

高等院校计算机专业系列教材



高级数据库技术

Advanced Database Technology

王占全 张 静 郑 红 胡国玲 李建华 © 编著



华东理工大学出版社
EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

高级数据库技术

王占全 张 静 郑 红 胡国玲 李建华 编著

图书在版编目(CIP)数据

高级数据库技术 / 王占全, 等编著. —上海: 华东理工大学出版社, 2011. 8

ISBN 978-7-5628-3082-5

I. ①高… II. ①王… 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 141375 号

高级数据库技术

编 著 / 王占全 张静 郑红 胡国玲 李建华

责任编辑 / 纪冬梅

责任校对 / 张 波

封面设计 / 裘幼华

出版发行 / 华东理工大学出版社有限公司

社 址: 上海市梅陇路 130 号, 200237

电 话: (021)64250306(营销部) 64250875(编辑室)

传 真: (021)64252707

网 址: press.ecust.edu.cn

印 刷 / 常熟华顺印刷有限公司

开 本 / 787mm×1092mm 1/16

印 张 / 15.75

字 数 / 382 千字

版 次 / 2011 年 8 月第 1 版

印 次 / 2011 年 8 月第 1 次

书 号 / ISBN 978-7-5628-3082-5/TP·175

定 价 / 48.00 元

(本书如有印装质量问题, 请到出版社营销部调换。)

前 言

数据库技术是计算机科学中一个非常重要的部分,数据库技术以及数据库的应用也正以日新月异的速度发展,高级数据库技术是目前数据库发展趋势的总结,其在空间数据库、内存数据库、时态数据库、Web 数据库、移动数据库、数据挖掘等方面有很大的发展,因此作为现代的大学生,特别是计算机专业的学生,学习和掌握高级数据库知识是非常必要和重要的。

本书是面向计算机专业研究生和高年级本科生学习高级数据库知识而编写的一本教材,其特点是内容全面,既包括数据库的基础理论知识,又包括高级数据库其他方面的知识,如数据挖掘、移动数据库、Web 数据库、空间数据库等。

本书是编者多年从事数据库教学的经验和感受总结。本书的出版得到了华东理工大学出版社优秀教材基金的大力帮助和支持,特别感谢高大启教授,他在我们编写此书的过程中给予了极大的帮助,提出了许多宝贵的意见和建议。

由于编者水平有限,书中难免有不妥之处,敬请广大读者批评指正。

编者

2011年8月

目 录

第 1 章 绪论	1
1.1 数据库系统基本概念	1
1.1.1 数据库系统的由来	1
1.1.2 数据库系统的特点	1
1.1.3 数据库系统的分代	2
1.1.4 数据库系统的分类	3
1.1.5 数据库管理系统	4
1.1.6 典型的关系数据语言	5
1.2 数据库体系结构	7
1.2.1 SPARC 三级模式	8
1.2.2 数据库二级映射	9
1.3 数据模型	10
1.3.1 数据模型的两个层次	10
1.3.2 数据模型的三项要素	10
1.3.3 层次模型和网状模型	11
1.3.4 关系模型	12
1.3.5 对象-关系数据模型	12
1.4 关系的数学方法	13
1.4.1 关系的数学定义	13
1.4.2 关系的性质	14
1.4.3 关系的键(码)	15
1.4.4 关系数据操作	16
1.4.5 创建关系数据库	20
1.5 关系数据库设计及其他	23
1.5.1 关系规范化	23
1.5.2 数据库设计	27
1.5.3 应用程序设计	33
1.6 数据库新技术概述	40
1.6.1 数据库发展回顾	40
1.6.2 数据库系统的新趋势	41
本章小结	44

第 2 章 面向对象数据库	45
2.1 面向对象数据库概述	45
2.2 面向对象设计方法	46
2.2.1 面向对象的特征	46
2.2.2 面向对象分析	48
2.2.3 面向对象分析模型	49
2.2.4 面向对象设计	50
2.3 面向对象数据库	50
2.3.1 面向对象数据模型的含义	51
2.3.2 面向对象数据库模式	53
2.3.3 面向对象数据库的特性	54
2.4 面向对象的数据库设计	56
2.5 对象关系数据库	59
2.6 对象关系数据库模型	61
本章小结	63
第 3 章 Web 数据库	64
3.1 Web 数据库概述	64
3.1.1 Web 数据库的概念	64
3.1.2 Web 数据库的特点	65
3.1.3 Web 数据库与其他概念的区别	65
3.2 Web 数据库系统体系结构	65
3.3 Web 数据库的访问技术	67
3.3.1 基于通用网关接口 CGI	67
3.3.2 基于服务器扩展的 API	68
3.3.3 ODBC/OLE DB	68
3.3.4 基于 JDBC 的 Web 数据库技术	69
3.3.5 对象关系映射	69
3.4 Web 数据库的数据交换技术	70
3.4.1 XML 技术的基本思想	70
3.4.2 XML 及其与 HTML 的比较	71
3.4.3 XML 命名空间	73
3.4.4 文档类型定义与 XML Schema	74
3.4.5 文档对象模型	75
3.4.6 XML 查询语言	75
3.4.7 XML 数据库管理系统	77
3.5 Web 数据库安全	77
3.5.1 审计追踪	77
3.5.2 数据库备份恢复策略	78
3.5.3 数据加密	79

3.5.4 防火墙技术	79
本章小结	80
第4章 多媒体数据库	81
4.1 多媒体数据库概述	81
4.1.1 多媒体数据库的概念	81
4.1.2 多媒体数据库与传统数据库	82
4.1.3 多媒体数据库研究动态	83
4.2 多媒体数据模型	84
4.2.1 关系数据模型	85
4.2.2 面向对象数据模型	85
4.2.3 其他数据模型	86
4.3 多媒体元数据	86
4.3.1 核心数据集元数据标准	87
4.3.2 图像资源元数据标准	88
4.3.3 多媒体元数据标准	90
4.4 多媒体数据库查询	93
4.4.1 多媒体数据库查询中的问题	93
4.4.2 多媒体数据库的查询处理与索引机制	94
4.5 多媒体数据库管理系统	95
4.5.1 多媒体数据库管理系统的基本功能	95
4.5.2 多媒体数据库管理系统的设计模式	96
4.5.3 多媒体数据管理系统的体系结构	97
4.6 多媒体数据的索引和检索	98
4.7 视频信息检索概述	99
4.7.1 视频信息检索系统框架	99
4.7.2 基于低层特征的视频信息检索	101
4.7.3 视频语义信息建模	109
4.7.4 多模态信息融合的视频检索	110
本章小结	112
第5章 移动数据库	113
5.1 移动数据库简单介绍	113
5.1.1 移动数据库特点及研究内容	113
5.1.2 移动数据库的关键技术	114
5.1.3 现有移动数据库产品	118
5.1.4 移动数据库典型系统模型	120
5.1.5 移动数据库系统的研究现状	121
5.2 移动数据库复制与缓存技术	122
5.2.1 数据复制技术	122
5.2.2 移动缓存技术	125

5.3	数据广播技术	130
5.3.1	数据广播系统的组成	132
5.3.2	广播数据的优化调度策略	134
5.3.3	优化访问时间和调谐时间	135
5.4	移动对象的查询处理	138
5.4.1	移动对象轨迹查询	138
5.4.2	移动对象坐标查询	140
5.5	移动事务处理技术	141
5.5.1	移动事务处理的关键问题	142
5.5.2	移动事务的处理模型	143
5.5.3	典型的移动事务处理模型	145
5.6	基于 Agent 的移动数据库模型	149
5.7	移动数据库发展的趋势	152
	本章小结	152
第 6 章	空间数据库	153
6.1	空间数据库概述	153
6.1.1	空间数据库的引入	153
6.1.2	空间数据库的意义	153
6.1.3	空间数据库适合的人群	154
6.1.4	空间数据库的特征	154
6.1.5	空间数据库作为常规数据库扩充	155
6.2	空间数据模型	156
6.2.1	空间数据抽象类型(SADT)	156
6.2.2	空间数据类型	173
6.3	空间对象的操作	176
6.3.1	空间对象之间的关系	176
6.3.2	空间查询语言	178
6.4	空间索引	180
6.4.1	空间索引特点	180
6.4.2	空间对象的近似表示	180
6.4.3	空间索引方法	181
6.4.4	针对 R-树的代价分析	184
6.5	空间数据库系统	188
6.5.1	空间信息系统	188
6.5.2	基于 MapBase 的 GIS 平台	189
	本章小结	191
第 7 章	数据挖掘	192
7.1	数据挖掘概述	192
7.1.1	数据挖掘的概念	192

7.1.2	数据挖掘的流程	193
7.1.3	数据挖掘的功能	194
7.1.4	数据挖掘研究的主要问题	195
7.2	数据挖掘的主要研究内容	196
7.2.1	广义知识	196
7.2.2	关联知识	197
7.2.3	分类和预测型知识	197
7.2.4	聚类知识	197
7.2.5	偏差型知识	197
7.3	关联分析	198
7.3.1	基本概念	198
7.3.2	Apriori 算法	198
7.3.3	FP-Growth 算法	199
7.4	分类和预测	200
7.4.1	基本概念	200
7.4.2	判定树归纳分类	201
7.4.3	贝叶斯分类	204
7.4.4	神经网络	206
7.4.5	遗传算法	208
7.4.6	k -近邻分类法	208
7.4.7	基于案例的推理	208
7.4.8	预测	209
7.5	聚类	210
7.5.1	基本概念	210
7.5.2	划分的方法	210
7.5.3	层次的方法	211
7.5.4	基于密度的方法	212
7.5.5	基于网格的方法	213
7.5.6	孤立点分析	214
7.6	数据挖掘研究的发展趋势	215
	本章小结	216
第 8 章	数据仓库	217
8.1	关于数据仓库	217
8.1.1	数据仓库的基本概念	217
8.1.2	数据仓库的发展历史	217
8.1.3	数据仓库与其他相关概念区别	218
8.1.4	数据仓库管理所面临的问题	219
8.2	数据仓库的体系结构	220
8.2.1	数据仓库的基本体系结构	220

8.2.2 数据仓库的特征	221
8.3 数据仓库关键问题	224
8.3.1 数据仓库的目标数据库	224
8.3.2 数据抽取和转换工具	224
8.3.3 前端数据访问和分析工具	225
8.3.4 数据挖掘技术和工具	226
8.3.5 数据仓库的多维建模	226
8.4 数据仓库开发流程	230
8.5 数据仓库产品实例	232
8.6 数据仓库发展的趋势	235
本章小结	236
参考文献	237

第1章 绪论

计算机应用日新月异,几乎已无处不在,无时不在。但无论应用的范围多么广泛,形式怎样变化,所涉及的技术不外乎“信息处理”和“信息管理”两个方面。前者包括信息的获取、表示、加工/计算、转换等技术,后者包括信息的存储、组织、查询使用等技术。数据库系统主要讨论信息管理的技术,在高校计算机课程中占有重要的地位。

1.1 数据库系统基本概念

1.1.1 数据库系统的由来

数据库系统是在 20 世纪 60 年代后期兴起的一种数据管理技术。在数据库出现前,计算机用户就使用数据文件来存放数据。常用高级语言从早期的 FORTRAN 到今天的 C 语言,都支持使用数据文件。一种常见的数据文件的格式是,一个文件包含若干个“记录”(record),一个记录又包含若干个“数据项”(data item),用户通过对文件的访问实现对记录的存取。通常称支持这种数据管理方式的软件为“文件管理系统”,它自 20 世纪 50 年代末问世以来,一直是操作系统的重要组成部分。

随着社会信息量的迅速增长,计算机处理的数据量不断增加。文件管理系统采用的一次最多存取一个记录的访问方式,以及在不同文件之间缺乏相互联系的结构,越来越不能适应管理大量数据的需要。于是数据库系统应运而生,并在 1969 年诞生了世界上第一个商品化的数据库系统——美国 IBM 公司的 IMS 系统(information management system)。

从文件管理到数据库管理,代表了两代不同的数据管理技术。今天,数据库管理已成为计算机信息管理的主要方式。但在数据量较小的一些场合中,由文件管理系统支持的数据文件仍在使用。

1.1.2 数据库系统的特点

与基于数据文件的应用系统(以下简称为文件应用系统)比较,数据库系统有以下特点。

1. 数据的结构化

在文件应用系统中,各个文件不存在相互联系。从单个文件来看,数据一般是有结构的;但从整个系统来说,数据在整体上又是没有结构的。数据库系统则不同,在同一数据库中的数据文件也存在联系,即在整体上服从一定的结构形式。

2. 数据共享

共享是数据库系统的目的,也是它的重要特点。一个库中的数据不仅可为同一企业或机构之内的各个部门共享,也可为不同单位、地域甚至不同国家的用户共享。而在文件应用

系统中,数据一般是由特定的用户专用的。

3. 数据独立性

在文件应用系统中,数据结构和应用程序相互依赖,一方的改变总是要影响另一方的改变。数据库系统则力求减小这种相互依赖,实现数据的独立性。虽然目前还未能完全做到这一点,但较之文件应用系统已大有改善。

4. 可控冗余度

数据专用时,每个用户拥有并使用自己的数据,难免有许多数据相互重复,这就是冗余。实现共享后,不必要的重复将全部消除,但为了提高查询效率,有时也保留少量重复数据,其冗余度可由设计人员控制。

表 1.1 以对照的形式,列出了数据库系统与一般文件应用系统的主要性能差别。

表 1.1 数据库系统与一般文件应用系统的性能对照

序号	文件应用系统	数据库系统
1	文件中的数据由特定的用户专用	库内数据由多个用户共享
2	每个用户拥有自己的数据,导致数据重复存储	原则上可消除重复。为方便查询允许少量数据重复存储,但冗余度可以控制
3	数据从属于程序,两者相互依赖	数据独立于程序,强调数据的独立性
4	各数据文件彼此独立,从整体看是“无结构”的	各文件的数据相互联系,从整体看是“有结构”的

1.1.3 数据库系统的分代

经过将近四十年的发展,数据库系统已走过了第一、第二两代——格式化数据库系统和关系型数据库系统,现正向第三代——对象-关系数据库系统迈进。

1. 格式化数据库系统

格式化数据库系统是对第一代数据库系统的总称,其中又包括层次型数据库系统与网状型数据库系统两种类型。这一代数据库系统的共同特征是:

(1) 采用“记录”为基本的数据结构。在不同的“记录型”(record type)之间,允许存在相互联系;

(2) 一次查询只能访问数据库中的一个记录,存取效率不高,操作也比较麻烦。

自 20 世纪 80 年代以来,格式化数据库系统逐渐被关系型数据库系统所取代,目前仅在少数大、中型计算机系统中继续使用。

2. 关系型数据库系统(relational database system, RDBS)

早在 1970 年,IBM 公司 San Jose 研究实验室的研究员科德(E. F. Codd)就发表了题为“大型共享库集(data bank)的关系模型”的论文,开创了关系数据库理论的研究。20 世纪 70 年代中期,国外已有商品化的 RDBS 问世,标志着数据库系统进入第二代。随着功能强大的“结构化查询语言”(structured query language, SQL 语言)于 20 世纪 80 年代被采纳为国际标准,更使 RDBS 如虎添翼,因而迅速取代了格式化数据库系统,在包括 PC 在内的各型计算机上纷纷实现。

目前使用的数据库系统主要是第二代数据库系统。与第一代相比,它具有以下优点:

(1) 采用人们习惯使用的表格作为基本的数据结构,简单明了,使用与学习都很方便;

(2) 一次查询仅用一条命令或语句,即可访问整个“关系”(或二维表),效率远较格式化数据库系统高。通过多表联合操作(也称为“多库”操作),还能对互有联系的若干二维表实现“关联”查询。

3. 对象-关系数据库系统(object-relational database system, ORDBS)

关系型数据库系统管理的信息,可包括字符型、数值型、日期型等多种类型,但本质上都属于单一的文本(text)信息。随着多媒体应用的扩大,人们对数据库提出了新的需求,希望数据库系统能存储图形、声音等复杂的对象,并能实现复杂对象的复杂行为。将数据库技术与面向对象技术相结合,便顺理成章地成为研究数据库技术的新方向,构成第三代数据库系统的基础。

20 世纪 80 年代中期以来,对于面向对象的数据库系统(object-oriented database system, OODBS)的研究十分活跃。1989 年和 1990 年,《面向对象数据库系统宣言》和《第三代数据库系统宣言》相继被发表,后者主要介绍对象-关系数据库系统(ORDBS)。一批代表新一代数据库系统的商品也陆续推出。由于 ORDBS 是建立在 RDBS 技术之上的,可以直接继承 RDBS 的原有技术和用户基础,所以其发展比 OODBS 更为顺利,正在成为第三代数据库系统的主流。

1.1.4 数据库系统的分类

1987 年,著名的美国数据库专家厄尔曼教授在一篇题为“数据库理论的过去和未来”的论文中,曾把数据库理论概括为 4 个分支:关系数据库理论、分布式数据库理论、演绎数据库和面向对象数据库。今天,关系数据库已在各种类型的计算机上获得普遍的应用,成为当今数据库系统的主流。其余 3 个分支,在过去二十余年也取得了不小的进展,并在理论研究的基础上开发出各种实用的数据库系统。现择要简述如下。

1. 面向对象数据库

如前所述,数据库的分代是根据所采用的数据模型划分的。这里的所谓数据模型,首先是指把数据组织起来所采用的数据结构,同时也包含数据操作和完整性约束等要素。与第一代数据库常见的层次模型和网状模型相比,关系模型不仅简单易用,理论也比较成熟,但如果用它们来存储和检索包括图、文、声、像在内的多媒体数据,就很难满足需要了。所以当面向对象技术兴起后,人们就探索用对象模型来组织多媒体数据库,推动并促进了第三代数据库——对象式数据库的诞生。

多媒体数据库是面向对象数据库的重要实例。它管理的数据不仅容量大,而且长短不一;检索方法也从传统数据库的“精确查询”,改变为以“非精确匹配和相似查询”为主的“基于内容”的检索。20 世纪 90 年代,一些著名的第二代数据库如 Oracle、SyBase 等都在原来关系模型的基础上引入了对象机制,扩展了对于多媒体数据的管理功能。1998 年,据称是世界上第一个“真正面向对象的”多媒体数据库——Jasmine 数据库在美国问世。

2. 分布式数据库

如果说多媒体应用促进了面向对象数据库的发展,则网络的应用与普及,无疑是推动分布式数据库发展的主要动力。在早期的数据库中,数据都是集中存放的,即所谓的集中式数据库。分布式数据库则把数据分散存储在网络的多个结点上,彼此用通信线路连接。例如,

一个银行有众多储户。如果他们的数据集中存放在一个数据库中,所有的储户在存、取款时都要访问这个数据库,网络通信量必然很大。若改用分布式数据库,将储户的数据分散存储在离各自住所最近的储蓄所,则大多数时候数据可就近存取,仅有少数数据需远程调用,从而大大减少了网上的数据传输量。今天在 Internet/Intranet 上流行的 Web 数据库,就是分布式数据库的实例。它使全城(市)的储户通过同一银行的任何一个储蓄所,都能够实现通存通兑。

分布式数据库也是多用户数据库,可供多个用户同时在网络上使用。但多用户数据库并非总是分布存储的。以飞机订票系统为例,它允许乘客在多个售票点进行订票,但同一航空公司的售票数据通常是集中存放的,而不是分散存放在各个售票点上。

3. 演绎数据库

传统数据库存储的数据都代表已知的事实,演绎数据库(deductive database)则除存储事实外,还能存储用于逻辑推理的规则。例如,某演绎数据库存储有“科长领导科员”的规则。如果它同时存有“甲是科长”、“乙是科员”等数据,它就能推理得出“甲领导乙”的新事实。

由于这类数据库是由“事实+规则”所构成,所以有时也称为“基于规则的数据库”(rule-based database)或“逻辑数据库”(logic database)。它采用的数据模型则称为逻辑模型(logic data model)或基于逻辑的数据模型。

随着人工智能不断走向实用化,对演绎数据库的研究也日趋活跃,与专家系统和知识库(knowledge base)一起被称为智能数据库。它们的共同关键是逻辑推理,如果推理模式出了问题,便可能导致荒诞的结果。

此外还有很多数据库分支,如空间数据库、多媒体数据库、Web 数据库等。

1.1.5 数据库管理系统

数据库管理系统(database management system, DBMS)是处于用户(应用程序)和操作系统之间的一种软件,其作用是对数据库中的数据实施有效的组织与管理。无论开发还是运行数据库系统,都需要 DBMS 的支持。

1. DBMS 的基本功能

由于数据库的建立和查询都是通过数据(库)语言进行的,所以 DBMS 首先要具有支持某一特定数据语言的功能,例如关系数据库通常都支持 SQL 语言,就像编译程序总是要支持某种高级语言一样。一般来说,数据库管理系统的基本功能应包括下列几个主要方面。

1) 数据定义功能

DBMS 能向用户提供“数据定义语言”(data definition language, DDL),用于描述数据库的结构。以上述的 SQL 为例,其 DDL 一般都设有 Create Table/Index、Alter Table/Drop Table/Index 等语句,分别供用户建立、修改或删除关系数据库的二维表结构,或者定义或删除数据库表的索引。

2) 数据操作功能

对数据进行检索和查询,是数据库的主要应用。为此,DBMS 能向用户提供“数据操作语言”(data manipulation language, DML),支持用户对数据库中的数据进行查询、更新(包括增加、删除、修改)等操作。仍以 SQL 语言为例,其查询语句的基本格式为:



```
Select <查询的字段名>  
From <库表的名称>  
Where <查询条件>
```

这种语句灵活多变,可包含多达十几种子句,使用十分方便。

3) 控制和管理功能

除 DDL 和 DML 两类语句外,DBMS 还具有必要的控制和管理功能。其中包括:在多用户使用时对数据进行的“并发控制”;对用户权限实施监督的“安全性检查”;数据的备份、恢复和转储功能;以及对数据库运行情况的监控和报告等。通常数据库系统的规模越大,这类功能也越强,所以大型机 DBMS 的管理功能一般比 PC 的 DBMS 管理功能更强。

4) 数据通信功能

数据通信功能主要包括:数据库与操作系统的接口以及用户应用程序与数据库的接口。

2. DBMS 的发展现状

随着数据库系统从第一代发展到第三代,DBMS 取得了迅速的发展,目前在计算机上使用的 DBMS 主要是关系数据库管理系统 (relational database management system, RDBMS)。

1964 年,美国通用电气公司成功开发世界上第一个 DBMS——IDS (integrated data store) 系统,奠定了网状数据库系统的基础。1969 年,美国 IBM 公司推出了基于层次模型的 IMS 系统,成为世界上第一个实现商品化的 DBMS 产品。目前网状数据库已几乎不再使用,层次模型数据库早些年在大型计算机中尚有使用,现在也很少见了。

早在 1970 年,E. F. Codd 就开创了关系数据库的理论研究。1974 年,该实验室成功开发了世界上最早的关系数据库系统 System R,随后又陆续推出了 SQL/DS 与 DB2 等商品化产品。1980 年以后,各种 RDBMS 先后问世,其中流行较广的有常用于大、小型计算机的 Oracle、Sybase、Informix,以及常用于 PC 的 dBASE、FoxBASE、VFP、Access 等。

近十年来,一批新一代的商品数据库系统已在欧洲、北美各国陆续推出。除 1.1.4 节已提到的 Jasmine 系统外,比较知名的还有美国 Object Design 公司的 ObjectStore, Versant Object Technology 公司的 Versant,以及法国 O2 Technology 公司的 O2, Ontos 公司的 ONTOS 等例子。有些关系数据库系统商品,也在原有 RDBMS 的基础上扩充了数据库管理系统的功能,使之适应于对象-关系数据库的需要。本书将要介绍的空间数据库、时空数据库,就是随着地理信息系统 (geographical information system, GIS) 的流行而涌现出来的对象-关系数据库系统的新鲜实例。

1.1.6 典型的关系数据语言

随着 SQL 被采纳为国际标准,许多大公司纷纷开发了自己的 SQL 版本。它们都基本遵循标准文本,但内容却不尽相同。犹如普通话从不同地域的人说出口,总伴有方言的特征一般。本节介绍的 Transact-SQL (简称 T-SQL),就是在 SQL Server 2000 中使用的一种版本。

语法简洁、命令子句可灵活组合,在应用开发中能支持 SPARC 三级模式,共同构成了 SQL 语言的主要特征。这也是它深受用户欢迎,成为数据库用户首选语言的原因。

SQL 模仿英语祈使句的语法,每一命令均由一个动词开头,只需说明要求计算机“做什

么”，不必介绍“怎样做”，因而是一种高度“非过程化”的语言。它集 DDL、DML 和 DCL 于一身，仅用了 9 种命令(表 1.2)，就基本上涵盖了关系数据库所需的所有操作。SQL-SELECT 命令的可选子句多达十余条，把它们巧妙地搭配，可完成各种不同的功能。除此之外，SQL 还可提供“自主式”和“嵌入式”两种语言形式。前者如 T-SQL，用户只需直接键入命令，就可对数据库进行操作；后者可嵌入诸如 VB、C 等“宿主式”语言(host language)编写的应用程序，为它们进行数据库操作提供了方便。

表 1.2 T-SQL 的 3 类 9 种命令

功能类型	命令开头的动词
数据定义(数据模式和索引定义、删除、修改)	CREATE、DROP、ALTER
数据操作(数据查询和维护)	SELECT、INSERT、UPDATE、DELETE
数据控制(数据存取控制权,如授权和回收)	GRANT、REVOKE

现在以表 1.2 的数据定义和数据操作命令为例,举例说明 T - SQL 命令的一般用法。

1. CREATE 命令

例:创建学生信息表(学号,姓名,性别,专业号,班级,出生年月,民族,籍贯),可使用以下命令:

```
CREATE TABLE 学生信息表( 学号 char(8),
                        姓名 char(8)NOT NULL,
                        性别 char(2)NOT NULL,
                        专业号 char(6),
                        班级 char(10),
                        出生年月 datetime NOT NULL,
                        民族 char(6),
                        籍贯 char(14),
                        PRIMARY KEY(学号));
```

例:按班级降序、学号升序,为学生信息表建立唯一索引。可使用下列 CREATE UNIQUE INDEX 命令:

```
CREATE UNIQUE INDEX 班级学号 on 学生信息表(班级 DESC,学号 ASC);
```

2. DROP 命令

例:删除学生信息表(学号,姓名,性别,专业号,班级,出生年月,民族,籍贯),可使用以下命令:

```
DROP TABLE 学生信息表
```

例:删除学生信息表已建立的班级学号索引。可使用下列命令:

```
DROP INDEX 学生信息表. 班级学号
```

3. ALTER 命令

例:为学生信息表增加“联系电话”和“照片”两个字段。可使用下列命令:

```
ALTER TABLE 学生信息表 ADD 联系电话 char(14),照片 image;
```

4. SELECT 命令

例:在学生信息表中查看所有上海籍男学生的学号。可使用下列命令:

```
SELECT 学号
FROM 学生信息表
WHERE 籍贯 = "上海" AND 性别 = "男"
```

5. INSERT 命令

例:在上面创建的学生信息表(学号,姓名,性别,专业号,班级,出生年月,民族,籍贯)空白表中插入两个记录,可依次使用以下命令:

```
INSERT INTO 学生信息表
VALUES('20000101','沈吉','女','000001','计应 001','1982-10-16','汉','上海');
INSERT INTO 学生信息表
VALUES('20000102','丁爽云','女','000001','计应 001','1981-11-15','汉','上海');
```

6. UPDATE 命令

例:将成绩表(学号,课程号,成绩)中所有选修课程号 000004 的选课成绩更新为 0。

```
UPDATE 成绩表
SET 成绩 = 0
WHERE 课程号 = '000004';
```

7. DELETE 命令

例:在学生信息表中删除学号为 20030211 的学生记录。

```
DELETE
FROM 学生信息表
WHERE 学号 = '20030211';
```

需要指出的是,作为计算机高级语言,SQL 要求用户在使用命令时遵循严格的语法。以 T-SQL 的 CREATE TABLE 命令为例,其格式一般为:

```
CREATE TABLE <表名>(<列名 1><数据类型>[列级完整性约束]
[,<列名 2><数据类型>[列级完整性约束]] ... [,<表级完整性约束>];
```

显然,这与 SQL 命令简洁明快、易学易用的特征风格,是并不矛盾的。关于完整性约束的含义,将在 1.3.2 节中说明。

1.2 数据库体系结构

1975 年,美国国家标准协会(ANSI)下属的“标准规划和要求委员会”(Standards Planning And Requirements Committee,SPARC)对数据库系统的“体系结构”(architecture)提出了一个标准模型。这一委员会指出,不管实际的数据库系统采用何种模型或有多大差异,其基本结构都可以划分为外模式、概念模式和内模式三级,这就是著名的 SPARC 报告。