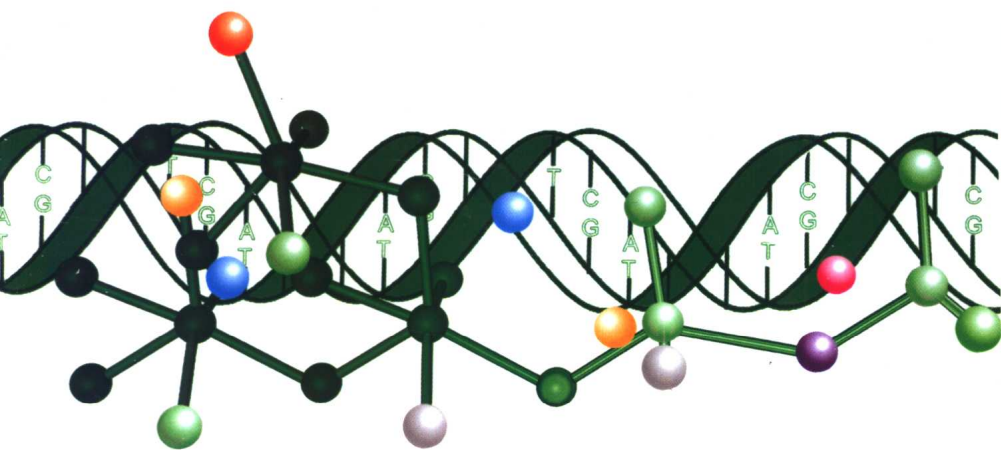


SQL Server 2000

实用教程



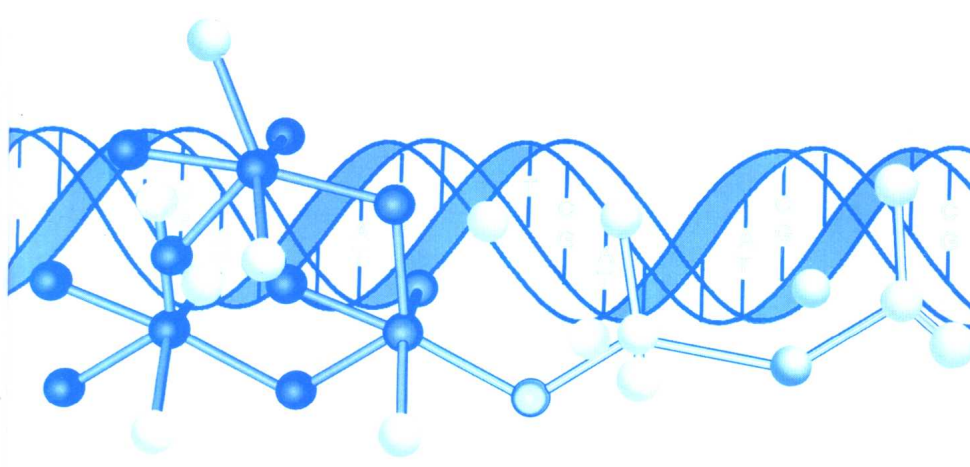
北京希望电子出版社 总策划
刘卫宏 编 著



科学出版社
www.sciencep.com

SQL Server 2000

实用教程



北京希望电子出版社 总策划
刘卫宏 编 著

 科学出版社
www.sciencep.com

内 容 简 介

这是一本讲解 MS-SQL Server 2000 应用技能的实用教材。本书知识全面,从 SQL Server 2000 的安装、调试、建立设备、建立数据库、建立表,到数据库的优化与调整、操作数据库、保证数据库的完整性和一致性,数据库的备份,以及建立各种应用的范例都作了详尽的介绍与剖析。

全书共分为 18 章,内容涉及数据库系统,SQL Server 2000 数据库构架,安装及配置,SQL 语言,T-SQL 程序设计,事务管理,数据库系统管理,数据库安全与维护,数据库备份及还原,性能调整与优化,利用 ODBC、ADO、DAO 进行 C/S 模式数据库应用程序开发,在 Delphi 7 IDE 下进行基于 SQL Server 2000 分布式数据库应用开发,以及利用 ASP 进行基于 Internet 的 Web 应用系统开发等知识。内容全面,从简单到复杂,讲解透彻,实例实用。

本书主要面向数据库系统管理员、高校学生,以及数据库应用开发人员。也可作为业余爱好者、社会培训班的理想教材。

需要本书或技术支持的读者,请与北京中关村 083 信箱(邮编 100080)发行部联系,电话:010-62528991,62524940,62521921,62521724,82610344,82675588(总机) 传真:010-62520573 E-mail: yanmc@bhp.com.cn。

图书在版编目(CIP)数据

SQL Server 2000 实用教程 / 刘卫宏编著. —北京:
科学出版社, 2003.9
ISBN 7-03-012146-5
I. S... II. 刘... III. 关系数据库—数据库管理系统,
SQL Server 2000—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2003)第 077735 号

责任编辑: 但明天 / 责任校对: 赵文博
责任印刷: 媛明 / 封面设计: 梁运丽

科学出版社 出版

北京东黄城根北街 16 号
邮政编码: 100717

<http://www.sciencep.com>

北京市媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2003 年 9 月第一版 开本: 787×1092 1/16
2003 年 9 月第一次印刷 印张: 28
印数: 1—5 000 字数: 649 000

定价: 39.00 元

前 言

数据库是计算机科学技术中发展最快，应用最广泛的重要分支之一，它已成为计算机信息系统、应用系统、科学数据库及其信息系统建设的重要技术基础和支柱。

数据库技术及其应用系统最早出现于 60 年代。70 年代初，E.F.Codd 提出数据库的关系模型，并对关系数据理论做出了创造性的贡献，从而使数据库从技术上升到理论。关系数据库理论和技术在 70~80 年代得到长足的发展和广泛而有效的应用；这些都是发生在上个世纪的事情。

在这个举足轻重的应用领域中，Microsoft SQL Server 以其低价位、多功能、良好的用户界面，以及与 Windows 系统的全面集成而后来居上，不仅在中小型数据库领域占有绝对领先的份额，而且随着 SQL Server 2000 的推出，向原来属于 Oracle 等大型数据库系统的领地不断扩展，市场占有率不断上升。

本书主要面向数据库系统管理员、高校学生，以及数据库应用开发人员。本书知识全面，从 SQL Server 2000 的安装、调试、建立设备、建立数据库、建立表，到数据库的优化与调整、操作数据库、保证数据库的完整性和一致性，数据库的备份，以及建立各种应用的范例都作了详尽的介绍与剖析。

本书共分 18 章，按照基础知识、系统安装和配置、各种对象管理、安全、备份和恢复、性能优化，以及开发各种类型数据库应用程序的顺序进行讲解，内容从简单到复杂，非常有利于具备一定数据库基本理论的高校学生学习 SQL Server，并能使用它较快地进行数据库应用程序的编程。一般书籍都是把编程内容放在后面，而本书尽可能把编程任务提前，使得更加有利于具备一定数据库知识的人学习和掌握。

本书由刘卫宏、李红玲组织编写，万方工作室的全体同仁参加了本书的校正、审查、输入等工作。其他参加本书部分编写、录排、校对工作的人员还有：于自跃、李小将、陈曙辉、邓波、田丽韞、龚超等。其间龚超同志对全稿进行了严格细致的复审。本书的出版是集体劳动的结晶，在此对万方工作室的全体工作人员深表感谢。

由于时间仓促，且经验和水平有限，文中难免有不妥之处，在此殷切期望广大读者能提出中肯、尖刻的意见，以便于提高水平，把更好的图书呈现给大家！

作者

2003 年 5 月于北京

目 录

第 1 章 数据库系统基础.....	1	小结.....	41
1.1 数据库的发展.....	1	习题.....	41
1.2 数据库模型.....	4	第 3 章 安装和配置.....	42
1.2.1 层次型数据库.....	4	3.1 软件、硬件要求.....	42
1.2.2 网络型数据库.....	5	3.1.1 软件要求.....	42
1.2.3 关系型数据库.....	6	3.1.2 硬件要求.....	43
1.3 关于 RDMS.....	8	3.2 SQL Server 2000 数据库文件.....	44
1.3.1 关系模型.....	8	3.3 安装 SQL Server 2000.....	44
1.3.2 关 联.....	11	3.3.1 本地安装.....	45
1.3.3 主键和外键.....	13	3.3.2 远程安装.....	52
1.3.4 数据完整性.....	15	3.3.3 升级安装.....	53
1.3.5 约 束.....	15	3.4 建立帐号.....	55
1.3.6 E. F. Codd 法则.....	16	3.4.1 分配 SQL Server 服务帐号.....	55
1.4 数据库体系结构.....	18	3.4.2 检测安装.....	56
1.4.1 外模式.....	19	3.4.3 启动 SQL Server 服务.....	56
1.4.2 概念模式.....	20	3.4.4 隐藏 SQL Server.....	57
1.4.3 内模式.....	20	3.4.5 解决安装问题.....	57
1.4.4 映 像.....	21	小结.....	57
1.5 E-R 方法.....	21	习题.....	58
1.6 数据库设计方法与步骤.....	24	第 4 章 SQL 语言基础.....	59
1.6.1 数据库设计方法.....	25	4.1 SQL 历史.....	59
1.6.2 数据库设计步骤.....	27	4.1.1 SQL 语言分类.....	60
小结.....	27	4.1.2 SQL 语句的基本语法.....	62
习题.....	27	4.1.3 对象命名约定.....	62
第 2 章 SQL Server 2000 概述.....	28	4.1.4 别 名.....	62
2.1 SQL Server 进化过程.....	28	4.2 Select 语句.....	63
2.2 SQL Server 2000 版本.....	29	4.2.1 From 子句.....	65
2.2.1 SQL Server 2000 的特性.....	30	4.2.2 Where 子句.....	67
2.2.2 SQL Server 2000 所增特性.....	31	4.2.3 GROUP BY 子句.....	72
2.3 SQL Server 构架.....	34	4.2.4 ORDER BY 子句.....	77
2.3.1 关系数据库引擎.....	35	4.2.5 COMPUTE 和 COMPUTE BY	
2.3.2 分析服务.....	35	子句.....	78
2.3.3 其他组件.....	36	4.2.6 使用 UNION 子句.....	80
2.4 SQL Server 2000 常用工具.....	36	4.3 汇总函数.....	81

4.3.1	SUM 函数	81
4.3.2	AVG 函数	82
4.3.3	MAX 函数	82
4.3.4	MIN 函数	83
4.3.5	COUNT 函数	83
4.4	联接查询	84
4.4.1	联接的类型	85
4.4.2	在 From 和 Where 分句中指定联接	87
4.5	嵌套查询	87
4.5.1	使用 IN 和 NOT IN 关键字	88
4.5.2	使用比较运算符	89
4.5.3	使用 EXISTS 和 NOT EXISTS 关键字	89
4.6	数据更改	90
4.6.1	Insert	90
4.6.2	Update	91
4.6.3	Delete	91
	小结	92
	习题	92
第 5 章	T-SQL 程序设计基础	93
5.1	SQL Server 数据类型	93
5.1.1	字符	94
5.1.2	数值	94
5.1.3	货币	96
5.1.4	时间/日期	97
5.1.5	Unicode	98
5.1.6	二进制	100
5.1.7	sql_variant	100
5.1.8	table 类型	102
5.1.9	自定义数据类型	102
5.2	批处理	103
5.3	注释	104
5.4	SQL Server 变量	105
5.4.1	全局变量	105
5.4.2	局部变量	107
5.5	SQL Server 函数	108
5.5.1	系统函数	109
5.5.2	日期函数	110

5.5.3	字符串函数	111
5.5.4	数学函数	113
5.5.5	用户自定义函数	114
5.6	流程控制	119
5.6.1	BEGIN...END 语句块	119
5.6.2	IF...ELSE 语句	120
5.6.3	WHILE 语句	123
5.6.4	CASE 表达式	124
5.6.5	RETURN 语句	124
5.6.6	WAITFOR 语句	125
5.6.7	GOTO 语句	125
5.6.8	PRINT 语句	126
5.7	游标	126
5.7.1	DECLARE	127
5.7.2	OPEN	128
5.7.3	FETCH	128
5.7.4	CLOSE	131
5.7.5	DEALLOCATE	131
	小结	132
	习题	132
第 6 章	管理事务	133
6.1	概述	133
6.1.1	显式事务	134
6.1.2	隐式事务	136
6.1.3	事务工作机制	137
6.2	检查点处理	138
6.3	分布式事务	139
6.4	锁	140
6.4.1	锁类别	140
6.4.2	控制锁	141
6.4.3	锁升级	142
6.4.4	LOCKS 选项	142
6.4.5	观察锁	142
6.4.6	死锁	142
6.4.7	事务独立层次	144
6.5	编写有效的事务	145
	小结	146
	习题	146
第 7 章	管理数据库	147

7.1 数据库的结构.....	147	9.2.2 修改视图.....	212
7.1.1 文件.....	147	9.2.3 重命名视图.....	213
7.1.2 文件组.....	149	9.2.4 删除视图.....	213
7.2 系统数据库.....	150	9.3 使用 Transact SQL 语言管理 视图.....	214
7.3 使用企业管理器管理数据库.....	150	9.3.1 创建视图.....	214
7.3.1 创建用户数据库.....	150	9.3.2 修改视图.....	219
7.3.2 设置数据库选项.....	154	9.3.3 删除视图.....	220
7.3.3 修改数据库.....	156	9.4 使用视图操作表数据.....	220
7.3.4 删除数据库.....	157	9.4.1 使用视图检索数据.....	220
7.4 使用 T-SQL 管理数据库.....	158	9.4.2 通过视图添加表数据.....	221
7.4.1 创建数据库.....	158	9.4.3 更新视图中的数据.....	223
7.4.2 设置数据库选项.....	161	9.4.4 删除视图中的数据.....	223
7.4.3 修改数据库.....	165	小结.....	223
7.4.4 删除数据库.....	170	习题.....	224
小结.....	171	第 10 章 管理索引	225
练习.....	171	10.1 索引基本概念.....	225
第 8 章 管理数据库表	172	10.1.1 聚集索引和非聚集索引.....	226
8.1 使用企业管理器创建和管理表.....	172	10.1.2 惟一索引和组合索引.....	226
8.1.1 创建表.....	172	10.2 使用企业管理器管理索引.....	228
8.1.2 约束.....	175	10.2.1 系统自动创建索引.....	228
8.1.3 规则.....	183	10.2.2 使用向导创建索引.....	230
8.1.4 默认值.....	185	10.2.3 创建索引.....	232
8.1.5 修改表.....	187	10.2.4 删除索引.....	234
8.1.6 删除表.....	189	10.3 使用 T-SQL 管理索引.....	235
8.2 使用 SQL 语句创建和管理表.....	190	10.3.1 创建索引.....	235
8.2.1 创建表.....	190	10.3.2 删除索引.....	237
8.2.2 约束.....	192	10.4 索引选择.....	238
8.2.3 规则.....	196	10.4.1 集群式索引选择.....	238
8.2.4 默认值.....	197	10.4.2 非集群式索引选择.....	239
8.2.5 修改表.....	198	10.5 索引维护.....	239
8.2.6 删除表.....	201	10.5.1 更新统计.....	239
小结.....	202	10.5.2 DBCC UpdateUSAGE.....	240
习题.....	202	10.5.3 DBCC SHOWCONTIG.....	240
第 9 章 管理视图	204	10.5.4 DBCC DBREIndex.....	240
9.1 视图基本概念.....	204	小结.....	241
9.1.1 视图的优点.....	204	习题.....	241
9.1.2 视图分类.....	205	第 11 章 管理触发器	242
9.2 使用企业管理器管理视图.....	205	11.1 触发器基本概念.....	242
9.2.1 创建视图.....	205		

11.1.1 触发器作用.....	242	第 13 章 数据库安全	277
11.1.2 触发器类型和触发操作.....	243	13.1 SQL Server 2000 安全模式.....	277
11.1.3 使用触发器的优点.....	243	13.1.1 Windows 身份验证.....	278
11.1.4 创建触发器的规则和限制.....	244	13.1.2 SQL Server 身份验证.....	278
11.2 使用企业管理器管理触发器.....	244	13.1.3 Windows 和 SQL Server 身份验证的比较.....	279
11.2.1 创建触发器.....	244	13.1.4 选择身份验证模式.....	279
11.2.2 修改触发器.....	246	13.2 创建和管理用户登录.....	281
11.2.3 删除触发器.....	246	13.2.1 使用企业管理器创建和管理登录账户.....	281
11.3 使用 T-SQL 管理触发器.....	247	13.2.2 数据库角色.....	283
11.3.1 创建触发器.....	247	13.2.3 创建数据库角色.....	287
11.3.2 查看触发器相关数据.....	249	13.3 权限.....	289
11.3.3 修改触发器.....	251	13.3.1 权限的分类.....	289
11.3.4 删除触发器.....	252	13.3.2 权限管理.....	290
11.4 确保数据完整性.....	252	13.3.3 管理语句权限.....	293
11.4.1 封装事务规则.....	254	13.4 安全策略.....	294
11.4.2 确保引用完整性.....	255	13.4.1 设计安全策略的步骤.....	294
小结.....	256	13.4.2 安全检查列表.....	299
习题.....	256	小结.....	300
第 12 章 管理存储过程	257	习题.....	301
12.1 存储过程基本概念.....	257	第 14 章 数据库备份与恢复	302
12.1.1 存储过程运行过程.....	257	14.1 备份和恢复基本概念.....	302
12.1.2 存储过程的优点.....	258	14.1.1 备份设备.....	303
12.1.3 创建存储过程的规则.....	259	14.1.2 备份策略.....	305
12.2 使用企业管理器管理存储过程.....	259	14.1.3 数据库恢复.....	306
12.2.1 创建存储过程.....	259	14.2 使用企业管理器管理备份和恢复.....	307
12.2.2 查看和修改存储过程.....	262	14.2.1 使用备份向导进行数据库备份.....	308
12.2.3 删除存储过程.....	264	14.2.2 使用企业管理器进行数据库备份.....	309
12.3 使用 T-SQL 管理存储过程.....	264	14.2.3 使用企业管理器恢复数据库.....	310
12.3.1 创建存储过程.....	264	14.3 使用 T-SQL 管理备份和恢复.....	312
12.3.2 执行存储过程.....	266	14.3.1 使用 T-SQL 语句进行数据库备份.....	312
12.3.3 查看存储过程.....	269	14.3.2 使用 T-SQL 语句恢复数据库.....	313
12.3.4 修改存储过程.....	270	14.4 恢复 master 数据库.....	315
12.3.5 删除存储过程.....	272	14.4.1 从好的备份中恢复.....	316
12.3.6 创建一组过程.....	272		
12.3.7 在存储过程中使用参数.....	273		
12.3.8 WITH RECOMPILE 选项.....	275		
小结.....	276		
习题.....	276		

14.4.2 无备份情况下恢复 master	318	15.8.3 查看数据库的数据空间使用 信息.....	345
小结	319	15.8.4 查看 SQL Server 活动及其常规 统计信息.....	345
习题	319	小结.....	346
第 15 章 数据库性能调整与优化	320	习题.....	346
15.1 性能调整原则	320	第 16 章 数据库应用程序开发基础	347
15.2 在配置中检查最大异步 IO 选项.....	321	16.1 项目管理.....	347
15.3 消耗 CPU 和磁盘 I/O 资源的 组件.....	321	16.1.1 软件开发生命周期模型	347
15.3.1 工作线程.....	321	16.1.2 项目设计文档.....	349
15.3.2 懒散记录器.....	322	16.1.3 数据库设计.....	350
15.3.3 检查点.....	322	16.2 数据库应用体系结构.....	353
15.3.4 日志管理器.....	323	16.2.1 C/S 模式与 B/S 模式的比较 分析.....	353
15.3.5 预读管理器.....	324	16.2.2 C/S 与 B/S 模式相结合方案	357
15.4 磁盘 I/O 性能.....	324	16.3 编程时应考虑性能优化	368
15.4.1 广播磁盘传输率和 SQL Server	325	16.4 数据库接口技术简介	359
15.4.2 序列化和非序列化磁盘 I/O 操作.....	325	16.4.1 DAO.....	360
15.4.3 磁盘 I/O 传输率和 PCI 总线 带宽.....	326	16.4.2 ADO.....	360
15.4.4 RAID.....	327	16.4.3 ODBC.....	362
15.4.5 磁盘 I/O 并行性	330	小结.....	367
15.5 SQL Server 索引.....	332	习题.....	367
15.5.1 聚集索引.....	332	第 17 章 数据库应用程序开发实践	368
15.5.2 非聚集索引.....	333	17.1 Delphi 集成开发环境简介.....	369
15.5.3 覆盖索引.....	333	17.1.1 主窗口 (Main Window)	368
15.5.4 自动覆盖索引和覆盖查询.....	333	17.1.2 窗体设计器.....	369
15.5.5 索引选择.....	333	17.1.3 对象观察器.....	370
15.5.6 聚集索引的选择.....	334	17.1.4 代码编辑器和代码浏览器	370
15.5.7 FILLFACTOR 和 PAD-Index	336	17.2 Delphi 数据库应用体系结构... ..	371
15.6 SQL Server 性能调整工具	337	17.2.1 传统数据库应用程序体系 结构.....	372
15.6.1 SQL Server 剖析器.....	337	17.2.2 Delphi 提供的 SQL Server 访问机制.....	374
15.6.2 SQL Server 查询分析器.....	339	17.3 Delphi 工具和控件	379
15.6.3 性能监视器.....	340	17.3.1 工 具.....	379
15.7 其他性能问题.....	342	17.3.2 常用数据控件.....	380
15.8 使用 T-SQL 监视 SQL Server	344	17.4 实 例	383
15.8.1 查看当前的锁.....	344	17.4.1 建立与数据库的连接	383
15.8.2 查看当前服务器活动	345	17.4.2 数据库设计.....	385

17.4.3 应用程序设计.....	388	18.4.3 操作存储过程.....	412
小结.....	399	18.5 ASP 制作留言系统.....	418
第 18 章 使用 ASP 建立信息化环境	400	18.5.1 数据库表的结构.....	418
18.1 ASP 环境安装与配置.....	400	18.5.2 实现代码.....	419
18.1.1 ASP 环境安装.....	400	小结.....	422
18.1.2 配置 ASP 运行环境.....	401	附录 A Master 系统表	423
18.2 ASP 内建对象.....	402	A.1 syslogins (sysxlogins).....	423
18.2.1 Request 对象.....	403	A.2 sysremotelogins.....	424
18.2.2 Response 对象.....	404	A.3 sysdevices.....	425
18.2.3 Server 对象.....	404	A.4 sysdatabases.....	426
18.2.4 Session 对象.....	405	A.5 syscharsets.....	428
18.2.5 Application 对象.....	406	A.6 sysconfigures.....	428
18.3 ASP 与数据对象模型 ADO.....	407	A.7 syscurconfigs.....	429
18.3.1 数据对象模型 ADO.....	407	A.8 syslanguages.....	429
18.3.2 ADO 编程模式.....	407	A.9 sysmessages.....	430
18.4 ASP 对数据库的操作.....	410	A.10 sysprocesses.....	431
18.4.1 ASP 分页显示数据库记录.....	410	A.11 sysservers.....	433
18.4.2 记录操作.....	412	参考文献	434

第1章 数据库系统基础

随着人类社会的发展，信息日益成为各种研究领域以及人们日常生活中不可或缺的一部分，需要对这些信息进行存储、加工、流通和运用。因此，随着人们需要处理的信息量的日益增加，对数据存储格式和管理方法的要求也就越来越高。数据库系统就是在这样的应用领域客观需求下产生并发展的。

本章主要内容



- 数据库发展历史
- 3种数据库模型
- 关系型数据库范式理论
- 数据库规范理论
- E-R模型
- 数据库体系结构

1.1 数据库的发展

数据库就是数据的集合。但是，数据库中的数据并非相互独立地简单归集到一起，而是根据数据之间固有的关系分门别类地存储起来。也就是说，数据库是存储在计算机系统内的有一定结构的数据的集合。下面简单介绍数据库发展的几个阶段。

- 第一个阶段——人工管理阶段 50年代左右，计算机上甚至没有完善的操作系统让用户使用。用户以极原始的方式使用数据，直接存取物理存储器内的数据。
- 第二个阶段——文件系统阶段 典型特征是出现了文件系统。在文件系统阶段，人们使用平面文件（操作系统文件不一定是平面的，例如，UNIX 树形文件系统）来建立数据库，并且使用第三代语言（3GL）访问例程。实际上，一些专用系统仍然是按这种方式建立的，只是进行了修改或根本没有变化。在大型机和微机中依然存在着许多这样的遗留数据库。
- 第三个阶段——数据库系统阶段 由于计算机软硬件技术的飞速发展，带来了数据管理的革命，于是就出现了数据库管理系统（DBMS）。

1970年，E. F. Codd创立了关系模式的概念。1977年，IBM开发了一种叫做System R的原型系统，这是世界上第一个关系数据库系统。随后，各种关系型数据库系统一如雨后春笋，层出不穷。例如SQL/DS（IBM），DB2（IBM），Oracle（Oracle Corp.），DG/SQL（Data General Corp.）和Sybase（Sybase Inc.）。标志着数据库真正进入到关系数据库系统的时代。

在IBM System R产生之前，层次（IMS）和网状（IDMS）模型是常见的。CODASYL（数据系统语言协会）是数据库任务组（Database Task Group, DBTG）创建的一种数据库标准，这是一种基于COBOL语言的网状模型数据库标准，IDMS是CODASYL标准一个商业化的实现。

1968年IBM发布的IMS和20世纪70年代Cullinet Software的IDMS，IMS提出了不同类型的记录通过层次结构相互联系的层次数据模型。例如，银行数据库系统可以把公司实体记录和诸如总部地址、电话号码这样的信息放在层次结构的顶部；接下来是银行的各个业务部门；在每个部门分支下是该部门的出纳员和其他职员记录。当要查询某个出纳员时，程序就会沿着各个分层导航。另一方面，在CODASYL委员会数据库任务组1971年发表的报告的基础上诞生的IDMS，被称为网状模型。网状模型是层次模型的一个推广，某一级的一个记录集合在上一级中可能对应二个不同的层次。

但是，从80年代起，RDBMS已经逐渐地控制了市场，代表性产品包括Oracle、DB2、Sybase、Informix、Microsoft SQL Server和Ingress。

从80年代中后期开始，面向对象（Object-Oriented, OO）的关系DBMS（ORDBMS）已经成为数据库管理系统新一轮研究和开发的热点。并找到了许多适当的应用环境，如在计算机集成制造系统（CIMS）、地理信息系统（GIS）、多媒体系统等等。面向对象DBMS适于在这些领域中应用，因为在一个几乎非事务性的环境中，它们具有控制复型数据类型实力。由于商业竞争的白热化，ORDBMS厂商为了提供包括文本、音频、图像和视频数据类型的面向对象/多媒体性能，已经开发出来商业化的通用服务器。如Oracle8i/Oracle9i。另外，用户定义的数据类型或可扩展类型，已经被扩大或增加到核心数据库服务器中，Oracle8就提供了这样的性能。类似这样的RDBMS产品被认为是混合的，然而它们明显比以前的RDBMS更具有主流性。

此外，多维数据库（Multi-Dimensional Database, MDD）也分享了部分市场份额，这些数据库为带有许多必须被多维存取或列表的变量（例如行为科学数据）的应用提供了高度索引化的数据。在传统的RDBMS中，这几乎是不可能实现的，数据库只允许单独使用。

那么，人们为什么需要使用数据库呢？下面介绍数据库能够为人类带来的若干好处。

- **数据共享** 共享不仅指现有的应用程序可以共享数据库的数据，而且新的应用程序也能对这些数据进行操作。换句话说，不向数据库中添加任何新数据也可能满足新应用程序的数据要求。
- **减少冗余** 在非数据库系统中，每个应用程序都有自己的专用文件。这种情况经常导致在存储数据上有相当大的冗余，结果浪费存储空间。例如，一个有关人事的应用程序和一个有关教育的应用程序可能同时拥有包含职员部门信息的文件。但是，这二个文件可以集成起来消除冗余，只要数据管理员意识到二个应用程序的数据要求——也就是企业应有必要的全局控制与管理策略。
- **避免不一致（某种程度上）** 这是前一点必然的结果。假定雇员 E3 在部门 D8 工作——数据库中有二个不同的条目。还假定 DBMS 也没有意识到冗余的存在（也就是对冗余失控），则必然会有二个记录不一致的情况。当其中一个更新时，另一个不变，这种情况称为数据库不一致。显然，处于不一致状态的数据库可能提供给用户错误的或矛盾

的信息。

当然，如果指定事实是由一条记录表示（如果排除了冗余），那么这样的不一致就不会发生。另一种选择是，冗余没有排除但是受到控制（被 DBMS 得知），那么数据库管理系统就可以保证数据库总是一致的，DBMS 确保二个记录中的任何一条改变会自动地应用到另一条，这一过程即为传播更新。

- 提供事务支持 事务是一个逻辑工作单元，它包括一些数据库操作（特别是一些更新操作）。常见的例子如从帐户 A 到帐户 B 转移一定的现金数。显然，这里要求二个更新操作：一个是从帐户 A 提出现金；另一个是把现金数存入帐户 B。如果用户已经说明二个更新是同一事务的一部分，那么系统要确保二个操作要么都做，要么都不做，即使在系统执行过程中出现故障（比如因为电源断）也应如此。

注意 刚刚举例说明的事务的原子性不是事务支持的惟一优点，但是与其他一些优点不同，它甚至可以应用到单用户的情况。

- 保持完整性 完整性的问题是确保数据库中的数据是正确的。同样事实的二条记录的不一致，就是缺少完整性的例子。当然，只要在存储的数据中有冗余，就会引起这样的问题。即使没有冗余，数据库也可能包含错误的信息。例如，可能显示雇员一周工作了 400 小时而不是 40 小时，或者属于一个不存在的部门。数据库的集中控制可以有效地避免此类问题。做法是通过支持数据管理员定义一些完整性约束（也称为商业规则），由 DBA 加以实施，完整性约束在任何操作执行时都得到有效的检验。

值得指出的是，数据完整性在数据库中要比在各自独立的文件系统中重要得多，因为数据库中的数据是共享的。要是没有正确的控制，有可能一个用户错误地更新数据库而生成的错误数据，会殃及其他无辜的用户。目前，数据库厂商对数据库的完整性约束的支持还相当不够（尽管最近这一方面的情况有所改善）。这一事实很不幸，因为数据库完整性既基本又非常重要。

- 增强安全性 数据库管理员可以确保访问数据库的惟一方式是通过正确的通道，因此可以定义安全性约束或规则。当试图访问敏感数据时，要检查这些安全性约束或规则。对于数据库的每条信息的不同类型的访问（修改、插入或删除等）可建立不同的约束。

注意 没有这样的约束，数据的安全性可能比传统的文件系统更处于危险之中，也就是说，某种意义上数据库系统的集中性要求相称的、好的安全系统。

- 平衡相互冲突的请求 数据库管理员了解企业的全局的需要，在他的指示下能建立系统的结构以提供对企业最佳的全局服务。例如，所选择的数据的物理表示应尽可能使重要的应用以最快的方式访问数据（可能会以降低其他某些应用的访问速度为代价）。
- 加强标准化 数据库管理员对数据库集中控制（在数据管理员的指示下），可以确保所有表示数据的可用标准都可以观察到。可用标准可包括下面的任意一种或全部：部门标准、安装标准、社团标准、工业标准、国家标准和国际标准。标准化的数据表示可以很有效地支持数据交换或者二个系统间的数据移动（随着分布式系统的出现，这一点就越来越重要）。同时，数据命名和文档标准也有效地支持了数据共享和易理解性。以上列出的大多数优点都是比较显而易见的。但是，有一点则不然，这就是对数据的

独立性的支持（严格地说，数据独立性是数据库系统的客观目标，而不仅仅是一个必要的优点）。数据独立性的概念十分重要。

1.2 数据库模型

数据库模型是指数据库中数据的存储结构，根据具体数据存储需求的不同，数据库可以使用多种类型的系统模型，其中较为常见的有结构模型（即层次模型）、网络模型和关系模型 3 种，而选择使用这 3 种模型的数据库就被分别称为层次型数据库、网络型数据库和关系型数据库。下面就针对这 3 种不同类型的数据库进行简单的介绍。

1.2.1 层次型数据库

层次型数据库使用结构模型作为自己的存储结构。这是一种树型结构，它由结点和连线组成，其中节点表示实体，连线表示实体之间的关系。在这种存储结构中，数据将根据需要分门别类地存储在不同的层次之下。图 1-1 是某数据处理公司的层次型数据库模型。

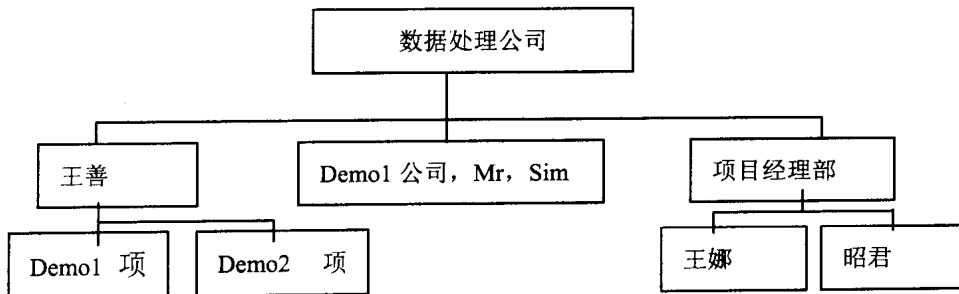


图 1-1 层次型数据库模型范例

(1) 基本结构

层次模型用树形结构来表示各类实体以及实体间的联系。每个结点表示一个记录类型，结点之间的连线表示记录类型间的联系，这种联系只能是父子联系。每个记录类型可包含若干个字段，这里，记录类型描述的是实体，字段描述实体的属性。

任何一个给定的记录值只有按其路径查看时，才能显出它的全部意义，没有一个子女记录值能够脱离父记录值而独立存在。要求：只有一个结点没有父结点，称之为根结点；根以外的其他结点有且只有一个父结点。这就使得层次数据库系统只能处理一对多的实体关系。

(2) 多对多联系在层次模型中的表示

用层次模型表示多对多联系，必须首先将其分解成一对多联系。分解方法有二种：冗余结点法和虚拟结点法。

(3) 基本操作

层次数据模型的操纵主要有查询、插入、删除和更新。进行插入、删除、更新操作时要满足层次模型的完整性约束条件。

- 进行插入操作时，如果没有相应的父结点值就不能插入子结点值。

- 进行删除操作时，如果删除父结点值，则相应的子结点值也被同时删除。
- 进行更新操作时，应更新所有相应记录，以保证数据的一致性。

(4) 存储结构

- 邻接法。按照层次树前序穿越的顺序把所有记录值依次邻接存放，即通过物理空间的位置相邻来实现层次顺序。
- 链接法。用指引元来反映数据之间的层次联系。

(5) 优缺点

优点：

- 数据模型比较简单，操作简单。
- 对于实体间联系是固定的，且预先定义好的应用系统，性能较高。
- 提供良好的完整性支持。

缺点：

- 不适合表示非层次性的联系。
- 对插入和删除操作的限制比较多。
- 查询子结点必须通过父结点。
- 由于结构严密，层次命令趋于程序化。

1.2.2 网络型数据库

网络型数据库使用网络模型作为自己的存储结构。在这种存储结构中，数据记录将组成网络中的节点，而记录和记录之间的关联组成节点之间的连线，从而构成了一个复杂的网状结构。这种类型的数据库管理系统有 Computer Associates 公司的 IDMS。图 1-2 是网络形数据库的范例。

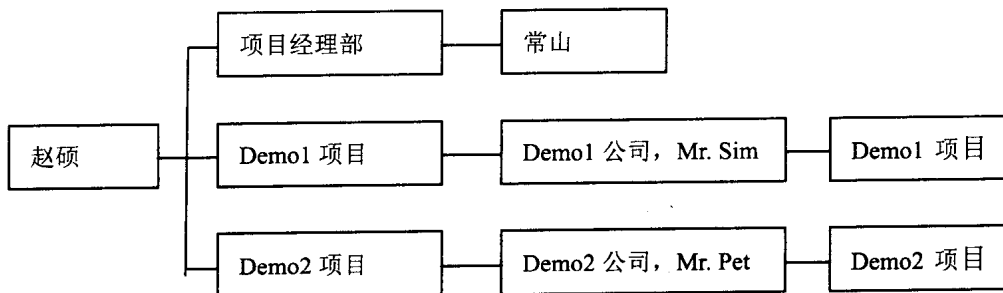


图 1-2 网络型数据库范例

从图 1-2 中可以看出，网络型数据库的节点是惟一的，而节点之间的联系却相当复杂。可以从项目负责人的姓名查出她负责的项目，在从项目查找到客户，而且还能根据客户直接查到客户的其他项目。现在感觉查找相当简单直接，这是因为只看到了整个数据库存储

系统中非常小的一部分。如果记录不是一个二个，而是成千上万，那么整个结构就会变得纷繁复杂、无从下手了。

(1) 基本结构

网状数据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二个限制，允许多个结点没有父结点，允许结点有多个父结点，此外它还允许二个结点之间有多种联系（称之为复合联系）。

(2) 操作

网状数据模型的操纵主要包括查询、插入、删除和更新数据。

- 插入操作允许插入尚未确定父结点值的子结点值。
- 删除操作允许只删除父结点值。
- 更新操作时只需更新指定记录即可。
- 查询操作可以有多种方法，可根据具体情况选用。

(3) 存储结构

网状数据模型的存储结构依具体系统不同而不同，常用的方法是链接法，包括单向链接、双向链接、环状链接、向首链拉等，此外还有其他实现方法，如指引元阵列法、二进制阵列法、索引法等。

(4) 优缺点

优点：

- 能够更为直接地描述现实世界。
- 具有良好的性能，存取效率较高。

缺点：

- 其 DDL 语言极其复杂。
- 据独立性较差。由于实体间的联系本质上通过存取路径指示的，因此应用程序在访问数据时要指定存取路径。

1.2.3 关系型数据库

关系型数据库使用的存储结构是多个二维表格。在每个二维表中，每一行称为一条记录，它用来描述一个对象的信息，每一列称为一个字段，它用来描述对象的一个属性。数据表与数据表之间存在相应的关联，这些关联将被用来查询相关的数据。这种存储结构是目前市场上使用最广泛的数据库模型，使用这种存储结构的数据库管理系统很多，例如，Oracle、Sybase、Informix、Dbase、DB2 等，将要在本书中详细介绍的微软公司的 SQL Server 也是其中之一。图 1-3 是关系型数据库的范例。

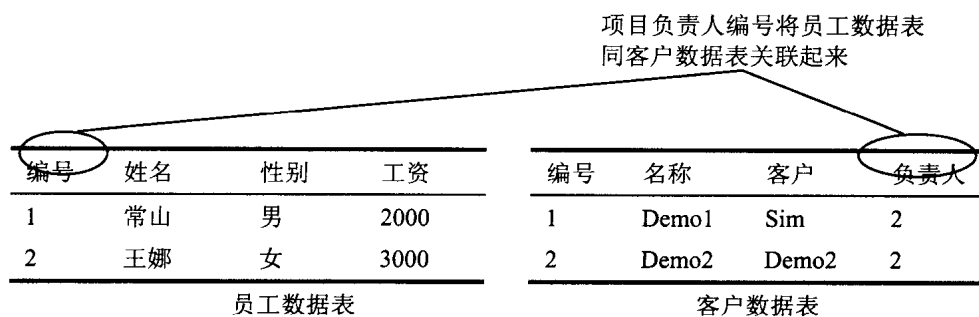


图 1-3 关系型数据库范例

(1) 基本结构

在用户看来，一个关系模型的逻辑结构是一张二维表，它由行和列组成。在关系模型中，实体以及实体间的联系都是用关系来表示。

关系模型要求关系必须是规范化的，最基本的条件就是，关系的每一个分量必须是一个不可分的数据项，即不允许表中还有表。

(2) 操作

关系数据模型的操作主要包括查询、插入、删除和更新数据。这些操作必须满足关系的完整性约束条件。

关系模型中的数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合。关系模型把存取路径向用户隐蔽起来，用户只要指出“干什么”，不必详细说明“怎么干”，从而大大地提高了数据的独立性，提高了用户生产率。

(3) 存储结构

关系数据模型中，实体及实体间的联系都用表来表示。在数据库的物理组织中，表以文件形式存储，每一个表通常对应一种文件结构。

(4) 优缺点

优点：

- 关系模型是建立在严格的数学概念的基础上的。
- 无论实体还是实体之间的联系都用关系来表示。对数据的检索结果也是关系（即表），因此概念单一，其数据结构简单、清晰。
- 关系模型的存取路径对用户透明，从而具有更高的数据独立性，更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。

缺点：

- 由于存取路径对用户透明，查询效率往往不如非关系数据模型。因此为了提高性能，必须对用户的查询请求进行优化，增加了开发数据库管理系统的负担。