进行定义。例如, 为保证数据库安全而定义的用户口令和存取权限, 为保证正确语义而定义的完整性规则。

2. 数据操纵

DBMS 提供数据操纵语言 DML (Data Manipulation Language) 实现对数据库的基本操作, 包括检索、插入、修改和删除等。SQL 就是 DML 的一种。

3. 数据库运行管理

数据库在建立、运行和维护时由数据库管理系统统一管理和统一控制。DBMS 通过对数据的安全性控制、数据的完整性控制、多用户环境下的并发控制及数据库的恢复, 来确保数据的正确有效和数据库系统的正常运行。

4. 数据库的建立和维护功能

该项功能包括数据库的初始数据的装入、转换, 数据库的转储、恢复、重组织, 系统性能监视、分析等功能。这些功能通常是由一些实用程序完成的。

5. 数据通信

DBMS 提供与其他软件系统进行通信的功能, 实现用户程序与 DBMS 之间的通信, 通常与操作系统协调完成。

1.2.4 数据库系统

数据库系统 (DataBase System, DBS) 是指在计算机系统中引入数据库后的系统, 一般由数据库、数据库管理系统 (及其开发工具)、应用系统、数据库管理员和用户构成 (见图 1-5)。

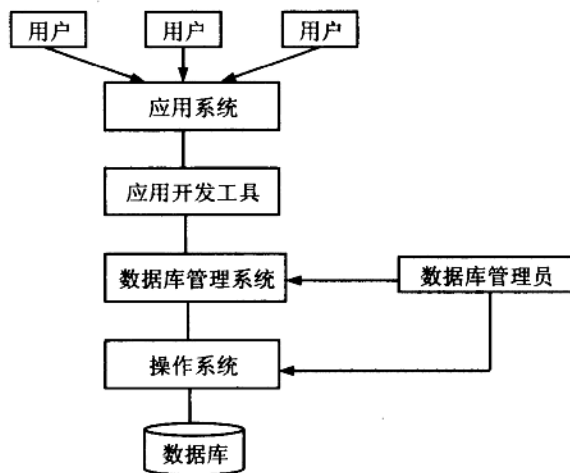


图 1-5 数据库系统

1.2.5 用户

用户 (User) 是指使用数据库的人, 即对数据库的存储、维护和检索等操作。用户大致可分为终端用户、应用程序员和数据库管理员。

1. 终端用户

终端用户 (End User) 主要是使用数据库的各级管理人员、工程技术人员、科研人员, 一般为非计算机专业人员。

2. 应用程序员

应用程序员 (Application Programmer) 负责为终端用户设计和编制应用程序, 以便终端用户对数据库进行存取操作。

3. 数据库管理员

数据库管理员 (Data Base Administrator, DBA) 是数据库所属单位的代表。一个单位决定开发一个数据库系统时, 首先就应确定 DBA 的人选。DBA 不仅应当熟悉系统软件, 还熟悉本单位的业务工作。DBA 应自始至终参加整个数据库系统的研制开发工作, 开发成功后, DBA 将全面负责数据库系统的“管理、维护和正常使用”。其职责如下:

(1) 参与数据库设计的全过程, 决定数据库的结构和内容。

(2) 定义数据的安全性和完整性, 负责分配用户对数据库的使用权限和口令管理。

(3) 监督控制数据库的使用和运行, 改进和重新构造数据库系统。当数据库受到破坏时, 应负责恢复数据库; 当数据库的结构需要改变时, 完成对数据结构的修改。

因此, DBA 不仅要有较高的技术专长和较深的资历, 而且应具有了解和阐明管理要求的能力, 特别对于大型数据库系统, DBA 极为重要。对于常见的微机数据库系统, 通常只有一个用户, 常常不设 DBA, DBA 的职责由应用程序员或终端用户代替。

1.3 数据模型

数据模型 (Data Model) 是专门用来抽象、表示和处理现实世界中的数据和信息的工具。

计算机系统是不能直接处理现实世界的, 现实世界只有数据化后, 才能由计算机系统来处理这些代表现实世界的的数据。为了把现实世界的具体事物及事物之间的联系转换成计算机能够处理的数据, 必须用某种数据模型来抽象和描述这些数据。数据模型是数据库系统的核心。通俗地讲, 数据模型是现实世界的模拟。

数据模型应满足三方面要求: 一是能比较真实地模拟现实世界; 二是容易理解; 三是易在计算机上实现。在数据库系统中针对不同的使用对象和应用目的, 应采用不同的数据模型。

不同的数据模型实际上是用于模型化数据和信息的不同工具。根据模型应用的不同, 可将模型分为两类, 它们分别属于两个不同的层次 (见图 1-6)。

第一类模型是概念模型, 也称信息模型, 它是一种独立于计算机系统的数据模型, 完全不涉及信息在计算机中的表示, 只是用来描述某个特定组织所关心的信息结构。概念模型是按用户的

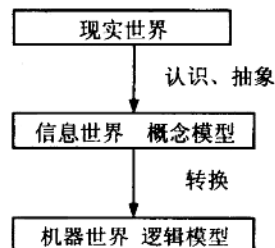


图 1-6 抽象的层次

观点对数据和信息建模，强调其语义表达能力。概念应该简单、清晰、易于用户理解，它是对现实世界的第一层抽象，是用户和数据库设计人员之间进行交流的工具。这一类模型中最著名的是“实体联系模型”。

第二类模型是数据模型，主要包括网状模型、层次模型和关系模型等，它是按计算机系统的观点对数据建模，是直接面向数据库的逻辑结构，是对现实世界的第二层抽象。这类模型直接与 DBMS 有关，称为“逻辑数据模型”，一般又称为“结构数据模型”。这类模型有严格的形式化定义，以便于在计算机系统中实现。它通常有一组严格定义的无二义性语法和语义的数据库语言，人们可以用这种语言来定义、操纵数据库中的数据。

数据模型是数据库系统的核心和基础。各种机器上实现的 DBMS 软件都是基于某种数据模型的。

1.3.1 概念模型

从图 1-6 中可以看出，概念模型实质上是现实世界到机器世界的一个中间层次。

1. 概念模型中的基本概念

概念模型涉及的概念主要有：

(1) 实体

实体 (Entity) 是一个数据对象，指应用中可以区别的客观存在的事物。实体既可以是实际存在的对象，也可以是某种概念，如一个工人、一个学生、一个学校、一个操作流程等都是实体。

(2) 属性

实体所具有的某一特性称为属性 (Attribute)。一个实体可以由若干个属性来描述，如职工实体由职工号、姓名、性别、年龄、职称、部门等属性组成，则 (1010, 陈平, 男, 34, 工程师, 02) 这组属性值就构成了一个具体的职工实体。属性有属性名和属性值之分，如“姓名”是属性名，“陈平”是“姓名”属性的一个属性值。

(3) 实体集

所有属性名完全相同的同类实体的集合称为实体集 (Entity Set)，如全体职工就是一个实体集。为了区分实体集，每个实体集都有一个名称，即实体名。职工实体指的是名为“职工”的实体集，而 (1010, 陈平, 男, 34, 工程师, 02) 是该实体集中的一个实体，同一实体集中没有完全相同的两个实体。

(4) 码

能唯一标识实体的属性或属性集，称为码 (Key)，有时也称为实体标识符，或简称为键，如“职工”实体中的“职工号”属性。

(5) 域

属性的取值范围称为该属性的域 (Domain)，如“职工性别”的属性域为 (男, 女)。

(6) 实体型

实体集的名及其所有属性名的集合，称为实体型 (Entity Type)，如职工 (职工号, 姓名, 性别, 年龄, 职称, 部门) 就是职工实体集的实体型。实体型抽象地刻画了所有同集实体，在不引起混淆的情况下，实体型往往简称为实体。