21世纪高职高专计算机类规划教材

Oracle 数据库系统原理

李爱武　编著

北京邮电大学出版社
www.buptpress.com

21世纪高职高专计算机类规划教材

TP311.138
384
2007

Oracle数据库系统原理

李爱武　编著

北京邮电大学出版社
·北京·

内 容 简 介

本书包括数据库基本理论以及Oracle数据库系统知识两部分。主要包括关系模型基本理论、ER图及范式理论、SQL语言、Oracle存储空间的管理、索引原理、事务处理及锁、备份恢复原理、存储过程以及触发器、网络连接配置等内容。在注重学习各种技能的同时，更强调理解关系型数据库理论的一些本质内容，能够对Oracle数据库的体系结构及运作原理有深层次的理解。

本书可以作为高校计算机各专业的数据库教材或参考书，也可以供相关Oracle数据库管理员或编程人员参考。

图书在版编目(CIP)数据

Oracle数据库系统原理/李爱武编著. —北京：北京邮电大学出版社，2007

ISBN 978-7-5635-0871-6

Ⅰ.O... Ⅱ.李... Ⅲ.关系数据库—数据库管理系统，Oracle—高等学校—教材 Ⅳ.TP311.138

中国版本图书馆CIP数据核字(2007)第003463号

书　　名：Oracle数据库系统原理

编　　著：李爱武

责任编辑：张佳音

出版发行：北京邮电大学出版社

社　　址：北京市海淀区西土城路10号(100876)

北方营销中心：电话　010-62282185　传真　010-62283578

南方营销中心：电话　010-62282902　传真　010-62282735

E-mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：北京忠信诚胶印厂

开　　本：787 mm×1 092 mm　1/16

印　　张：13.75

字　　数：325千字

印　　数：1—3 000册

版　　次：2007年3月第1版　2007年3月第1次印刷

ISBN 978-7-5635-0871-6/TP·136　　定价：21.00元

·如有印装质量问题，请与北京邮电大学出版社营销中心联系·

前　言

数据库工业经过近40年的发展，已经度过了群雄逐鹿的时期。当今的数据库市场主要被Oracle、微软、IBM这3家公司控制，而Oracle不论在技术或是市场占有率方面都遥遥领先于其他产品。

本书包括关系型数据库基本理论以及Oracle数据库系统知识两部分内容。

在对关系型数据库基本理论的处理上，未使用严格的理论化描述，而是力争用比较简短的篇幅以及通俗的语言讲清其最本质的东西，主要选择了以下几个方面的内容：

(1) Codd提出关系模型时，当时的数据库产品主要面临什么问题；

(2) Codd提出的关系模型如何解决这些问题；

(3) 关系模型中的关系与表有哪些区别；

(4) 关系代数的作用及其主要运算；

(5) 为什么实体完整性及引用完整性比其他完整性约束更重要；

(6) ER图及范式理论。

在Oracle数据库系统知识方面，强调Oracle数据库体系结构及其内在原理，并构造合适的简单实验验证各种结论，使得读者在掌握Oracle数据库基本技能的同时，能够对其运作原理有深层次的理解，能够学会探索一个DBMS运作原理的方法，为以后使用Oracle数据库或其他数据库产品打下坚实的基础。这是本书与其他数据库书籍及Oracle书籍不同的地方。在具体内容方面主要有下面几个特点：

(1) 存储结构方面强调了存储空间的分配及释放；

(2) 在数据库体系结构方面详细解释了各种物理文件的作用及各种内存区域的结构；

(3) 比较彻底地研究了Oracle的索引结构，用实验对比了在索引创建前后查询效率的差异，讲述了使用索引时应该注意的各种情况；

(4) 系统分析了Oraclc中的事务处理及锁的原理；

(5) 详细解释了网络连接的各种不同情形，讲解了专用服务器及共享服务器模式的不同体系结构，以及服务器端与客户端配置方法，解释了客户端采用非默认端口连接服务器的时候，客户端及监听器的配置方法。

(6) 在备份恢复部分，强调了Oracle备份恢复的原理，详细解释了Oracle的归档模式的设置方法以及归档日志文件的使用方式，并讲解了在丢失联机重做日志文件的情况下，如何打开数据库，通过这些内容使得读者对重做文件的功能有更高层次的理解。

本书的代码均在 Oracle 10g 版本执行，但是多数内容并未局限于某个特定版本，所涉及的原理及各实验的代码在 8i 及 9i 版本上都是适用的。

本书可以作为高校计算机各专业的数据库教材或参考书，也可以供相关 Oracle 数据库管理员或编程人员参考。

作　者
2007 年 3 月

此为试读，需要完整PDF请访问：www.ertongbook.com

目　录

第 1 章　数据库技术基础

1.1　数据库应用的场合 …… 1
1.2　常用术语 …… 1
1.3　数据库技术的产生和发展 …… 2
1.4　数据库系统的构成 …… 3
1.4.1　硬件 …… 3
1.4.2　软件 …… 3
1.4.3　人员 …… 3
1.5　关系型数据库 …… 4
1.5.1　数据模型 …… 4
1.5.2　关系模型 …… 5
1.6　常用关系型 DBMS 介绍 …… 8
1.7　数据库技术研究的领域 …… 9

第 2 章　ER 模型

2.1　数据库设计的主要步骤 …… 10
2.2　ER 图的表示方法 …… 10
2.3　联系的映射约束 …… 11
2.4　ER 图转化为表 …… 12

第 3 章　规范化理论

3.1　引入范式理论的原因 …… 14
3.2　第一范式 …… 15
3.3　第二范式 …… 17
3.4　第三范式 …… 17

第 4 章　SQL 语言

4.1　SQL 语言概述 …… 19
4.1.1　SQL 语言的特点 …… 19
4.1.2　SQL 语言的分类 …… 19
4.2　SQL * Plus 的使用方法 …… 20

4.2.1 Oracle 中的预置用户 …… 21
4.2.2 连接数据库的方式 …… 22
4.2.3 使用 SQL * Plus …… 22
4.2.4 Windows 平台下有关 Oracle 数据库的常用注册表项 …… 26
4.3 简单查询 …… 26
4.3.1 select-from-where 结构 …… 26
4.3.2 设置查询结果字段的别名及排序 …… 27
4.3.3 where 关键字的使用 …… 28
4.3.4 汇总函数 …… 31
4.3.5 group by 及 having 子句 …… 32
4.3.6 理解 NULL …… 33
4.4 update、delete、insert 语句 …… 34
4.5 复杂查询 …… 35
4.5.1 表连接 …… 35
4.5.2 集合运算 …… 38
4.5.3 子查询 …… 40
4.6 exists 与 not exists 的用法 …… 41

第 5 章 表及完整性约束

5.1 数据类型 …… 47
5.2 创建简单的表 …… 48
5.3 完整性约束 …… 49
5.4 创建带有约束的表 …… 50
5.5 利用数据字典视图查询约束信息 …… 54
5.6 修改表的结构 …… 55

第 6 章 Oracle 数据库的存储结构

6.1 Oracle 中的几个逻辑存储概念 …… 57
6.2 表空间种类 …… 58
6.3 创建表空间 …… 59
6.4 存储空间分配 …… 60
6.4.1 autoallocate 的情形 …… 60
6.4.2 uniform size 的情形 …… 62
6.4.3 有关 uniform size 参数使用的一个误解 …… 63
6.4.4 由多个数据文件构成的表空间 …… 64
6.5 删除数据或删除表对其存储空间的影响 …… 68
6.5.1 delete 操作对表使用空间的影响 …… 68
6.5.2 truncate 操作对表使用空间的影响 …… 70
6.5.3 drop table 对表使用空间的影响 …… 71

6.6　数据文件管理…… 73

第 7 章　Oracle 数据库体系结构

7.1　与 Oracle 数据库相关的物理文件 …… 77
7.1.1　警告文件…… 77
7.1.2　口令文件…… 79
7.1.3　初始化参数文件…… 80
7.1.4　数据文件…… 85
7.1.5　临时数据文件…… 85
7.1.6　控制文件…… 85
7.1.7　重做日志文件…… 87
7.2　控制文件与重做日志文件的管理…… 88
7.2.1　备份控制文件…… 88
7.2.2　解决丢失部分控制文件问题…… 89
7.2.3　解决丢失全部控制文件及重做文件问题…… 90
7.2.4　重做文件的管理…… 91
7.3　Oracle 实例的内存结构及后台进程 …… 92
7.3.1　SGA 的构成 …… 93
7.3.2　PGA 的构成 …… 94
7.3.3　实例中的进程…… 95
7.4　数据库的启动和关闭…… 96
7.4.1　数据库的启动…… 96
7.4.2　数据库的关闭…… 96
7.5　在 Oracle 中手工建立数据库 …… 97

第 8 章　数据字典

8.1　查询所有的数据字典名称及其数量 …… 102
8.2　常用静态数据字典视图 …… 103
8.3　常用动态数据字典视图 …… 108

第 9 章　索引原理

9.1　在 SQL * Plus 中查看执行计划 …… 109
9.2　影响查询效率的主要因素 …… 111
9.3　Oracle 中如何创建索引 …… 111
9.4　应用索引提高查询速度的检验 …… 112
9.5　索引的结构 …… 114
9.6　通过导出数据块查看索引结构 …… 117
9.7　聚簇索引与非聚簇索引 …… 122
9.8　需要创建索引的场合 …… 123

9.9 索引未被使用的原因 …… 125
9.10 索引与空值 …… 130
9.11 DML 语句对索引的影响 …… 133
9.11.1 insert 语句对索引的影响 …… 133
9.11.2 delete 语句对索引的影响 …… 136

第 10 章 视图

10.1 视图的创建和使用 …… 138
10.2 视图的作用 …… 140
10.3 更新视图 …… 142

第 11 章 事务处理与锁

11.1 事务概念 …… 145
11.2 事务的 ACID 属性 …… 145
11.3 事务控制命令 …… 146
11.4 commit 的作用 …… 148
11.5 事务隔离级别 …… 149
11.6 锁 …… 153
11.6.1 insert 操作被阻塞 …… 154
11.6.2 update 或 delete 操作被阻塞 …… 154
11.6.3 死锁 …… 155
11.6.4 锁的种类 …… 156

第 12 章 用户及权限管理

12.1 几个特殊用户 …… 158
12.2 用户属性 …… 158
12.3 创建用户 …… 159
12.4 修改用户的属性 …… 159
12.5 删除用户 …… 160
12.6 权限种类 …… 161
12.7 授予权限及收回权限 …… 162
12.7.1 授予系统权限 …… 162
12.7.2 查询用户的系统权限 …… 162
12.7.3 收回系统权限 …… 163
12.7.4 授予对象权限 …… 164
12.8 管理角色 …… 166
12.8.1 常用的预定义角色 …… 166
12.8.2 创建角色 …… 167
12.9 关于 sys 用户的操作系统验证 …… 168

第 13 章　网络连接

13.1　简单的网络配置…… 169
13.1.1　服务器端配置…… 169
13.1.2　客户端配置…… 171
13.2　复杂的网络连接配置…… 176
13.2.1　共享服务器模式及专用服务器模式…… 176
13.2.2　配置共享服务器模式…… 176
13.2.3　在客户端指定连接方式…… 179
13.2.4　使用服务名连接数据库…… 180
13.2.5　使用非默认端口连接数据库…… 182

第 14 章　存储过程以及触发器

14.1　PL/SQL 程序的基本结构…… 186
14.2　简单的 PL/SQL 程序设计 …… 188
14.2.1　声明…… 188
14.2.2　使用%type 和%rowtype …… 189
14.3　游标和异常…… 190
14.3.1　显式游标…… 190
14.3.2　隐式游标…… 191
14.4　异常处理…… 192
14.4.1　Oracle 预定义异常 …… 192
14.4.2　用户定义的异常…… 193
14.5　触发器…… 194
14.5.1　语句触发器…… 194
14.5.2　行触发器…… 195
14.5.3　instead of 触发器 …… 196
14.6　存储过程…… 197

第 15 章　Oracle 数据库备份恢复原理

15.1　逻辑备份实践…… 200
15.2　设置归档日志模式…… 201
15.3　物理备份恢复实践…… 202
15.4　有关备份恢复的几个概念…… 205
15.5　不完全恢复示例…… 206
15.6　联机重做日志组丢失后的处理…… 208

参考文献…… 209

第1章　数据库技术基础

1.1　数据库应用的场合

近年来，随着多媒体技术、空间数据库技术和计算机网络的飞速发展，数据库系统的发展十分迅速，应用领域愈来愈广。以下领域是数据库系统应用的一些典型场合：

(1) 企事业单位、政府部门的行政管理、办公自动化；

(2) 银行财务管理；

(3) 铁路、民航票务预定系统；

(4) 宾馆、酒店房间预定系统；

(5) 百货公司订购与销售系统；

(6) 医院病房、病历管理；

(7) 学校师资、教学、学生学籍管理；

(8) 图书馆管理；

(9) 气象预报、地震预测、地质勘探；

(10) 电子商务；

(11) 电信行业。

可以说，人们的生产、生活与数据库技术息息相关，社会的发展也已经离不开数据库技术。

这里介绍数据库的3个常用术语。

1.2　常用术语

1. 数据库

数据库(Database，DB)是存放数据的仓库，它是长期存储在计算机内的、有组织、有结构的大量的可共享的数据集合。其数据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，并可以供各种用户共享。

2. 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。数据库管理系统的作用是科学、有效地组织和存储数据，高效地获取和维护数据。

3. 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。

此为试读，需要完整PDF请访问：www.ertongbook.com

1.3 数据库技术的产生和发展

从计算机工业早期开始，存储和处理数据一直是应用的核心，数据库技术是计算机科学中发展最快的领域之一，也是应用最广的技术之一，它已经成为了计算机信息系统与应用系统的核心技术和重要基础。

20 世纪 60 年代初，通用电气(General Electric，GE)公司的 Charles Bachman 设计了第一个通用功能的 DBMS，称为 IDS(Integrated Data Store)，这也奠定了网状数据模型(Network Data Model)的理论基础，网状数据模型经过了 CODASYL(Conference on Data System Language)的标准化后，对 20 世纪 60 年代的数据库系统产生了巨大影响。Bachman也因为其对数据库技术的贡献，在 1973 年获得计算机科学的最高奖——ACM 图灵奖，他也是第一个因为对数据库技术的贡献而获此殊荣的学者。

在 20 世纪 60 年代末期，IBM 开发了 IMS(Information Management System)。IMS 奠定了另外一种数据模型——层次模型(Hierarchical Data Model)——的理论基础。也是在这个时期，美国航空公司(American Airlines)与 IBM 公司合作开发了 SABRE 系统，它允许几个用户通过网络同时访问相同的数据，至今，SABRE 系统还被 Travelocity(世界著名旅游网站)用于为客户提供旅游服务。

在数据库技术发展的历史上，1970 年是发生伟大转折的一年，这一年，IBM 公司的 E. F. Codd[①]博士在其论文“A Relational Model of Data for Large Shared Data Banks”中提出了一种新的数据模型，称为关系数据模型，这篇论文被普遍认为是数据库系统发展史上具有划时代意义的里程碑[②]。

在关系模型提出之后，一大批商用关系数据库系统很快被开发出来并迅速占领了市场，其交替速度之快、除旧布新之彻底是软件史上所罕见的。基于 20 世纪 70 年代后期到 80 年代初期这一引人注目的现象，1981 年的图灵奖很自然地授予了这位“关系数据库之父”——E. F. Codd 博士。

1970 年以后，Codd 继续致力于完善与发展关系理论。1972 年，他提出了关系代数和关系演算的概念，定义了关系的并、交、投影、选择、连接等各种基本运算，为日后成为标准的 SQL 奠定了基础。

在 Codd 提出关系模型以后，IBM 投巨资开展关系数据库管理系统的研究，其“System R”项目的研究成果极大地推动了关系数据库技术的发展[③]，在此基础上推出的 DB2 和 SQL 等产品成为 IBM 的主流产品。

20 世纪 80 年代，关系模型居于主导地位，数据库系统开始得到更广泛的应用。SQL 语言作为 IBM 的 System R 项目的一部分被开发出来，现在已经成为关系型数据库的标准操作语言。

---

① 即 Edgar Frank Codd。

② ACM 在 1983 年把这篇论文列为从 1958 年以来的 25 年中最具里程碑意义的 25 篇论文之一。

③ System R 本身作为原型并未问世，但鉴于其影响，ACM 还是把 1988 年的“软件系统奖”授予了 System R 开发小组。

1984 年，IBM 公司推出了 DB2 数据库产品。由于 IBM 公司当时在计算机工业无可比拟的巨人地位，使得数据库领域内关系模型数据库与网状模型数据库的纷争立刻有了结论。基于网状模型及层次模型的数据库产品迅速走向衰败以致消亡。从这个时期一直到 20 世纪 90 年代，各种关系型数据库产品像雨后春笋般涌现出来。

当今的数据库工业经过 40 余年的发展，已经度过了群雄逐鹿的时期。数据库市场主要被 IBM 公司、微软公司以及 Oracle 公司这 3 家公司控制，其产品都是关系型数据库。在近期，数据库行业在技术上很难再有像 20 世纪 70 年代关系数据模型那样巨大的突破。

1.4　数据库系统的构成

数据库系统一般由硬件、软件及各种人员构成，下面分别介绍。

1.4.1　硬件

数据库系统的数据量一般很大，加之 DBMS 要提供复杂的功能，因此整个数据库系统对硬件资源的要求很高，要有足够大的内存、足够大的硬盘以及足够多的 CPU，这是数量的要求；另外，数据库服务器一般要保持常年 24 小时运转，所以更重要的是对硬件的质量和性能的要求，能够用作数据库服务器硬件的一般都是各计算机厂商生产的专用服务器，比如 SUN、IBM、HP 等著名计算机厂商的服务器产品，甚至是小型机、中型机、大型机，这些机器一般都使用专用的 CPU 和内存，硬盘则采用磁盘阵列，根本不同于普通用户平时使用的 PC 机上的硬件，其性能和价格也是 PC 机上的硬件不能相比的。

1.4.2　软件

数据库系统的软件主要包括操作系统、DBMS 及前端应用软件。

大型企事业单位的数据库服务器一般采用专用服务器，而多数高端服务器厂商的服务器产品一般只能安装专用的 UNIX 操作系统，如 SUN 公司的服务器产品只能安装 Solaris 操作系统(SUN 公司出品，UNIX 各种版本中比较常用的一种)，而 IBM 公司的服务器有的只能安装 AIX 操作系统(IBM 公司的 UNIX 版本)；中小企事业单位，因为其数据量中等，一般也不需要 24 小时运行，多数使用 PC 服务器，安装 Windows 操作系统平台。

根据实际应用的要求不同，所需要的 DBMS 也不同。一般单用户应用使用各种桌面数据库产品就足够了，而大型、关键的应用[①]一般采用大型数据库产品，一些流行产品的介绍请参考本书其他章节的内容。

前端应用软件一般由另外的软件公司根据企业的具体要求进行开发。

1.4.3　人员

数据库系统所涉及的人员主要包括系统分析人员和数据库设计人员、应用程序开发人员、数据库管理员及最终用户。

---

① 关键应用一般指涉及金融或财务的应用。

系统分析人员负责应用系统的需求分析和规范说明，和用户及 DBA 相交流确定系统的软硬件配置，并参与数据库系统的概要设计。数据库设计人员负责数据库中数据的确定。数据库设计人员必须参加用户需求调查和系统分析。

应用程序开发人员负责设计和编写应用系统的程序模块及前端界面，并进行安装和调试，对于关系型数据库，一般要求程序开发人员精通相应 DBMS 存储过程的编写方法，并且要精通前台开发语言的使用。

数据库管理人员负责全面管理和维护数据库系统，具体职责如下。

(1) 决定数据库的存储结构和存取策略。主要是从安全和效率的方面考虑，不同的数据和不同的文件应该在多个磁盘上如何分布。

(2) 确定数据库的安全性要求，主要指用户和权限控制。

(3) 负责数据库的备份和恢复，主要是备份策略的制订和实施，数据库出现故障后的恢复实施。

(4) 监控数据库的运行，主要是空间的使用情况、发现操作出现低效率情况时的原因诊断及解决。

以上人员都要求精通数据库理论及其所使用的 DBMS 产品，另外数据库管理员还要熟悉相关操作系统产品的原理及操作，并具备一定的程序开发水平。

最终用户指数据库系统要服务的对象，这些人一般不需要懂数据库技术，程序开发人员所设计的前台界面主要就是供这部分人使用，比如，使用取款机取款的人，就是银行的数据库系统的最终用户。

1.5 关系型数据库

1.5.1 数据模型

通俗地讲，数据模型就是现实世界数据特征的模拟和抽象。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和信息。

数据模型的三要素是数据结构、数据操作、数据的约束条件。

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，是对系统静态特性的描述。数据结构是刻画一个数据模型性质最重要的方面，不同的数据模型具有不同的数据结构形式。

数据操作是指对数据库中各种对象(型)的实例(值)允许执行的操作的集合，包括操作及有关的操作规则，是对系统动态特性的描述。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据规则，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。

1.5.2　关系模型

1. 关系数据模型要解决的问题

IBM 公司的 IMS 数据库系统的数据独立性比较差，当数据库的逻辑结构或物理结构发生改变时，使用 IMS 的程序员要花费大量时间用在对应用程序的修改和维护上。这就是 Codd 在 1970 年发表他的著名论文“A Relational Model of Data for Large Shared Data Banks”时的背景，也是 Codd 提出关系数据模型要解决的主要问题，他的目的是构造一种能够更好地提供数据独立性的数据模型。

2. Codd 的建议

Codd 的建议包括以下 3 个方面。

(1) 用一种简单的数据结构存储数据。

(2) 通过一种更高层次的面向集合的语言访问数据。

(3) 不需要说明数据的物理存储方式。

用简单的数据结构存储数据，可以提供更好的逻辑数据独立性，也就是当改变数据的逻辑结构时，应用程序不需要很大修改；使用更高层次的操作语言，可以提供更好的物理数据独立性。这样也不需要像 IMS 与 CODASYL 那样要规定数据的物理存储方式了。

3. 关系数据模型的主要特征

虽然关系模型已经提出了 40 余年，在各方面都有了很大发展，但关系数据模型的核心特征基本未发生太大的变化，这也是理论与技术的一个显著区别。可以从构成数据模型的 3 个方面考查关系模型的特征。

(1) 关系数据模型的数据结构

关系数据模型用于存储数据的逻辑结构是关系(relation)，关系是一个数学概念，由元组(tuple)和属性(attribute)构成。可以把关系看作由行(row)和列(column)构成的表，这里的行对应关系中的元组，列对应关系中的属性。在关系型数据库中，行也称为记录(record)，列也称为字段(field)，图 1-1 所示为一个企业的员工表及部门表，可以分别称为 emp 表及 dept 表。

emp 表

empno	ename	job	sal	deptno
7369	SMITH	CLERK	800	20
7499	ALLEN	SALESMAN	1600	30
7521	WARD	SALESMAN	1250	30
7566	JONES	MANAGER	2975	20

dept 表

deptno	dname	loc
10	ACCOUNTING	NEW
20	RESEARCH	DALLAS
30	SALES	CHICAGO

图 1-1　一个企业的员工表及部门表

严格来说，关系模型中的关系与表有如下关键区别[①]。

① 这里也是关系模型中的关系与关系型数据库中的表的区别。

① 关系是元组的集合，在数学中，集合中的元素不能重复，元素之间也没有前后顺序，相应地，关系中的元组不能重复，元组之间没有前后顺序；而表中的行可以重复，一般会有顺序。

② 关系的属性没有从左到右的顺序，而表上的列一般有固定的顺序。

③ 关系的属性不能为多值，而表中的列可以是多值的，如可以为图 1-1 中的 dept 表增加一个 phone(电话)列，其中的值显然可以有多个。

(2) 关系数据模型的数据操纵

关系模型的数据操纵部分称为关系代数[①]，主要由一系列针对关系的运算构成，这些运算作用在一个或多个关系上，返回的结果还是关系，这样关系运算的输出结果可以作为另一个关系运算的输入，从而可以像算术运算一样把关系运算表达式嵌套。

关系代数是一种面向集合的语言，是关系型数据库数据操纵语言(DML)的基础，它不需要使用循环来处理关系中的每一个元组，而只需要一个命令即可。关系代数作为逻辑语言并未实现，一般作为数据操纵语言能够实现的功能的基础。

在 Codd 最初提出的关系代数理论中，包括 8 种运算，如图 1-2 所示，分别是 restrict(选择)、project(映射)、product(积，或称为 cartesian product)、intersect(交)、union(并)、difference(差)、join(连接，或称为 natural join)、divide(除)，其中 join、intersection、divide 可以由其他 5 个运算表示，所以也可以说关系代数由关系结构及针对关系的 5 种运算构成。

每种运算的作用简单解释如下。

① 选择：返回由一个关系中满足指定条件的元组构成的另一个关系。

② 映射：返回由一个关系中某些指定列中的值构成的另一个关系。

③ 积：返回一个关系，这个关系中的每一个元组由两个元组合并而成，而这两个元组分别属于另外两个关系。

④ 交：返回一个由同时属于另外两个关系中的元组构成的关系。

⑤ 并：返回一个由两个关系中的元组构成的关系，这些元组或者属于一个关系，或者同时属于两个关系。

⑥ 差：返回一个关系，这个关系中的元组属于第一个关系却不属于第二个关系。

⑦ 连接：返回一个关系，这个关系中的元组由分别属于两个关系中的两个元组合并而成，并且这两个元组在两个指定属性上有相同的值。

⑧ 除：其含义可参考图 1-2。

(3) 关系数据模型中的完整性约束

首先说明几种键的含义。

① 候选键：关系中能唯一地标识一个元组的最少的属性构成的集合称为候选键(Candidate Key)。候选键可以有多个。

② 主键：选择为唯一地标识关系中元组的候选键称为主键，一个关系中的主键只能有一个。

③ 外键：一个关系中，属性值要匹配于另一个关系(或自身)中的候选键的一个或多个属性的集合称为外键。如图 1-1 中的 emp 表的 deptno 字段可以作为其外键引用 dept 表的

---

① 之所以称为关系代数，是因为参与运算的只有关系，基本运算只有 5 种，而且运算结果还是关系，这些特点与数学中的代数学科研究的内容类似。

deptno 字段。外键的详细解释请参考 5.4 节相关内容。

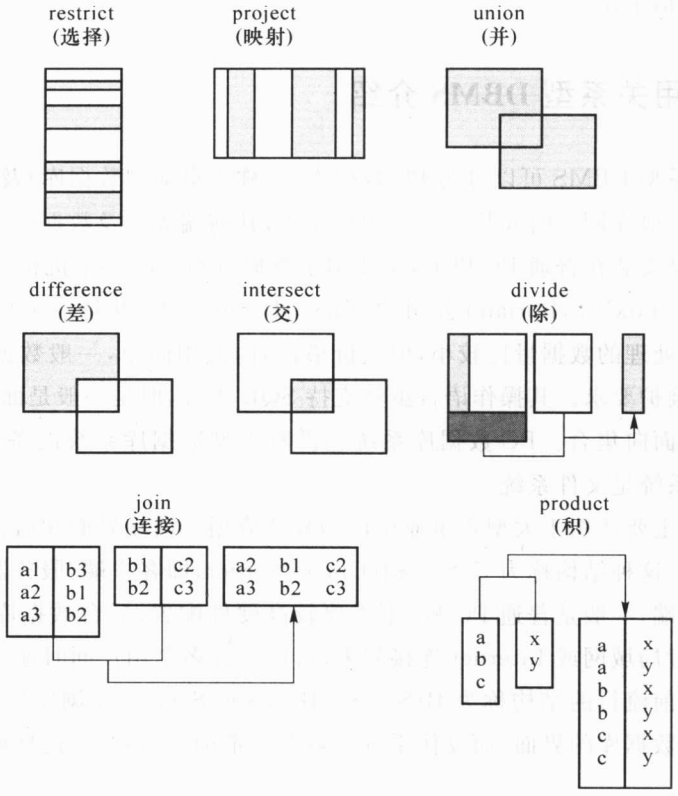

图 1-2　关系代数理论中的 8 种运算含义

完整性约束(Integrity Constraint)在一定程度上保证数据库中的数据都是正确的，如图 1-1 中 emp 表中的 sal 列表示员工工资，其值显然不能为负数，这即是一个完整性约束的例子，类似这种约束，在数据库中会有很多，也各不相同。有两种约束是所有关系型数据库都要应用的：一是实体完整性约束，二是引用完整性约束。

(1) 实体完整性约束：主键属性不能允许空值，空值也称为 null 值，指其值不确定或不存在。

(2) 引用完整性约束：外键的值或者为空，或者匹配于其引用的键值。或者说："如果 B 引用了 A，则 A 必须存在"。

4. 关系型数据库的特点

(1) 关系型数据库是建立在关系模型理论基础上的，是先有理论再有产品，并且关系数据模型是建立在严格的数学概念的基础上的，而网状模型及层次模型都是在产品出现之后总结出来的，缺乏严格的理论基础。

(2) 在关系型数据库中，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，而表的数据结构简单、清晰，这样可以得到更高的逻辑数据独立性。

(3) 关系数据库中，对数据的操纵语言是面向集合的，数据的存取路径对用户透明(即

此为试读，需要完整PDF请访问：www.ertongbook.com