

普通高等教育“十二五”规划教材

数据库 原理与应用

S

HUJUKU YUANLI YU YINGYONG

林小玲 编著



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”规划教材

数据库原理与应用

林小玲 编 著

机械工业出版社

本书从数据库实际应用与开发的角度出发进行编写,内容涵盖了数据库系统的基本概念、基本理论、设计方法、管理技术,目前流行的数据库管理系统 Microsoft SQL Server 和 Access 的功能与使用方法,以及基于 .NET 的 Web 数据库开发技术等。

本书的主要特点是:全面介绍数据库技术的基本原理与基本概念;系统阐述数据库系统的开发方法;完整地介绍了中小型数据库管理系统 Access 及适用于网络环境下进行数据库应用开发的 Microsoft SQL Server 系统的功能及开发方法;应用两个典型实用的示例把教程中的原理与开发技术进行串联,以此来介绍应用程序开发的全过程。

本书内容深入浅出,可作为相关专业的教材,也可作为工程技术人员的自学参考书。

图书在版编目 (CIP) 数据

数据库原理与应用/林小玲编著. —北京:机械工业出版社, 2011. 8
普通高等教育“十二五”规划教材
ISBN 978-7-111-34847-4

I. ①数… II. ①林… III. ①数据库系统—高等学校—教材
IV. ①TP311.13

中国版本图书馆 CIP 数据核字 (2011) 第 157684 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
策划编辑:于苏华 责任编辑:于苏华 郭娟 任正一
版式设计:霍永明 责任校对:常天培
封面设计:张静 责任印制:李妍
北京诚信伟业印刷有限公司印刷

2011 年 10 月第 1 版第 1 次印刷

184mm × 260mm · 18.5 印张 · 469 千字

标准书号: ISBN 978-7-111-34847-4

定价: 35.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

前 言

数据库技术从诞生到现在，在不到半个世纪的时间里，逐渐形成了坚实的理论基础、成熟的商业产品和广泛的应用领域，使得数据库成为一个研究者众多且被广泛关注的研究领域。

本书在介绍数据库原理理论概念的基础上，将理论与实际相结合作为重点，力求使读者通过学习本书，能够完成系统的需求分析、数据库的概念设计、逻辑设计、物理设计、安全设计、数据库实现等设计开发全过程，最终实现一个较为完整的、能够反映应用需求的管理信息系统，以此来培养和提高读者的数据库开发能力和水平。

本书在第1~6章都附有习题，以加强读者对所学知识的理解，及时掌握重点和难点。本书结合数据库设计的过程，在相应的章节中编写了分析设计题，以便读者掌握各个设计环节的基本方法、基本技能和设计要点。

本书采用“案例驱动”的编写方式，用案例来贯穿理论与设计的学习过程，内容涵盖原理、设计、使用、管理和维护等各个层面的知识。

本书的主要特点如下：

1. 全面介绍数据库技术的发展、基本原理与基本概念。
2. 应用小结、重点与难点强调概念的理解和分析方法之间的关系，以此来概括该章的知识体系结构。
3. 系统地介绍了数据库系统的开发流程和方法。
4. 完整地介绍了中小型数据库管理系统 Access，以及适用于网络环境下进行数据库应用开发的 Microsoft SQL Server 的功能及具体的开发方法。
5. 应用两个具有代表性的案例把书中的数据库原理与开发技术进行串联，以此来介绍应用程序开发的全过程。

本书是作者多年从事数据库教学的经验和体会的总结。在本书的编写过程中，得到了上海大学教务处、自动化系许多同志的关心与支持，在此向他们表示衷心的感谢。

由于编者的学识和水平有限，疏漏之处在所难免，恳请读者给予批评指正。

林小玲

目 录

前言

第 1 章 绪论	1	3.1.2 SQL 语言的基本概念	38
1.1 数据库概述	1	3.2 数据定义	39
1.1.1 基本概念	1	3.2.1 定义、删除与修改基本表	39
1.1.2 数据管理技术的产生和发展	2	3.2.2 建立与删除索引	42
1.1.3 数据库技术的研究领域	5	3.3 查询	42
1.2 数据模型	6	3.3.1 单表查询	44
1.2.1 数据模型的组成要素	6	3.3.2 连接查询	49
1.2.2 概念模型	7	3.3.3 嵌套查询	50
1.2.3 常用的数据模型	9	3.3.4 集合查询	53
1.3 数据库的体系结构	10	3.4 数据更新	54
1.3.1 数据库系统模式的概念	10	3.4.1 插入数据	54
1.3.2 数据库系统的三级模式结构	10	3.4.2 修改数据	55
1.3.3 数据库的二级映像功能 与数据独立性	11	3.4.3 删除数据	55
1.3.4 数据库系统外部的体系结构	12	3.5 视图	56
1.4 数据库系统的组成	15	3.5.1 定义视图	56
1.5 数据库技术的研究领域	17	3.5.2 查询视图	57
1.6 本章小结	18	3.5.3 更新视图	58
习题	19	3.5.4 视图的作用	58
第 2 章 关系数据库	21	3.6 数据的控制	59
2.1 关系模型概述	21	3.7 嵌入式 SQL	60
2.2 关系数据结构及形式化定义	22	3.7.1 嵌入式 SQL 的一般形式	60
2.2.1 关系	22	3.7.2 嵌入式 SQL 语句与主 语言之间的通信	60
2.2.2 关系模式	25	3.7.3 不用游标的 SQL 语句	62
2.2.3 关系数据库的定义	26	3.7.4 使用游标的 SQL 语句	62
2.3 关系的完整性	26	3.7.5 动态 SQL 简介	62
2.3.1 实体完整性	27	3.8 本章小结	63
2.3.2 参照完整性	27	习题	64
2.3.3 用户定义的完整性	28	第 4 章 关系数据理论	71
2.4 关系代数	29	4.1 概述	71
2.4.1 传统的集合运算	29	4.2 关系规范化	72
2.4.2 专门的关系运算	30	4.2.1 函数依赖	72
2.5 本章小结	32	4.2.2 关系模式中的码	73
习题	33	4.2.3 范式	74
第 3 章 关系数据库标准语言 SQL	37	4.2.4 规范化	76
3.1 SQL 概述	37	4.3 关系模式的分解	77
3.1.1 SQL 的特点	37	4.4 本章小结	78

习题	78	6.2.3 活锁和死锁	122
第5章 数据库设计	83	6.2.4 并发调度的可串行性	123
5.1 数据库设计概述	83	6.2.5 两段锁协议	124
5.1.1 数据库设计的特点	83	6.2.6 封锁的粒度	124
5.1.2 数据库设计方法	84	习题	125
5.1.3 数据库设计的基本步骤	85	第7章 数据库开发技术	129
5.2 需求分析	86	7.1 数据库系统开发方法	129
5.2.1 需求分析的任务	86	7.1.1 瀑布式开发方法	129
5.2.2 需求分析的方法	87	7.1.2 快速原型开发方法	130
5.2.3 数据字典	88	7.1.3 软件系统开发中用到的技术	131
5.3 概念结构设计	89	7.2 数据库管理系统 Access	132
5.3.1 概念结构设计的特点	89	7.2.1 Access 基础	132
5.3.2 概念结构设计的方法与步骤	89	7.2.2 Access 数据库基本操作	134
5.3.3 数据抽象与局部视图设计	91	7.2.3 Access 数据库的查询	143
5.3.4 视图的集成	96	7.2.4 Access 数据库窗体	147
5.4 逻辑结构设计	99	7.2.5 Access 数据库的报表	153
5.4.1 E-R 图向关系模型的转换	99	7.2.6 Access 数据库宏的应用	155
5.4.2 数据模型的优化	101	7.2.7 Access 系统工具栏的设计	157
5.4.3 设计用户子模式	102	7.3 数据库管理系统	158
5.5 数据库的物理设计	102	7.3.1 SQL Server 简介	159
5.5.1 数据库的物理设计的 内容和方法	103	7.3.2 数据库的管理	165
5.5.2 关系模式存取方法选择	103	7.3.3 表的管理	180
5.5.3 确定数据库的存储结构	105	7.3.4 视图	197
5.5.4 评价物理结构	105	7.3.5 存储过程	199
5.6 数据库的实施和维护	106	7.3.6 触发器	205
5.6.1 数据的载入和应用程序的调试	106	7.4 本章小结	210
5.6.2 数据库的试运行	106	第8章 基于 .NET 的 Web 数据库 开发技术	211
5.6.3 数据库的运行和维护	107	8.1 Web 数据库概述	211
5.7 本章小结	108	8.1.1 Web 中的基本概念	211
习题	108	8.1.2 静态网页与动态网页	213
第6章 数据库的管理	113	8.1.3 动态网站开发技术概述	214
6.1 数据库恢复技术	113	8.2 ASP.NET 基础知识	216
6.1.1 事务的基本概念	113	8.2.1 .NET Framework 概述	217
6.1.2 数据库恢复概述	114	8.2.2 Visual Studio .NET 的特色	219
6.1.3 数据库故障的类型	114	8.2.3 ASP.NET 运行和开发环境	220
6.1.4 数据库故障恢复的实现技术	116	8.2.4 ASP.NET 页面、控件与事件	221
6.1.5 数据库故障恢复策略	118	8.2.5 ASP.NET Web 窗体	225
6.1.6 数据库镜像	119	8.2.6 ASP.NET TextBox Button 控件	225
6.1.7 Oracle 的恢复技术	120	8.3 数据库连接	226
6.2 并发控制	120	8.3.1 ADO.NET 概述	227
6.2.1 并发控制概述	120	8.3.2 连接数据库	229
6.2.2 封锁	122	8.4 数据集	238

8.4.1 使用 DataTables、DataColumns 和 DataRows	239	9.1.2 学生信息管理系统的 编程与实现	260
8.4.2 DataReader 类	241	9.2 酒店库存管理系统	271
8.4.3 DataGrid 和 GridView 控件	243	9.2.1 酒店库存管理系统的设计	271
8.4.4 DataList 控件	245	9.2.2 酒店库存管理系统的 编程与实现	275
第9章 综合应用编程示例	257	参考文献	289
9.1 学生信息管理系统	257		
9.1.1 学生信息管理系统的设计	257		

第 1 章 绪 论

数据库是数据管理的最新技术，是计算机科学的重要分支。当今，信息资源已成为各个部门的重要财富和资源。建立一个满足各级部门信息处理要求的、行之有效的信息系统也成为企业或组织生存和发展的重要条件。目前，基于数据库技术的计算机应用已成为计算机应用的主流，管理信息系统、办公信息系统、银行信息系统等都属于这一类的应用分支。对于一个国家来说，数据库的建设规模、数据库信息量的大小和使用频度已成为衡量这个国家信息化程度的重要标志之一。

数据库技术是一门研究如何应用计算机与通信技术对数据进行科学有效的管理和安全高效的用的数据管理技术。

1.1 数据库概述

1.1.1 基本概念

数据、数据库、数据库管理系统和数据库系统是数据库技术密切相关的 4 个基本概念。

1. 数据

数据是数据库存储的基本对象，可以是数字（传统和狭义的数据理解），也可以是文字、声音、图像等。数据有多种表现形式，它们都可以经过数字化后存入计算机。

数据是信息的载体。所谓信息，就是指真实的可传播的消息。比如，一个学生的信息。

在日常生活中，为了交流信息、了解世界，我们需要对事物进行描述，通常采用的是自然语言。而在计算机中，为了存储和处理这些事物，就需要抽取对事物感兴趣的特征，组成一个记录进行描述，这个记录就是数据。比如，学生档案，如果人们关心的是学生的姓名、性别、年龄、出生年月、籍贯、所在系别、入学时间，则可以这样描述：

李明，男，21，1972，江苏，计算机系，1990 等。

所以，我们看到数据是对所感兴趣的特征的描述，而不是对事物的所有特征的描述。

对于上面的记录（数据），了解其含义的人会得到下面的信息：李明是个大学生，1972 年生，男，江苏人，1990 年考入计算机系。而不了解其语义的人则无法理解其含义。可见，数据的形式还不能完全表达其内容，需要经过解释。

数据和关于数据的解释是不可分的。数据的解释是指对数据含义的说明，数据的含义称为数据的语义，数据与其语义是不可分的。

2. 数据库（DataBase，DB）

人们收集并抽取出一个应用所需要的数据后，应将其保存起来，以供进一步加工处理，进一步提取有用的信息。在科学技术飞速发展的今天，数据量急剧增大。过去人们把数据存放在文件柜里，现在人们借助于计算机和数据技术，可以科学地保存和管理大量的数据，以便充分地利用这些宝贵的信息资源。

数据库，顾名思义，就是存放数据的仓库。只是仓库在计算机存储设备上，而且数据是按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性和易扩展性，并可以为各种用户共享。

数据库对数据的高效处理，主要是借助于数据库技术进行的。

3. 数据库管理系统 (DataBase Management System, DBMS)

收集并抽取一个应用所需要的数据之后，如何科学地组织这些数据，并将其存储在数据库中，高效地处理这些数据，需要一个软件系统，即数据库管理系统。

它是位于用户和操作系统之间的一层数据管理软件。

数据库在建立、运用和维护时，由数据库管理系统统一管理、统一控制。数据库管理系统使用户能方便地定义数据和操纵数据，并能够保证数据的安全性、完整性、多用户对数据的并发使用，以及发生故障后的系统恢复。

4. 数据库系统 (DataBase System, DBS)

数据库系统是指在计算机系统中，引入数据库后的系统的构成，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。

应当特别指出的是，数据库的建立、使用和维护等工作，只靠一个 DBMS 远远不够，还要有专门的人员来完成，这些被称为数据库管理员 (DataBase Administrator, DBA)

在不引起混淆的情况下，人们常常把数据库系统简称为数据库，如图 1-1 所示。

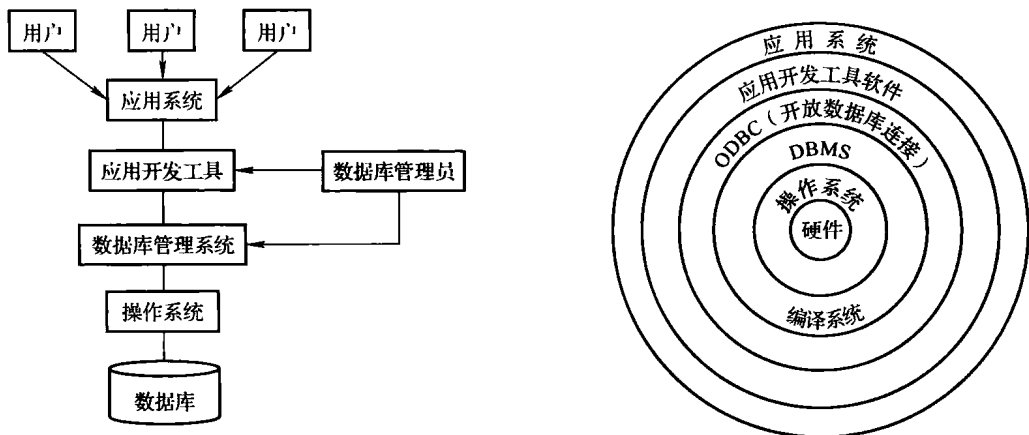


图 1-1 数据库系统

1.1.2 数据管理技术的产生和发展

数据库技术是应数据管理任务的需要而产生的。数据管理是指如何对数据进行分类、组织、编码、存储、检索和维护，它是数据处理的中心问题。随着计算机硬件和软件的发展，数据管理经历了人工管理、文件系统和数据库系统 3 个发展阶段。

1. 人工管理阶段

20 世纪中期以前，计算机主要用于科学计算。当时的硬件状况是，外存只有纸带、卡片、磁带，没有磁盘等直接存取的存储设备；软件的状况是，没有操作系统，没有管理数据的软件；数据处理的方式是批处理。该阶段具有以下特点：

1) 数据不保存。由于当时计算机主要用于科学计算，一般不需要将数据长期保存，只

是在计算某一课题时将数据输入，用完就撤走。不仅对用户数据如此处置，对系统软件有时也采用这种方式。

2) 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。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，程序员的负担很重。

3) 由于数据是面向应用的，一组数据只能对应一个程序。当多个应用程序涉及某些相同的数据时，由于必须各自定义，无法互相利用、互相参照，因此，程序与程序之间有大量的数据冗余。

4) 数据的逻辑结构或物理结构发生变化后，必须对程序做出相应的修改，这就进一步加重了程序员的负担。

在人工管理阶段，程序与数据之间是一一对应的关系，如图 1-2 所示。

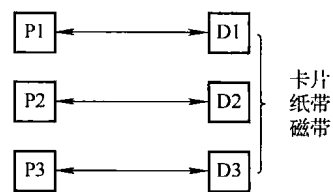


图 1-2 人工管理阶段程序与数据之间的关系

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代中期，在硬件方面，已经有了磁盘、磁鼓等直接存取数据的存储设备；在软件方面，操作系统中已经有了专门的数据管理的软件（即文件系统），在处理方式上不仅有了批处理，而且能够实现联机实时处理。文件系统阶段具有如下特点：

1) 由于计算机大量用于数据处理，数据需要长期保留在外存中，以便反复进行查询、修改、插入和删除等操作。

2) 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，使应用程序与数据之间有了一定的独立性。程序员可以不必过多地考虑物理细节，从而将精力集中在算法上。此外，数据在存储上的改变，不一定反映在程序上，这样就大大节省了维护程序的工作量。

3) 数据的共享性差。在文件系统中，一个文件基本上是对应于一个应用程序，即文件仍然是面向应用的。当不同的应用程序具有大部分相同的相同数据时，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。同时，由于相同数据的重复存储、各自管理，给数据的修改和维护带来了困难，容易造成数据的不一致。

4) 数据的独立性低。文件系统中的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，因此，要想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。一旦数据结构改变，必须修改应用程序，修改文件结构的定义。而应用程序的改变，也将引起文件的数据结构的改变。因此，数据与程序之间仍缺乏独立性。

可见，文件系统仍然是一个不具有弹性的无结构的数据集合，即文件之间是孤立的，不能反映现实世界事物之间的内在联系。

文件系统阶段应用程序与数据之间的关系如图 1-3 所示。

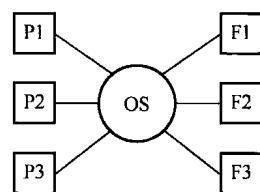


图 1-3 文件系统阶段应用程序与数据之间的关系

3. 数据库系统阶段

20 世纪 60 年代后期，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；此时硬件价格下降，而软件价格上升，使开发和维护系统软件的成本增加。文件系统的数据库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。数据库管理系统的特点如下：

(1) 数据的共享性高、冗余少

这是数据库系统阶段的最大改进，数据不再面向某个应用程序而是面向整个系统，所有用户可同时存取库中的数据。这样便减少了数据冗余，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。

(2) 数据结构化

按照某种数据模型，将整个系统的各种数据组织到一个结构化的数据库中，系统的数据不是一盘散沙，可表示出数据之间的有机关联。例如，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3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，必须通过编写程序来实现不同文件之间的数据联系。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，上述查询的问题就可以非常容易地联机实现。

(3) 数据的独立性高

数据的独立性包括逻辑独立性和物理独立性

数据的逻辑独立性是指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，所以，应用程序不必修改，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当数据的存储结构改变时，数据的逻辑结构不变，从而应用程序也不必改变。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) 具有统一的数据控制功能

数据库是为多个用户和应用程序所共享的，因此，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。为确保数据库中数据的正确有效以及数据库系统的有效运行，数据库管理系统提供4个方面的数据控制功能。

1) 数据的安全性 (Security) 控制：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。

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：系统通过设置一些完整性规则以确保数据的正确性、有效性和相容性。

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。

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

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, 防止相互干扰而提供给用户不正确的数据, 并使数据库受到破坏。

4) 数据恢复 (Recovery): 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。

数据库系统的实现示例如图 1-4 所示。

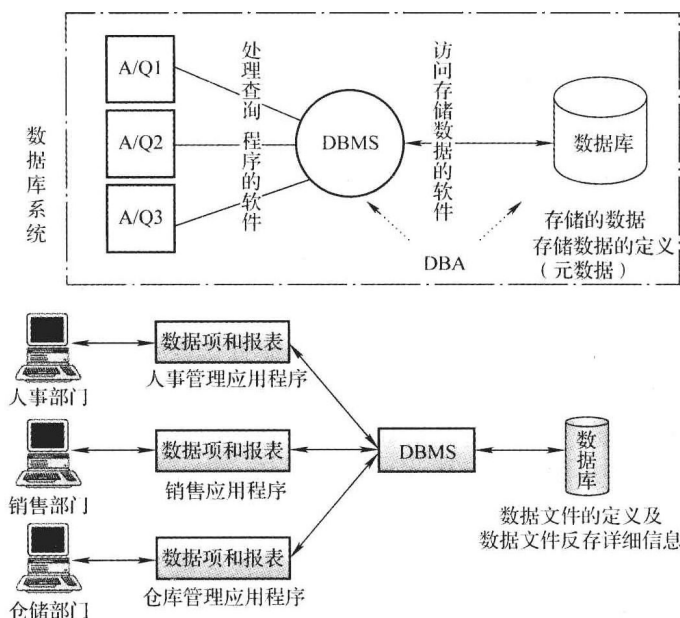


图 1-4 数据库系统的实现示例

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 3 个阶段, 已经比较成熟, 但随着计算机软硬件的发展, 数据库技术仍在不断向前发展。

1.1.3 数据库技术的研究领域

数据库学科的研究范围主要包括 3 个领域。

1. 数据库管理系统软件的研制

DBMS 是数据库系统的基础。DBMS 的研制包括研制 DBMS 本身及以 DBMS 为核心的一组相互联系的软件系统, 包括工具软件和中间件。研制的目标是提高系统的性能和提高用户的生产率。

2. 数据库设计

数据库设计的研究包括以下内容:

- 1) 数据库的设计方法、设计工具和设计理论的研究。
- 2) 数据模型和数据建模的研究。
- 3) 计算机辅助数据库设计及其软件系统的研究。
- 4) 数据库设计规范和研究等。

3. 数据库理论

数据库理论的研究主要集中在关系规范化理论、关系数据理论等。

近年来，随着人工智能与数据库理论的结合以及并行计算技术的发展，数据库逻辑演绎和知识推理、并行算法等都已成为新的研究方向。

随着数据库应用领域的不断扩展，计算机技术的迅猛发展，数据库技术与人工智能技术、网络通信技术、并行计算技术等相互渗透、相互结合，使数据库技术不断涌现出新的研究方向。

1.2 数据模型

模型，特别是具体模型，人们并不陌生。一张地图，一组建筑设计沙盘，一架精致的航模飞机都是具体的模型。模型是现实世界特征的模拟和抽象。数据模型（Data Model）也是一种模型，它是现实世界数据特征的抽象。

数据库是某个企业、组织或部门所涉及的数据的一个综合，它不仅要反映数据本身的内容，而且要反映数据之间的联系。由于计算机不可能直接处理现实世界的具体事物，所以，人们必须事先把具体的事物转换成计算机能够处理的数据。

在数据库中，用数据模型这个工具来抽象、表示和处理现实世界中的数据 and 信息。通俗地讲，数据模型是现实世界数据特征的抽象，是现实世界的模拟，是对客观事物及其联系的数据的描述。

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；二是容易为人所理解；三是便于在计算机上实现。一种数据模型要很好地满足这三方面的要求在目前尚且困难。在数据库系统中针对不同的使用对象和应用目的，采用不同的数据模型。

不同的数据模型实际上是提供给我们模型化数据和信息的不同工具。根据模型应用的不同目的，可以将这些模型划分为两类，它们属于两个不同的层次。

一类模型是概念模型，也称信息模型，它是按用户的观点来对数据和信息建模，主要用于数据库设计。另一类模型是数据模型，主要包括网状模型、层次模型、关系模型等，它是按计算机系统的观点对数据建模，主要用于 DBMS 的实现。

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。

为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型，人们常常首先将现实世界抽象为信息世界，然后将信息世界转换为机器世界。也就是说，首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统，不是某一个 DBMS 支持的数据模型，而是概念级的模型；然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型。可以说，数据模型是现实世界的两级抽象的结果，如图 1-5 所示。

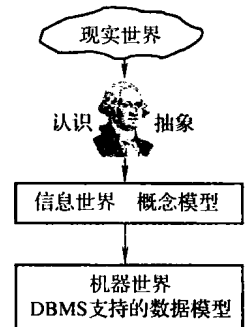


图 1-5 从现实世界到机器世界的过程

1.2.1 数据模型的组成要素

任何一种数据模型都是严格定义的概念和集合。这些概念和集合，必须能够精确地描述系统的静态特性、动态特性和完整性约束条件。因此，数据模型通常都是由数据结构、数据操作和完整性约束 3 个要素组成的。

1. 数据结构

数据结构是所研究的对象类型的集合。这些对象是数据库的组成成分，它们包括两类，一类是与数据类型、内容、性质有关的对象，例如，网状模型中的数据项、记录，关系模型中的域、属性、关系等；另一类是与数据之间联系有关的对象，例如，网状模型中的系型 (Set Type)。

数据结构是刻画一个数据模型性质最重要的方面。因此在数据库系统中，人们通常按照其数据结构的类型来命名数据模型。例如，层次结构、网状结构和关系结构的数据模型分别命名为层次模型、网状模型和关系模型。

2. 数据操作

数据操作用于描述系统的动态特性，是对数据库中各种数据操作的集合，包括操作及相应的操作规则。数据库主要有检索和更新（包括插入、删除、修改）两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级）以及实现操作的语言。数据操作是对系统动态特性的描述，例如，数据的检索、插入、删除和修改等。

3. 数据的约束条件

数据的约束条件是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容。

数据模型应该反映和规定本数据模型必须遵守的基本通用的完整性约束条件。例如，在关系模型中，任何关系必须满足实体完整性和参照完整性两个条件（第2章将详细介绍这两个完整性约束条件）。

此外，数据模型还应该提供定义完整性约束条件的机制，以反映具体应用所涉及的数据必须遵守的特定的语义约束条件。例如，在学校的数据库中规定大学生入学年龄不得超过30岁，硕士研究生入学年龄不得超过38岁，学生累计成绩不得有3门以上不及格等。

1.2.2 概念模型

概念模型实际上是现实世界到信息世界的第一层抽象，是用户与数据库设计人员之间进行交流的语言。因此，一方面，概念模型应该具有较强的语义表达能力，可以方便、直接地表达应用中的各种语义；另一方面，还应该简单、清晰、易于用户理解。

1. 常用的主要概念

在信息世界中，常用的主要概念如下：

(1) 实体 (Entity)

客观存在并且可以相互区别的“事物”称为实体，实体可以是可触及的对象，如一个学生、一本书、一辆汽车；也可以是抽象的概念或联系，如一堂课、一次比赛、教师与所在系的工作关系（即某位老师在某系）等都是实体。

(2) 属性 (Attributes)

实体的某一特性称为属性。例如，学生实体有学号、姓名、年龄、性别、系等方面的属性。属性有“型”和“值”之分，“型”即为属性名，如姓名、年龄、性别是属性的型；“值”即为属性的具体内容，如（990001，张立，20，男，计算机）这些属性值的集合表示了一个学生实体。

(3) 实体型 (Entity Type)

若干个属性型组成的集合可以表示一个实体的类型, 简称实体型, 如学生 (学号, 姓名, 年龄, 性别, 系) 就是一个实体型。

(4) 实体集 (Entity Set)

同型实体的集合称为实体集, 如所有的学生、所有的课程等。

(5) 键 (Key)

能唯一标识一个实体的属性或属性集称为实体的键, 如学生的学号, 而学生的姓名可能有重名, 不能作为学生实体的键。

(6) 域 (Domain)

属性值的取值范围称为该属性的域, 如学号的域为 6 位整数, 姓名的域为字符串集合, 年龄的域为小于 40 的整数, 性别的域为 (男, 女)。

(7) 联系 (Relationship)

在现实世界中, 事物内部以及事物之间是有联系的, 这些联系同样也要抽象和反映到信息世界中, 被抽象为实体型内部的联系和实体型之间的联系。

实体型内部的联系通常是指组成实体的各属性之间的联系; 实体型之间的联系通常是指不同实体集之间的联系。

反映实体型及其联系的结构形式称为实体模型, 也称为信息模型, 它是现实世界及其联系的抽象表示。

两个实体型之间的联系可以分为三类。

(1) 一对一联系 (1 : 1)

实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应, 反之亦然, 则称实体集 A 与实体集 B 为一对一的联系, 记作 1 : 1。例如, 班级与班长, 观众与座位, 病人与床位。

(2) 一对多联系 (1 : n)

实体集 A 中的一个实体与实体集 B 中的多个实体相对应, 反之, 实体集 B 中的一个实体至多与实体集 A 中的一个实体相对应, 记作 1 : n。例如, 班级与学生, 公司与职员, 省与市。

(3) 多对多 (m : n)

实体集 A 中的一个实体与实体集 B 中的多个实体相对应, 反之, 实体集 B 中的一个实体与实体集 A 中的多个实体相对应, 记作 (m : n)。例如, 教师与学生, 学生与课程, 工厂与产品。一个教师可以教授多个学生, 一个学生可以有多位教师; 一个工厂可以生产多种产品, 同一种产品可以有多个工厂生产。

实际上, 一对一联系是一对多联系的特例, 而一对多联系又是多对多联系的特例。实体联系示例如图 1-6 所示。

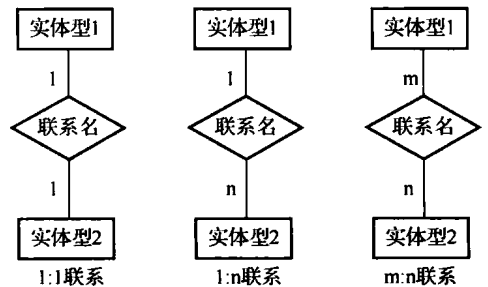


图 1-6 实体联系示例

2. 概念模型的表示方法

概念模型是对信息世界建模, 所以, 概念模型应该能够方便、准确地表示出信息世界的常用概念。实体-联系方法 (Entity-Relationship Approach) 是一种最常用的表示概念模型的方法。该方法用 E-R 图来描述现实世界的概念模型, E-R 图提供了实体型、属性和联系的方法, 如图 1-7 所示。

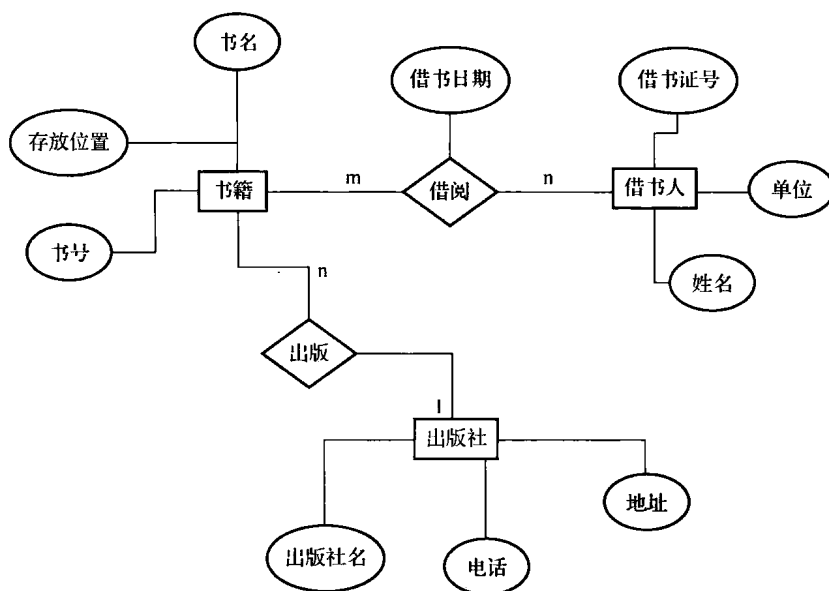


图 1-7 E-R 模型图

1) 用长方形表示实体型，在框内写上实体名。

2) 用椭圆形表示实体的属性，并用无向边把实体与其属性连接起来。

3) 用菱形表示实体间的联系，菱形框内写上联系名，用无向边把菱形分别与有关实体连接，在无向边旁标上联系的类型（1:1, 1:n, 或 m:n）。若实体之间的联系也有属性，则属性和菱形也用无向边连接。

在图 1-7 所示的 E-R 图中，注意联系“借阅”具有借阅日期和还书日期属性。有时为了清晰起见，还可以将各实体及其属性、实体及其联系分别用 E-R 图表示。

实体-联系方法（E-R）是抽象和描述现实世界的有用工具。用 E-R 图表示的概念模型独立于具体的 DBMS 所支持的数据模型，它是各种数据模型的共同基础，因此，它比数据模型更一般、更抽象、更接近现实世界。

有关如何认识和分析现实世界，从中抽取实体和实体之间的联系，建立概念模型的方法，将在后面的相关章节中详细介绍。

1.2.3 常用的数据模型

数据模型是按计算机系统的观点对数据建模，是对实体型的描述框架，提供了信息表示和操作的手段。数据模型的好坏，直接影响数据库的性能。数据模型的选择，是设计数据库的一项首要任务。

目前最常用的数据模型有层次模型（Hierarchical Model）、网状模型（Network Model）和关系模型（Relational Model）。这 3 种数据模型的根本区别在于数据结构不同，即数据之间联系的表示方式不同。

1) 层次模型是用“树结构”来表示数据之间的联系。

2) 网状模型是用“图结构”来表示数据之间的联系。

3) 关系模型是用“二维表”来表示数据之间的联系。

其中层次模型和网状模型是早期的数据模型，统称为非关系模型。

在非关系模型中，实体用记录表示，实体之间的联系转换成记录之间的两两联系。非关系模型的基本单位是基本层次联系。所谓基本层次联系，是指两个记录以及它们之间的一对多（包括一对一）的联系。

20世纪70年代至80年代初，非关系模型的数据库系统非常流行，在数据库系统产品中占据了主导地位，现在已逐渐被关系模型的数据库系统取代。但在美国及一些国家，由于早期开发的应用系统都是基于层次数据库或网状数据库系统，因此，目前层次数据库或网状数据库的系统仍然很多。

20世纪80年代以来，面向对象的方法和技术在计算机各个领域，包括程序设计语言、软件工程、计算机硬件等各方面都产生了深远的影响，出现了一种新的数据模型——面向对象的数据模型。

1.3 数据库的体系结构

考察数据库系统的结构可以从多种不同的层次或不同的角度来考。

从数据库管理系统的角度看，数据库系统通常采用三级模式结构。这是数据库管理系统内部的系统结构。

从数据库最终用户的角度看，数据库系统的结构分为单用户结构、主从式结构、分布式结构和客户机/服务器结构。这是数据库系统外部的体系结构。

1.3.1 数据库系统模式的概念

在数据模型中有“型”（Type）和“值”（Value）的概念。型是指对某一类数据的结构和属性的说明，值是型的一个具体赋值。例如，学生记录定义为（学号，姓名，性别，系别，年龄，籍贯）这样的记录型，而（900201，李明，男，计算机，22，江苏）则是该记录型的一个记录值。

模式（Schema）是数据库中全体数据的逻辑结构和特征的描述，它仅仅涉及型的描述，不涉及具体的值。模式的一个具体值称为模式的一个实例（Instance）。同一个模式可以有多个实例。模式是相对稳定的，而实例是相对变动的，因为数据库中的数据是在不断更新的。模式反映的是数据的结构及其联系，而实例反映的是数据库某一时刻的状态。

虽然实际的数据库管理系统产品种类很多，它们支持不同的数据模型，使用不同的数据库语言，建立在不同的操作系统之上，数据的存储结构也各不相同，但它们在体系结构上通常都具有相同的特征，即采用三级模式结构（早期计算机上的小型数据库系统除外），并提供两级映像功能。

1.3.2 数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由外模式、模式和内模式三级构成的，如图1-8所示。

1. 模式（Schema）

模式也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。它是数据库系统模式结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序所使用的应用开发工具及高级程序设计语言（如C、COBOL、Fortran）无关。