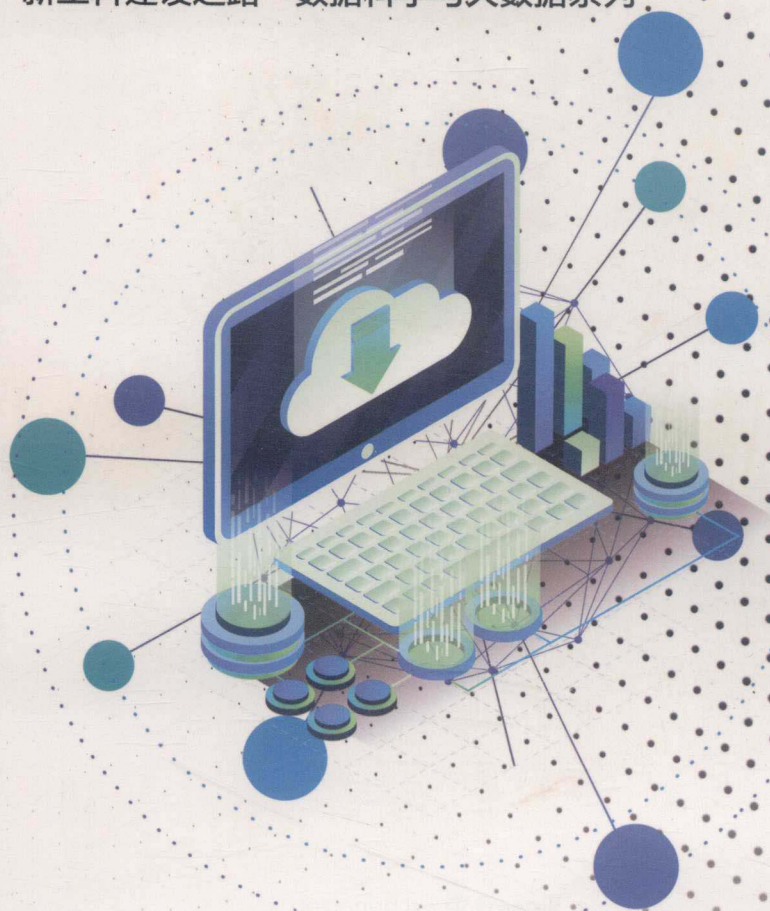



普通高等教育“十三五”规划教材  
新工科建设之路·数据科学与大数据系列




# 数据库系统及应用

(第 3 版)

魏祖宽 郑莉华 牛新征 孙 明 编著

 中国工信出版集团

 电子工业出版社  
http://www.phei.com.cn

普通高等教育“十三五”规划教材  
新工科建设之路·数据科学与大数据系列

# 数据库系统及应用

## (第3版)

魏祖宽 郑莉华 牛新征 孙 明 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书从实用性和先进性出发,通过一个完整的数据库应用实例和数据,全面介绍数据库的基本理论、数据库的系统管理及数据库的设计与开发技术。全书共 14 章,主要内容包括:数据库系统概论、关系数据模型、关系数据库标准 SQL 语言、查询处理优化、事务与并发控制、数据库恢复技术、数据库安全、规范化理论、数据库设计方法、数据库新技术、典型商业数据库和选型。附录是完整的数据库实例和数据,以及实验指导书。本书提供电子课件、实验指导、程序代码、习题参考答案和实例数据库文件等。

本书可作为高等学校计算机和软件工程专业本科及研究生的教材,也可供相关领域的技术和管理人员学习、参考。

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

数据库系统及应用 / 魏祖宽等编著. —3 版. —北京:电子工业出版社,2020.3

ISBN 978-7-121-38178-2

I. ①数… II. ①魏… 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(2019)第 289896 号

策划编辑:王羽佳

责任编辑:底波

印刷:北京盛通商印快线网络科技有限公司

装订:北京盛通商印快线网络科技有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开本:787×1092 1/16 印张:25 字数:739.2 千字

版次:2008 年 8 月第 1 版

2020 年 3 月第 3 版

印次:2020 年 3 月第 1 次印刷

定 价:69.90 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式:(010) 88254535, [wylj@phei.com.cn](mailto:wylj@phei.com.cn)。

# 前 言

计算机技术的发展不仅极大地促进了科学技术的发展,而且明显加快了经济信息化和社会信息化的进程。因此,计算机教育在各国都备受重视,具备计算机知识与使用能力已成为 21 世纪人才的基本素质之一。

数据库应用技术是其中的核心技术之一,以其为核心的各种数据库应用管理,无可争议地改变了政府部门和企事业单位的运营和管理方式。随着数据库应用广度和深度的扩展,不仅是计算机和信息技术行业,而且包括技术管理、工程管理甚至决策人员在内的众多行业都开始关心数据库技术。

为了适应各高校计算机学科正在开展的课程体系与教学内容的改革,及时反映相关研究成果,积极探索适应 21 世纪计算机人才培养的教学模式,我们编写了这本数据库应用技术的教材。

本教材具有如下特色。

1. 根据读者的层次分类。将数据库技术分为数据库基础知识、数据库管理技术、数据库应用技术、现代数据库技术及主流商业数据库四部分,且内容上保持连贯性。读者可以根据自身需求选择适当的内容阅读,不同层次的读者可以从不同的深度学习数据库知识。这样就使得本教材更加符合软件工程应用的特点。

2. 面向软件工程理念,采用工程应用型学习方法,即“提出问题→解决问题→应用分析”的问题驱动方式,突出学生主动探究在整个教学中的作用。

3. 在内容描述上,我们换位思考,站在学生的角度阐述概念和理论,避免堆砌大量学生不常用的专业词汇,使得整个教材通俗易懂。

4. 在内容组织上,以一个典型的数据库应用系统(简化的医院管理数据库 HIS)为案例,在 MS SQL Server 平台上,以理论和实际相结合的方式,讲解数据库的概念和应用开发技术,以期达到高效的学习效果。

5. 本教材的内容分为四部分:①数据库基础知识,讲述数据库的基本概念和理论知识,这部分是基础知识,面向所有读者;②数据库管理技术,讲述数据库维护管理技术,面向关心数据库维护的读者;③数据库应用技术,讲述数据库的设计开发技术,面向关心行业应用系统开发的读者;④现代数据库技术及主流商业数据库,介绍数据库技术的前沿热点及主要的数据库管理系统的商业产品,面向关心数据库发展动向及实际数据库产品的读者。

6. 本教材注重将计算机理论知识和现实中的工程应用相结合,适当引入数据库技术的最新发展,保持了教学内容的先进性和实用性。本教材源于基础教育的教学实践及科研实践中的许多心得体会,凝聚了工作在教学和科研第一线教师多年的教学与科研成果。

通过学习本教材,你可以了解:

- 数据库的概念、组成结构等基础知识;
- 关系数据库的核心——关系数据模型及数据库的操作语言 SQL;
- 数据库的查询优化、并发控制、系统恢复、安全性等数据库管理技术;
- 数据库的应用设计方法和开发技术;
- 现代数据库技术的热点——数据挖掘、数据仓库、分布式数据库及空间数据库;

● 目前的主流数据库管理系统产品——Oracle、SQL Server、DB2、MySQL 及 Sybase 的概况。

教学中，可以根据教学对象和学时等具体情况对教材中的内容进行删减和组合，也可以进行适当扩展，参考学时为 32~64。为适应教学模式和教学方法的改革，本教材每章配套安排了习题及参考答案、实验指导书、多媒体电子课件及相应的网络教学资源，可以登录华信教育资源网 (<http://www.hxedu.com.cn>) 注册下载。

本教材第 1、2、3、4、5 章由魏祖宽编写，第 9、10、11、12 及 13 章的部分小节由郑莉华编写，第 13 章的主体部分、本教材的案例数据库、实验设计及指导的内容由牛新征编写，第 6、7、8、14 章由孙明编写。本教材由魏祖宽统稿并定稿。参加本教材编写的还有重庆邮电大学的刘兆宏，电子科技大学的胡旺、周益民、代林、胡红梅、陈佳、张乐信、刘小龙、姬海波、梁继东、陶晶晶、尹畅等，他们承担了收集基本素材、案例数据、相关技术资料及稿件校对等大量的基础工作。

本教材在编写过程中参考了大量新近出版的相关资料和书籍，吸取了许多专家的宝贵经验，在此向他们深表谢意！由于数据库应用技术发展迅速，作者学识有限，书中难免存在误漏之处，望广大读者批评指正。

# 目 录

第 1 章 数据库系统概论	1	2.2.1 关系	28
1.1 数据库的基本概念	1	2.2.2 关系的性质	30
1.1.1 信息与数据	1	2.2.3 关系模式	30
1.1.2 数据库	2	2.3 关系操作	32
1.1.3 数据库管理系统	2	2.4 关系的完整性	32
1.1.4 元数据	2	2.4.1 实体完整性	32
1.2 数据模型	2	2.4.2 参照完整性	33
1.2.1 组成要素	3	2.4.3 自定义完整性	34
1.2.2 概念数据模型	3	2.5 关系数据模型的优缺点	34
1.2.3 逻辑数据模型	4	2.6 关系代数	35
1.3 数据管理技术的产生和发展	7	2.6.1 基本运算	35
1.4 数据库管理系统的功能与特点	10	2.6.2 专门的关系运算	37
1.4.1 数据库管理系统的功能	10	2.6.3 扩展的关系运算	42
1.4.2 数据库管理系统的特点	11	2.7 关系演算	43
1.5 数据库系统的结构	13	2.7.1 元组关系演算	43
1.5.1 数据库三级模式结构	13	2.7.2 域关系演算语言 QBE	44
1.5.2 数据库二级映像与数据独立性	14	小结	47
1.5.3 数据库的体系结构	15	习题	48
1.5.4 数据库系统的组成	17	第 3 章 数据库设计过程与方法	50
1.6 数据库语言	18	3.1 数据库设计概述	50
1.6.1 数据定义语言	19	3.1.1 数据库的设计方法	51
1.6.2 数据操纵语言	19	3.1.2 数据库开发生命周期方法	53
1.6.3 数据控制语言	19	3.1.3 数据库设计的基本过程	55
1.6.4 事务控制语言	19	3.2 数据库需求分析	57
1.7 数据库技术的新发展	19	3.2.1 需求描述与分析	57
1.7.1 数据库技术发展趋势	19	3.2.2 需求分析的步骤	58
1.7.2 新型数据库与大数据	20	3.2.3 数据字典	62
小结	22	3.3 数据库概念设计	63
习题	23	3.3.1 概念设计的必要性	63
第 2 章 关系模型与关系代数	24	3.3.2 概念设计的方法和步骤	64
2.1 关系模型	24	3.3.3 概念设计工具: E-R 图	66
2.1.1 基本概念	24	3.4 数据库逻辑设计与优化	66
2.1.2 关系模型的数据结构	25	3.4.1 逻辑设计环境	66
2.1.3 数据操作	27	3.4.2 E-R 模式向关系模型的转换	66
2.1.4 数据约束	28	3.4.3 用关系规范化理论对关系模型进行优化	67
2.2 关系数据结构	28		

3.5	数据库的物理设计	68	5.3.1	模式分解问题	105
3.5.1	数据库的物理结构确定	68	5.3.2	无损分解	105
3.5.2	物理结构的评价	70	5.3.3	保持函数依赖的分解	108
3.6	数据库的実施和运行维护	70	5.3.4	模式分解与模式等价问题	110
3.6.1	数据库实际结构的建立	70	5.4	范式	111
3.6.2	装载数据	70	5.4.1	第一范式(1NF)	112
3.6.3	应用程序开发测试	71	5.4.2	第二范式(2NF)	112
3.6.4	数据库试运行	71	5.4.3	第三范式(3NF)	113
3.6.5	数据库运行与维护	71	5.4.4	BC范式(BCNF)	114
小结		72	5.5	多值函数依赖与4NF	115
习题		73	5.5.1	多值函数依赖	115
第4章	实体-联系模型	74	5.5.2	4NF	116
4.1	实体-联系模型概述	74	5.6	函数依赖与5NF	117
4.1.1	实体集	74	小结		117
4.1.2	属性	74	习题		118
4.1.3	联系集	76	第6章	基础SQL语言	121
4.2	约束	77	6.1	SQL概述	121
4.2.1	映射基数	77	6.1.1	SQL标准与历史	121
4.2.2	参与约束	77	6.1.2	SQL标准数据类型及SQL标准 内置函数	122
4.3	实体-联系图	77	6.1.3	SQL的功能与特点	123
4.4	扩展的实体-联系模型特性	78	6.2	数据库基本结构定义	124
4.5	实体-联系设计	80	6.2.1	数据库的创建、修改与删除	125
4.5.1	E-R图设计	80	6.2.2	基本表的定义、修改与删除	128
4.5.2	用实体集还是用属性	83	6.3	数据查询语句基本结构	130
4.5.3	用实体集还是用联系集	83	6.3.1	查询语句概述	130
4.5.4	二元还是 $n$ 元联系集	84	6.3.2	基本查询结构	131
4.6	数据建模的其他表示法及工具	84	6.3.3	查询语句中使用的运算符	133
4.6.1	统一建模语言	84	6.3.4	用聚集函数统计查询结果	137
4.6.2	数据建模的其他方法	90	6.3.5	INTO子句	139
小结		93	6.3.6	GROUP BY子句	140
习题		94	6.3.7	HAVING子句	140
第5章	规范化	96	6.3.8	ORDER BY子句	141
5.1	关系模式设计中的问题	96	6.3.9	用TOP谓词限制结果集	141
5.2	函数依赖	99	6.4	集合运算	143
5.2.1	函数依赖的定义	99	6.4.1	并运算	143
5.2.2	Armstrong公理	100	6.4.2	差/交运算	143
5.2.3	函数依赖与码的联系	101	6.4.3	集合运算的使用原则	144
5.2.4	属性集的闭包	102	6.5	连接查询	145
5.2.5	FD推理规则的完备性	103	6.5.1	连接概述	145
5.2.6	FD集的最小依赖集	103	6.5.2	内连接	145
5.3	模式分解	104			

6.5.3	自连接	146	7.3.2	函数的类型	182
6.5.4	外连接	147	7.3.3	函数的操作	182
6.6	嵌套查询	148	7.4	触发器	187
6.6.1	嵌套查询的基本概念	148	7.4.1	触发器的概念和作用	187
6.6.2	非相关子查询和相关子查询	149	7.4.2	触发器的分类	188
6.6.3	IN 和 NOT IN 运算符	150	7.4.3	触发器的工作原理	188
6.6.4	ANY 和 ALL 运算符	151	7.4.4	创建和使用 DML 触发器	189
6.6.5	EXISTS 和 NOT EXISTS 运算符	152	7.4.5	创建和使用 DDL 触发器	192
6.7	数据修改	153	7.5	递归查询	193
6.7.1	数据插入	153	7.5.1	递归查询基本概念	193
6.7.2	数据更新	155	7.5.2	递归查询的操作	193
6.7.3	数据删除	155	7.6	记录排序与分页	195
6.8	视图	156	习题		196
6.8.1	视图的定义	156	<b>第 8 章 数据库访问接口与应用程序开发</b>		197
6.8.2	查询中使用视图	157	8.1	数据库连接访问	197
6.8.3	视图的更新	157	8.1.1	ADO	198
6.8.4	视图的作用	158	8.1.2	ODBC	199
6.9	完整性约束	159	8.1.3	JDBC	201
6.9.1	完整性约束的定义	159	8.2	XML 数据交换	206
6.9.2	PRIMARY KEY 约束	159	8.2.1	XML 数据模型	207
6.9.3	UNIQUE 约束	159	8.2.2	XML 数据库	207
6.9.4	NOT NULL 约束	160	8.2.3	XML 数据交换	209
6.9.5	CHECK 约束	160	8.3	数据库应用程序开发	213
6.9.6	FOREIGN KEY 约束	161	8.3.1	数据库应用程序架构	213
6.9.7	域约束	163	8.3.2	数据访问层和对象关系映射 (ORM)	215
6.9.8	断言	164	8.3.3	数据展现: 查询和报表	216
6.9.9	完整性约束的修改	164	8.3.4	C/S 应用程序开发	217
6.9.10	完整性约束的验证	166	8.3.5	B/S 应用程序开发	217
习题		168	习题		218
<b>第 7 章 高级 SQL 语言</b>		171	<b>第 9 章 数据存储和查询处理与优化</b>		219
7.1	游标	171	9.1	数据存储	219
7.1.1	游标的基本概念	171	9.1.1	物理存储介质概述	219
7.1.2	游标种类	172	9.1.2	文件组织	221
7.1.3	游标操作	173	9.1.3	RAID	223
7.2	存储过程	176	9.2	索引	227
7.2.1	存储过程的概念	176	9.2.1	B+树索引	229
7.2.2	存储过程的类型	177	9.2.2	散列索引	233
7.2.3	存储过程的操作	178	9.2.3	位图索引	233
7.3	函数	181	9.3	查询处理及查询优化	235
7.3.1	SQL 中函数的概念	181	9.3.1	查询处理	235

9.3.2	关系代数运算的执行	236	第 12 章	数据库安全	284
9.3.3	查询优化技术	238	12.1	数据库安全基础	284
9.3.4	代数优化	241	12.2	数据库安全性控制	286
9.3.5	物理优化	243	12.3	用户标识和鉴别	287
习题		245	12.4	自主访问控制	288
第 10 章	事务与并发控制	246	12.4.1	权限类型	289
10.1	事务的概念与性质	246	12.4.2	授权及权限回收	290
10.2	事务的并发执行可能引起的问题	249	12.5	基于角色的访问控制	290
10.3	可串行化	251	12.6	强制访问控制	291
10.3.1	串行调度	251	12.7	安全审计	292
10.3.2	可串行化调度	252	12.8	其他数据库安全机制	294
10.3.3	可恢复性	255	12.8.1	使用视图实现安全控制	294
10.3.4	事务隔离性级别	256	12.8.2	使用存储过程实现安全控制	294
10.4	基于锁的并发控制协议	257	12.9	外部安全机制	295
10.4.1	封锁	257	12.9.1	数据加密	295
10.4.2	两段锁协议(2PL)	259	12.9.2	数字签名	297
10.4.3	锁的升级及更新锁	260	12.9.3	认证技术	297
10.5	活锁与死锁	261	习题		298
10.5.1	活锁	261	第 13 章	新型数据库和前沿技术	299
10.5.2	死锁	262	13.1	数据仓库和数据挖掘	299
10.6	多粒度封锁	264	13.1.1	数据仓库	299
习题		266	13.1.2	数据挖掘	302
第 11 章	故障恢复	267	13.2	分布式数据库	304
11.1	数据库恢复概述	267	13.2.1	分布式数据库系统概述	304
11.1.1	数据库恢复概述	267	13.2.2	分布式数据库系统的体系结构	306
11.1.2	故障种类	267	13.2.3	分布式网络数据的特性	310
11.1.3	日志记录	268	13.2.4	分布式查询处理	311
11.2	恢复与原子性	269	13.2.5	分布式事务管理	313
11.2.1	事务管理器	269	13.2.6	分布式数据库实例	317
11.2.2	使用日志撤销和重做事务	270	13.3	空间数据库	317
11.3	缓冲区管理	275	13.3.1	空间数据	317
11.3.1	缓冲区管理器结构	275	13.3.2	空间数据模型	318
11.3.2	缓冲区管理策略	275	13.3.3	空间对象关系	319
11.3.3	日志记录缓冲	276	13.3.4	空间数据操作	319
11.3.4	检查点	277	13.3.5	空间数据查询语言	321
11.4	恢复处理	280	13.3.6	空间索引	321
11.4.1	事务故障的恢复	280	13.4	多媒体数据库	322
11.4.2	系统故障的恢复	280	13.4.1	多媒体数据	323
11.4.3	介质故障的恢复	282	13.4.2	多媒体数据库的层次结构	323
习题		283	13.4.3	多媒体数据模式	324

13.4.4	多媒体元数据	324	14.4.3	Sybase 数据库特点	354
13.4.5	多媒体数据库查询	325	14.4.4	Sybase ASE 16 新特性	355
13.4.6	多媒体数据库	326	14.4.5	Sybase 数据库体系结构	355
13.4.7	多媒体数据库的检索	328	14.4.6	Sybase 数据库管理工具	355
13.5	NoSQL 数据库	328	14.5	MySQL 数据库	356
13.5.1	NoSQL 数据库概述	329	14.5.1	MySQL 数据库简介	356
13.5.2	NoSQL 数据库的架构及数据组织方式	329	14.5.2	MySQL 发展简史	356
13.5.3	NoSQL 数据库的特性	330	14.5.3	MySQL 特点	357
13.5.4	关系数据库和 NoSQL 数据库的区别	330	14.5.4	MySQL 5.7 新特性	357
13.5.5	几种常见的 NoSQL 数据库产品	331	14.5.5	MySQL 版本介绍	358
<b>第 14 章</b>	<b>商业数据库管理系统及选型</b>	<b>333</b>	14.5.6	MySQL 体系结构	358
14.1	Oracle 数据库	333	14.5.7	MySQL 开发工具	362
14.1.1	Oracle 数据库简介	333	14.6	工程应用中数据库管理系统的选型	362
14.1.2	Oracle 发展简史	333	<b>附录 A</b>	<b>数据库设计</b>	<b>367</b>
14.1.3	Oracle 特点	334		医院信息管理系统 (HIS-2018 版)	367
14.1.4	Oracle 12c 新特性	335		一、案例说明与实现目标	376
14.1.5	Oracle 12c 版本介绍	337		二、HIS 中涉及的缩写说明	376
14.1.6	Oracle 体系结构	337		三、HIS 中的实体间关系说明	376
14.1.7	Oracle Developer Suite	339		四、HIS 中涉及的各种实体、联系类型及实例数据	377
14.2	SQL Server 数据库	340	<b>附录 B</b>	<b>实验指导</b>	<b>385</b>
14.2.1	SQL Server 数据库简介	340		实验一 数据库 E-R 设计	385
14.2.2	SQL Server 版本介绍	340		一、实验内容及要求	385
14.2.3	SQL Server 特点	341		二、实验重点与难点	385
14.2.4	SQL Server 2016 新特性	343		三、上机实验作业	385
14.2.5	SQL Server 版本介绍	345		四、本实验参考资料	385
14.2.6	SQL Server 体系结构	345		五、项目需用仪器设备名称	385
14.2.7	SQL Server 开发工具	346		六、教学后记	385
14.3	DB2 数据库	347		实验二 数据库逻辑及物理设计	385
14.3.1	DB2 数据库简介	347		一、实验内容及要求	385
14.3.2	DB2 发展简史	347		二、实验重点与难点	386
14.3.3	DB2 版本介绍	348		三、上机实验作业	386
14.3.4	DB2 核心数据库的特点	349		四、本实验参考资料	386
14.3.5	DB2 V11.1 新特性	349		五、项目需用仪器设备名称	386
14.3.6	DB2 体系结构	351		六、教学后记	386
14.3.7	DB2 开发工具	352		实验三 数据库实现	386
14.4	Sybase 数据库	353		一、实验内容及要求	386
14.4.1	Sybase 数据库简介	353		二、实验重点与难点	386
14.4.2	Sybase 数据库发展简史	353		三、上机实验作业	386
				四、本实验参考资料	387

387	五、项目需用仪器设备名称	387
387	六、教学后记	387
388	实验四 数据库应用开发——服务器编程	387
387	一、实验内容及要求	387
387	二、实验重点与难点	387
387	三、上机实验作业	387
387	四、本实验参考资料	387
388	五、项目需用仪器设备名称	388
388	六、教学后记	388
388	实验五 数据库应用开发——访问接口编程	388
388	一、实验内容及要求	388

388	二、实验重点与难点	388
388	三、上机实验作业	388
388	四、本实验参考资料	388
388	五、项目需用仪器设备名称	388
388	六、教学后记	388
388	实验六 数据库备份与恢复	388
388	一、实验内容及要求	388
389	二、实验重点与难点	389
389	三、上机实验作业	389
389	四、本实验参考资料	389
389	五、项目需用仪器设备名称	389
389	六、教学后记	389

# 第1章 数据库系统概论

数据库是数据管理的最新技术，是计算机学科的重要分支。十余年来，数据库管理系统已从专业的应用程序包发展成为通用的系统软件。由于数据库系统具有数据结构化、最低冗余度、较高的程序与数据独立性等优点，所以较大的信息管理系统都是以数据库作为基础的。在这一章里，我们将学习数据库系统的基本概念和术语，了解数据管理技术的发展历史，明白数据库管理系统的功能与特点，并且要学习数据库系统的体系结构，包括三级模式、两级映像和当前使用的数据库结构，了解数据库的语言，理解数据库的组成，同时，了解当今数据库技术的发展趋势和大数据的基本概念，为后续的学习打下一个良好的基础。

## 学习目标：

- 掌握数据库的基本概念和相关术语
- 了解数据库技术的产生与发展
- 了解数据库管理技术发展的三个阶段
- 理解数据库中的各种数据模型
- 理解数据库系统的体系结构和一般组成
- 理解数据库系统的模式结构
- 了解数据库的各种语言
- 了解大数据和数据库发展趋势

## 1.1 数据库的基本概念

### 1.1.1 信息与数据

#### 1. 数据 (Data)

小杨：我这里有纸质的《哈利·波特与混血王子》，它算是数据吗？

老肖：不是，数据必须是保存在计算机中，能够被计算机识别、存储、处理的，如果你把这本书扫描成电子书，存储在计算机中，它就可以作为数据了。

小杨：那么在计算机的世界里，还有哪些可以看作数据呢？

老肖：目前数据不仅包括数字、字母、文字和其他特殊字符，而且还包括图形、图像、声音等多媒体数据。

小杨：原来数据有这么多种形式。能举个例子具体阐述数据吗？

老肖：可以，比如我，四川成都人，在电子科技大学工作，职称是教授，如果想把这条信息存储在计算机的数据库中，可以写成(肖老师，四川成都，电子科技大学，教授)，这样就把我的姓名，籍贯，工作单位，职称组织在了一起，形成一条有结构的数据记录。

#### 2. 信息 (Information)

小杨：听了上面您的例子，那么什么是信息呢？

老肖：信息是经过加工处理的数据，是对数据的具体描述。数据和信息既有联系又有区别。数据是信息的载体，而信息则是对数据的语义解释。同一数据也可能有不同的解释。

小杨：明白了，“肖老师，四川成都，电子科技大学，教授”是数据；“肖老师，四川成都人，在电子科技大学工作，职称是教授”便是信息。

老肖：真聪明，信息是反映客观现实世界的知识，用不同的数据形式可以表示同样的信息。例如，同样的新闻可以通过报纸、电台和电视来报道，它的表现形式不同，但其信息的内容可以相同。

### 1.1.2 数据库

小杨：我现在有好多书籍堆在家里，每次找书都十分不方便，应该怎么办呢？

老肖：不用担心。我们可以借用数据库的概念来管理大量的书籍。首先，把众多书籍归类：生活类、法律类、计算机类；其次，把它们分别放在书柜的不同隔间里，给隔间贴上类别标签，也记得留一定空间为以后的书籍归档使用；最后，把隔间中的书按照某种规则排列，如按照作者、国家等，以便家人查找。

小杨：好主意，那么书籍=数据，书柜=计算机，隔间=数据模型，家人=各种用户，按照规则排列=有组织的，留下多余空间=易扩展，对吗？

老肖：嗯嗯，数据库（Database，简称 DB）是长期存储在计算机内的、有组织的、可共享的大量数据的集合。数据库中的数据按照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，并为各种用户共享。

### 1.1.3 数据库管理系统

小杨：那么归类之后，我该怎么管理这些书籍呢，如果别人借走了一本书我却忘了，怎么办？如果书籍买重复了，岂不是花冤枉钱？唉，好烦！

老肖：没事的，同样可以借用数据库管理系统的理念。数据库管理系统是位于用户与操作系统之间的一层数据管理软件，其主要目标是使数据成为方便各种用户使用的资源，并提高数据的安全性、完整性和可用性。通过授权、存取控制、用户验证等手段达到安全性，利用实体完整性、参照完整性和自定义完整性达到完整性，利用查询更新等操作达到可用性。

小杨：那么这和我管理书籍有什么联系呢？

老肖：你可以建立一套机制，规定只有父母可以拿取所有的书，这等于授权，但是不允许他们在书上记笔记，只允许读，不允许写，这等于存取控制；每次购买新书时，先查阅书柜中相同种类的书籍，如果发现存在要买的书，则停止购买，这就等于实体完整性，不允许重复书籍出现，并且规定每本书的购买价格不得超过 100 元，这就是自定义完整性。

### 1.1.4 元数据

老肖：元数据即描述数据的数据，相当于数据字典。主要是描述数据属性的信息，如数据的类型、格式、存储大小等。拿书柜举例，书柜相当于数据库，元数据记录了数据库的版本、创建日期等。每个隔间相当于二维表，元数据记录了表的创建时间、创建者、创建语句等。

## 1.2 数据模型

数据模型是对现实世界数据特征的抽象。例如，汽车模型是对现实世界中汽车的一种模拟和抽象，它抽象了汽车的基本特征——车身、轮胎。但是由于计算机不能直接处理现实世界中的具体事物，所

以人们必须事先把具体的事物转换为计算机能够处理的数据，这种现象称为数字化。数字化是把现实世界中具体的人、物用数据模型来抽象、表示和处理。

数据模型就是对现实世界的模拟。现有的数据库系统均建立在数据模型的基础上，因此数据模型是数据库系统的核心和基础，现有的数据模型分为概念数据模型和逻辑数据模型。

### 1.2.1 组成要素

小杨：数据模型的组成要素之一是数据结构，那么，什么是数据结构呢？

老肖：数据结构用于描述数据库的组成对象以及对象之间的联系。它分为两类：一类是与对象的类型、内容、性质有关；另一类是与数据之间联系有关。例如，关系模型中的域、属性、关系等。它们描述了对对象的类型、内容、性质。当然，对象之间也是有联系的，能说出来吗？

小杨：一对一、一对多、多对多算是联系吗？

老肖：当然是了，它们描述了对对象之间的联系。在数据库中，按照数据结构可以划分为层次结构、网状结构和关系结构，分别命名为层次模型、网状模型和关系模型。总之，数据结构是对系统静态特征的描述。

小杨：数据模型的组成要素之二是数据操作，可以简单理解为对数据的操作吗？

老肖：嗯，数据操作就是对数据库中的各种对象的实例（或取值）执行允许的操作。它包括查询和更新（插入、删除、修改）两大类操作。数据操作是对系统动态特征的描述。

小杨：那么那些软件工程师说的 DAO 和 CRUD 和这些操作是否有关系？

老肖：DAO (Data Access Object) 是一个数据访问接口，位于业务逻辑和数据库之间，应用程序可以通过 DAO API 来访问数据库。对数据库的基本操作包括增加 (Create)、读取 (Retrieve, 重新得到数据)、更新 (Update) 和删除 (Delete)；采用这几个单词的首字母简写成 CRUD。通常对数据库的操作就是 CRUD。

小杨：数据模型的组成要素之三是数据完整性，它是为了保证数据的完整吗？

老肖：正是这样。数据完整性约束是关于数据状态和状态变化的一组完整性约束条件（规则）的集合。它保证了数据库的正确性、有效性和相容性。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，此外，数据模型还提供自定义完整性的约束机制。例如，每个月个人所得超过一定的金额必须交税，每个人必须有身份证号码等。

### 1.2.2 概念数据模型

小杨：老师让我设计一款人事管理系统模型，但我还不知道使用 SQL Server 还是 Oracle，担心即使模型设计出来了，用户也看不懂这个系统，这该如何是好？

老肖：概念数据模型就是用来解决上述问题的。概念数据模型 (Conceptual Data Model) 是用户容易理解的、对现实世界特征的数据抽象，它与具体的数据库管理系统 (Database Management System, DBMS) 无关，是数据库设计员与用户之间进行交流的语言。它具有强大的表达能力，能够方便、直接地表达应用中的各种语义知识，同时简单清晰，易于用户理解。常用的概念数据模型是实体-联系 (Entity-Relationship) 模型。

小杨：怪不得工程师们常说数据库设计的第一步是做 E-R 模型图，那么应该怎么画呢？

老肖：先别急着画图，听我介绍 E-R 模型中的一些概念。

- 实体：客观存在并可相互区别的事物。实体可以是具体的对象，如一名学生、一辆汽车，也可以是抽象的事件，如一次选课、一次驾车等。
- 属性：实体具有的若干特征。例如，每个学生有学号、姓名、年龄、宿舍号等属性。

- 实体集：性质相同的同类实体的集合。例如，计算机学院的所有学生就是一个实体集。
- 键（码）：唯一标识实体集中每个实体的属性集合。例如，一个学生可以以身份证、学号等具有唯一性的属性作为键，选择其中之一作为主键。
- 域：属性的取值范围，例如，学号必须是 201421012345 这样的 12 位数，性别的域为 {男, 女}。
- 联系：反映事物之间的联系。实体内部的联系为各属性之间的联系；实体之间的联系通常是不同实体集之间的联系。

两个实体集之间的联系可以分为 3 类。

- 一对一联系 (1:1)。

如果对于实体集  $A$  中的每个实体，实体集  $B$  中至多有一个实体与之联系，反之亦然，则称实体集  $A$  与实体集  $B$  具有一对一联系。例如，火车上的座位与乘客之间的联系。

- 一对多联系 (1:n)。

如果对于实体集  $A$  中的每个实体，实体集  $B$  中有  $n$  个实体 ( $n \geq 2$ ) 与之联系，反之，对于实体集  $B$  中的每个实体，实体集  $A$  至多有一个实体与之联系，则称实体集  $A$  与实体集  $B$  具有一对多联系。例如，学校的学院与学生之间的联系。

- 多对多联系 ( $m:n$ )。

如果对于实体集  $A$  中的每个实体，实体集  $B$  中有  $n$  个实体 ( $n \geq 2$ ) 与之联系，反之，对于实体集  $B$  中的每个实体，实体集  $A$  也有  $m$  个实体与之联系，则称实体集  $A$  与实体集  $B$  具有多对多联系。例如，学生与课程之间的联系。

小杨：好多概念啊，不过都还是挺容易理解的。看 E-R 图中有矩形、椭圆形和菱形，它们分别代表什么意思呢？

老肖：实体集用矩形，属性用椭圆形，联系用菱形，并要注明联系的类型 (1:1、1:m、m:n)。下面来展示一张 E-R 图供参考，如图 1.1 所示。

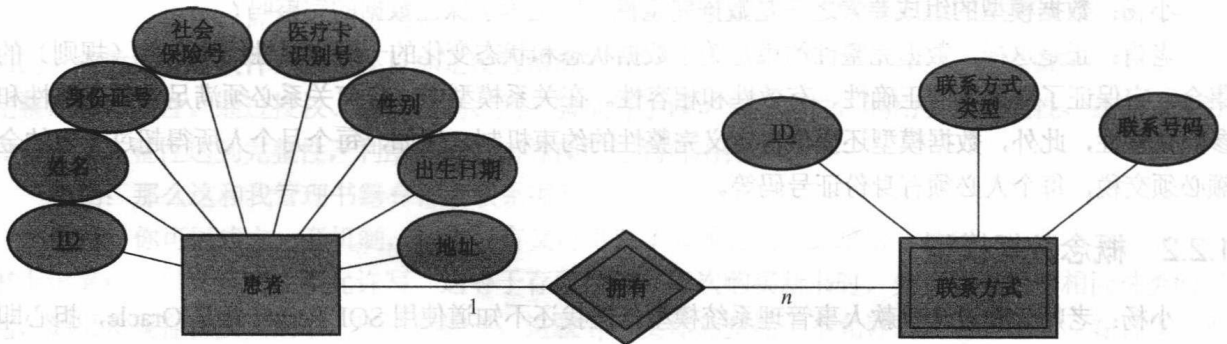


图 1.1 E-R 图

### 1.2.3 逻辑数据模型

老肖：逻辑数据模型即数据模型，是用户从数据库中所看到的数据模型，是具体的 DBMS 所支持的数据模型。目前，数据库领域中常用的数据模型是关系模型和面向对象模型。层次模型、网状模型已渐渐被淘汰。

#### 1. 层次模型

小杨：听说层次模型是数据库系统中最先出现的数据模型，曾经得到广泛的使用。

老肖：是的。层次数据库系统的典型代表是 IBM 公司的 IMS (Information Management System) 数据库管理系统。它是 IBM 公司在 1968 年推出的第一个大型商用数据库管理系统。

小杨：我看层次模型和数据结构课程中的树形结构很像嘛。

老肖：没错！层次模型用树形（层次）结构来表示各类实体以及实体间的联系，每个节点表示一个记录类型（实体型），每个记录类型包含若干个字段（实体的属性）。现实世界中许多实体之间的联系本来就呈现出一种很自然的层次关系，如行政机构、家族关系等。医院实体层次模型如图 1.2 所示。

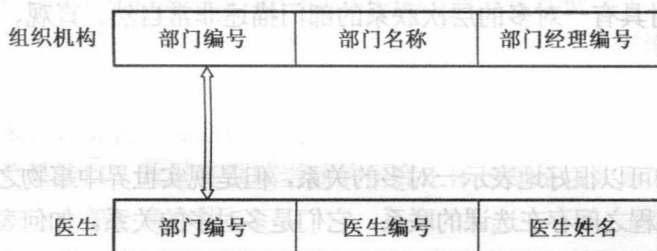


图 1.2 医院实体层次模型

小杨：这个图看起来数据结构比较简单清晰。

老肖：嗯，这是层次模型的一个优点，此外，因为记录之间的联系用有向边表示，当存取某个节点的记录值时，DBMS 就沿着存取路径很快找到该记录值，因此，层次数据库的查询效率高。同时，它也提供了良好的完整性支持，例如，指定了实验室子节点，必定存在学院父节点，父子之间必有联系。

小杨：有这么好的特点，为什么层次模型在 20 世纪 70 年代末和 80 年代初迅速失去了市场呢？

老肖：层次模型有以下几个缺点。

#### ● 实现复杂。

尽管层次模型的 DBMS 减轻了 DBMS 设计者和程序员对数据依赖问题的负担，但它们仍然必须对数据的物理存储特性有非常深入的了解。因此，数据库设计的实现仍然非常复杂。

#### ● 难于管理。

对插入和删除操作的限制比较多。数据库结构的任何修改，如节点的重新定位，都要求所有访问这个数据库的应用程序也做相应的修改。因此，数据库的管理可能会变得非常琐碎和麻烦。

#### ● 实现的限制。

许多一般的联系并不遵守层次模型所要求的 1:m 标准。比如，大学里面每门课程可能包括许多学生，并且每个学生也可以选修许多门课程。这样一个多对多 (m:n) 的联系很难在层次模型中实现。此外，很多现实世界中的联系是基于一个子节点对应多个父节点的。比如，在一个订单系统中，一个订单行有两个父节点：订单和部件。

#### ● 缺乏标准。

层次模型没有一组精确的标准概念，也没有遵守一个特定标准模型的实现。实现的问题尤其麻烦，因为层次数据库管理组件没有一个标准的数据定义语言 (DDL) 来定义数据库的各个组成部分，也没有一个数据操作语言 (DML) 来操作数据库内容，从一个层次模型的 DBMS 转移到另一个层次模型的 DBMS 非常困难，移植性受到限制。

#### ● 缺乏结构独立性。

结构独立性是指当修改数据库结构时，DBMS 访问数据的能力不受影响。层次数据库也称为导航系统，数据访问要求使用物理存储路径来“导航”以获得正确的节点，查询子女节点必须通过双亲节点。对数据库结构的修改可能会导致一些之前可以正常运行的应用程序出现问题，这种结构依赖限制了数据独立性所带来的好处。

### ● 应用程序编写和使用复杂性。

给定了一个导航的数据库系统结构，应用程序员和最终用户为了存取数据还必须准确地知道数据在数据库内部的物理分布。即使他们知道数据的存取路径，要得到这个数据还要对整个复杂指针系统有所了解。

可见，用层次模型对具有一对多的层次联系的部门描述非常自然、直观、容易理解。这是层次数据库的突出优点。

## 2. 网状模型

小杨：虽然层次结构可以很好地表示一对多的关系，但是现实世界中事物之间的联系更多的是非层次关系，比如，学生和课程之间存在选课的联系，它们是多对多的关系。如何表示这种非树形结构呢？

老肖：不用担心，网状结构正是为了克服这一弊病而诞生的。网状模型是一种比层次模型更具普遍性的结构。它去掉了层次模型的两个限制，允许多个节点没有双亲节点，允许节点有多个双亲节点。网状模型可以更直接地去描述现实世界，层次模型实际上是网状模型的一个特例。让我们来举个例子解释这些概念，如图 1.3 所示。

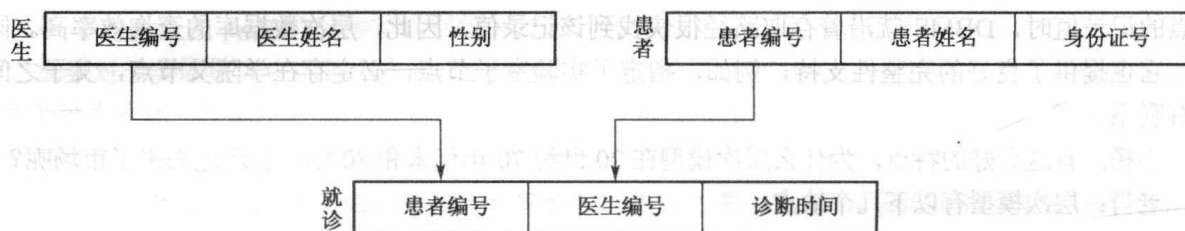


图 1.3 就诊网状模型

医生、患者、就诊三个节点，但是就诊的父节点是医生、患者（允许节点有多个双亲节点），患者、医生没有双亲节点（允许多个节点没有双亲节点）。

小杨：那么，网状模型的优点有哪些呢？

老肖：网状模型保留了许多层次模型的优点，同时，它也对层次模型进行了改进。

### ● 能够更加方便地描述现实世界，节点之间可以有多种联系。

### ● 具有良好的性能，存取效率高。

小杨：如果我想扩展医生、患者、就诊这三个节点，再添加处方节点，处方和医生、患者、就诊都有联系。以此下去，网状模型不是乱得和蜘蛛网似的吗？

老肖：没错。这是网状模型的缺点之一：**结构复杂性**，随着应用环境的扩大，数据库的结构变得越来越复杂。

### ● 系统复杂性。

由于系统复杂性，网状模型用来管理联系时，数据库完整性控制及其效率有时会降低甚至失效。为访问数据库，数据库管理员、程序员和最终用户必须非常熟悉它的内部结构。因此，网状数据库并没有设计成一个对用户友好的系统。

### ● 缺乏结构独立性。

如果网状数据库结构做了改变，则所有的应用程序必须重新授权。简单地讲，网状模型具有数据独立性，但不具备结构独立性。

### ● 用户不容易掌握和使用。

网状模型没有设计成用户容易掌握和使用的系统，是一个高技能的系统。

由于记录之间联系是通过存取路径实现的，应用程序在访问数据时必须选择适当的存取路径。因