数据库理论与新技术

周志逵　江涛　　编著

北京理工大学出版社

需要全本请访问 www.ertongbook.com

内 容 简 介

本书较全面和深入地介绍了数据库理论和数据库新技术。全书分为上下二篇共十章。上篇为关系数据库理论篇，由六章组成，内容包括 关系和关系模式、关系运算、数据依赖、关系模式的规范化、符号表和跟踪算法、无环数据库模式。下篇为数据库新技术篇，由四章组成，内容包括：分布式数据库、面向对象数据库、主动数据库、并行数据库、多媒体数据库、工程数据库、数据仓库等。每章后均附有习题。

本书注意取材合理，理论联系实际，尽量反映当前数据库技术的发展水平，可作为高等院校计算机专业和信息管理等专业的高年级学生和研究生的教材和参考书，也可供从事计算机研究、开发和应用的科技工作者参考。

图书在版编目(CIP)数据

数据库理论与新技术/周志逵、江涛编著. —北京：北京理工大学出版社，2001.4
ISBN 7-81045-789-6

Ⅰ.数… Ⅱ.①周… ②江… Ⅲ.①数据库系统-理论②数据库系统-新技术 Ⅳ.TP311.13

中国版本图书馆 CIP 数据核字(2001)第 03887 号

责任印制 李绍英 责任校对 郑兴玉

北京理工大学出版社出版发行
(北京市海淀区中关村南大街 5 号)
邮政编码 100081 电话(010)68912824
各地新华书店经售
北京神剑印刷厂印刷
*
787 毫米×1092 毫米 16 开本 14.75 印张 345 千字
2001 年 4 月第 1 版 2001 年 4 月第 1 次印刷
印数 1—4000 册 定价：21.00 元

※图书印装有误，可随时与我社退换※

前　言

数据库技术是计算机科学中发展最快、应用最广泛的技术之一。从60年代末开始在30年的发展历程中它已经历了第一代层次、网状数据库系统，第二代关系数据库系统，到正在研究的新一代数据库系统。30年来数据库技术的研究和应用取得了巨大的成就，数据库技术已成为当今计算机信息系统的基础和核心，成为现代计算机环境的重要组成部分。

对数据库技术的研究主要有三个方面：数据库管理系统的研究、数据库设计方法和工具的研究、数据库理论的研究。多年来在三个方面的研究都取得了丰硕的成果。在数据库理论方面，以关系数据库理论为核心形成了较为完善的数据库理论体系。近年来随着数据库应用的不断深入和扩展，原有数据库系统已不能满足新应用领域的需求，许多人致力于新系统的研究和开发，新的技术和研究成果层出不穷。

本书是作者在多年从事数据库教学和科研的基础上编写出来的，是一本面向高等院校计算机专业和信息管理等专业的高年级学生和研究生的教材和参考书，也可供从事计算机研究、开发和应用的科技工作者参考。书中较全面和深入地介绍数据库理论和新技术，重点介绍关系数据库设计理论和近年来研究的数据库新理论和新技术。

全书分为上下篇两部分共十章。上篇为关系数据库理论篇，由六章组成。第一章和第二章介绍关系和关系模式的基本概念和操作、关系运算和关系代数。第三章和第四章较详细地讲述了关系数据库的设计理论，包括数据依赖理论、关系模式的规范化理论及其算法。第五章介绍符号表和跟踪算法。第六章介绍无环数据库模式。下篇为数据库新技术篇，由四章组成。第七章分布式数据库，较详细地介绍分布式数据库系统的特点、分布透明性、分布式查询优化、事务管理和分布式目录等。第八章介绍面向对象数据库，包括数据模型、面向对象的查询、并发控制和面向对象的数据库管理系统、对象—关系数据库系统等。第九章介绍主动数据库，较详细地介绍ECA规则模型、ECA规则系统、规则语言、主动数据库管理系统等。第十章介绍进一步的数据库新技术，包括并行数据库、多媒体数据库、工程数据库、数据仓库等。数据库理论和新技术所包括的内容非常丰富，受篇幅所限，本书仅介绍了主要的关系数据库理论和部分新技术。本书在编写过程中注意取材合理，理论联系实际，尽量反映当前数据库技术的发展水平。

中国科学院研究生院的罗晓沛教授和邵佩英教授详细审阅了书稿，提出了许多宝贵的意见和建议，在此向他们表示衷心的感谢。我还要感谢我的家人，没有他们的支持、理解和帮助，这本书是不可能完成的。同时感谢在成书过程中所有支持、关心和帮助过我的人们，向他们表示深深的谢意。

由于作者水平有限，书中难免存在许多错误和不足之处，恳请读者批评指正。

周志逵　江涛

2000年12月于北京理工大学

目　录

上篇　关系数据库理论

第1章　关系和关系模式……（3）
1.1　数据模型 ……（3）
1.1.1层次数据模型 ……（3）
1.1.2网状数据模型 ……（4）
1.1.3关系数据模型 ……（4）
1.1.4面向对象数据模型 ……（5）
1.2　关系和关系模式 ……（6）
1.2.1关系 ……（6）
1.2.2关系模式 ……（7）
1.3　键 ……（7）
1.4　关系的更新 ……（8）

第2章　关系运算 ……（12）
2.1　布尔运算 ……（12）
2.2　选择 ……（14）
2.3　投影 ……（15）
2.4　连接 ……（16）
2.4.1自然连接 ……（16）
2.4.2θ_连接 ……（18）
2.5　除 ……（19）
2.6　常关系 ……（20）
2.7　属性重命名 ……（21）
2.8　关系代数 ……（22）

第3章　数据依赖 ……（25）
3.1　函数依赖 ……（25）
3.2　函数依赖公理 ……（26）
3.2.1函数依赖公理 ……（26）
3.2.2公理的完备性 ……（27）
3.2.3函数依赖集闭包及成员测试算法 ……（28）
3.3　函数依赖的等价和覆盖 ……（32）
3.3.1函数依赖的等价和覆盖 ……（32）
3.3.2无冗余覆盖 ……（32）
3.3.3规范覆盖 ……（34）
3.3.4最小覆盖 ……（36）
3.4　多值依赖 ……（40）

3 4.1 多值依赖 …… (40)
3 4 2 多值依赖的性质 …… (41)
3 4.3 多值依赖的推理公理 …… (43)
3.4 4 依赖基 …… (45)
3 4 5 嵌入多值依赖 …… (48)
3 5 连接依赖 …… (50)

第4章 关系数据库范式 …… (54)
4 1 数据库及其范式 …… (54)
4 1.1 第一范式 …… (54)
4 1.2 第二范式 …… (55)
4.1 3 第三范式 …… (56)
4 1 4 Boyce-Codd 范式(BCNF) …… (57)
4 2 关系模式的规范化 …… (57)
4 2 1 关系模式的分解 …… (58)
4 2 2 通过分解实现规范化 …… (63)
4 2.3 通过合成实现规范化 …… (64)
4 2.4 规范化关系模式为 BCNF …… (67)
4.3 第四范式和投影-连接范式 …… (69)
4.3 1 第四范式 …… (69)
4.3 2 投影-连接范式(Project-Join NF) …… (71)

第5章 符号表和追踪算法 …… (74)
5 1 符号表 …… (74)
5 2 符号表等价和模式等价 …… (75)
5.3 包含映射 …… (77)
5 4 带限制符号表的等价 …… (78)
5 4 1 带限制的符号表 …… (79)
5.4.2 F-规则和 J-规则 …… (80)
5 4 3 追踪算法 …… (81)
5 4.4 带限制集符号表的等价 …… (83)
5 5 检测隐含依赖 …… (83)
5 5.1 检测隐含的连接依赖 …… (84)
5 5.2 检测隐含的函数依赖 …… (84)
5 5 3 检测隐含的多值依赖 …… (86)
5 6 追踪算法的计算复杂度 …… (87)

第6章 无环数据库模式 …… (90)
6.1 数据库模式的超图表示 …… (90)
6 2 无 α 环数据库模式 …… (92)
6.2.1 无 α 环数据库模式的特性 …… (92)
6 2.2 Graham 算法 …… (98)

需要全本请访问 www.ertongbook.com

6.2.3 无 α 环数据库模式的设计 …… (100)
6.3 无 γ 环数据库模式 …… (104)
6.3.1 无 γ 环数据库模式的特性 …… (104)
6.3.2 无 γ 环数据库模式的判定算法 …… (108)
6.3.3 无 γ 环数据库模式的设计 …… (110)

下篇 数据库新技术

第7章 分布式数据库 …… (115)
7.1 分布式数据库系统的特点 …… (115)
7.2 分布式数据库系统的体系结构 …… (117)
7.2.1 分布式数据库系统的参考体系结构 …… (117)
7.2.2 数据分片 …… (119)
7.2.3 分布透明性 …… (120)
7.3 分布式查询处理与优化 …… (122)
7.3.1 分布式查询处理 …… (122)
7.3.2 查询优化 …… (125)
7.4 分布式事务管理 …… (131)
7.4.1 分布式事务的并发控制 …… (131)
7.4.2 分布式事务的恢复 …… (137)
7.5 分布式目录 …… (142)
7.5.1 目录的分布和管理 …… (142)
7.5.2 不同分布式数据库系统中的目录分布和管理 …… (143)

第8章 面向对象数据库 …… (145)
8.1 新应用的需求与传统数据库的局限性 …… (145)
8.2 面向对象的程序设计语言 …… (146)
8.2.1 Smalltalk 语言 …… (146)
8.2.2 C++语言 …… (148)
8.3 面向对象数据模型 …… (149)
8.3.1 对象 …… (149)
8.3.2 类 …… (149)
8.3.3 类层次和继承 …… (151)
8.3.4 对象标识 …… (152)
8.4 面向对象数据库系统的查询 …… (153)
8.5 面向对象数据库系统的并发控制 …… (155)
8.6 面向对象数据库管理系统 …… (157)
8.6.1 ORION 的数据模型 …… (157)
8.6.2 复合对象 …… (158)
8.6.3 模式进化 …… (158)
8.6.4 版本管理 …… (159)
8.6.5 对象的存储管理 …… (160)
8.7 对象-关系数据库系统 …… (161)

8.7.1 复杂对象 …… (161)
8.7.2 继承 …… (162)
8.7.3 函数 …… (164)
8.7.4 ORDBMS 中的数据查询 …… (165)

第9章 主动数据库 …… (167)
9.1 主动数据库系统概述 …… (167)
9.2 ECA 规则模型 …… (168)
9.2.1 知识模型 …… (168)
9.2.2 执行模型 …… (171)
9.3 ECA 规则系统 …… (175)
9.3.1 ECA 规则系统结构 …… (175)
9.3.2 事件检测器 …… (176)
9.3.3 条件评估 …… (178)
9.3.4 事务模型 …… (179)
9.3.5 规则的处理执行 …… (180)
9.3.6 恢复 …… (181)
9.4 规则语言 …… (181)
9.5 主动数据库管理系统 …… (184)
9.5.1 系统结构 …… (184)
9.5.2 数据模型和编程语言 …… (185)
9.5.3 规则管理 …… (186)
9.6 主动数据库管理系统 HiPAC …… (187)
9.6.1 规则和事件类 …… (187)
9.6.2 规则执行语义 …… (187)
9.6.3 举例 …… (188)
9.6.4 系统结构 …… (190)

第10章 数据库新进展 …… (193)
10.1 并行数据库 …… (93)
10.1.1 并行数据库系统的体系结构 …… (193)
10.1.2 加速比和扩展性 …… (194)
10.1.3 数据划分 …… (195)
10.1.4 并行处理技术 …… (196)
10.1.5 并行数据库的查询优化 …… (198)
10.1.6 大型数据库系统中的并行处理技术 …… (198)
10.2 多媒体数据库 …… (199)
10.2.1 多媒体数据的特点 …… (199)
10.2.2 多媒体数据库管理系统 …… (200)
10.2.3 多媒体数据库系统的体系结构 …… (200)
10.2.4 多媒体数据模型 …… (201)
10.2.5 多媒体数据的存储 …… (202)

10 2.6 多媒体数据的查询 …… (203)
10 2 7 多媒体数据库的事务处理 …… (203)
10 3 工程数据库 …… (204)
10.3 1 工程数据库的特点 …… (204)
10 3 2 工程数据库的数据模型 …… (205)
10 3 3 版本管理 …… (206)
10.3.4 长事务管理 …… (207)
10.3 5 典型工程数据库管理系统 …… (209)
10 4 数据仓库 …… (210)
10.4 1 什么是数据仓库 …… (211)
10 4 2 数据仓库中的几个重要概念 …… (212)
10 4 3 数据仓库的结构 …… (213)
10 4 4 数据仓库的数据库模式 …… (214)
10 4 5 数据仓库设计 …… (215)
10 4 6 数据仓库的前端工具 …… (218)

参考文献 …… (222)

****

* 上篇　关系数据库理论 *

****

数据库技术从60年代末至今已有30年的历史，30年来对数据库技术的研究和应用都取得了巨大的成功，使其成为计算机科学中发展最快、应用最广泛的技术之一。数据库理论是数据库技术的重要组成部分，其中关系数据库理论在数据库理论中占有主要地位。与最早出现的层次和网状数据库系统相比，关系数据库系统是数据库技术发展的主流，而具有坚实的理论基础是关系数据库系统的突出优点之一。尽管人们已提出，新一代数据库系统是以面向对象数据模型为基础的数据库系统，但在今后的若干年内关系数据库系统不会被新一代数据库系统所取代，而且会不断地扩展和完善其功能并在新的应用领域发挥作用，而传统的数据库理论和技术也会在新一代数据库系统中得到继承。

关系数据库理论的创始人是美国IBM公司的研究员E. F. Codd。1970年E. F. Codd发表了题为"A Relatıonal Model for Large Shared Databank"的论文，之后在不到10年时间里他发表了一系列有关关系数据库理论的文章，奠定了关系数据库理论的基础。在这期间，许多优秀的计算机专家如J. D. Ullman、W. W. Armstrong、D. Maır、C. Beerı、R. fagın、C. J. Date、P. A. Bernstein等也发表了大量的关系数据库论文，特别是分别由C. J. Date、J. D. Ullman和D. Maır出版的著作"数据库系统导论"、"数据库系统原理"和"关系数据库理论"，标志着关系数据库理论已走向成熟。之后，以关系数据库理论为核心形成了较为完善的数据库理论体系。

数据库是一个十分活跃的研究领域，新的技术和研究成果不断涌现，不断丰富着数据库的理论。数据库理论主要包括：关系代数和关系演算、数据依赖理论、规范化理论、查询优化理论、符号表追踪理论、表示理论、无环超图理论、泛关系理论以及空值理论等。随着数据库应用领域的不断深入和扩大，数据库技术与分布处理、并行处理、人工智能、多媒体等技术相结合出现了许多新的技术和理论，人们在不断地探索这些新技术和新理论，而其中许多研究是建立在关系数据库技术和理论基础上的。本篇将介绍关系数据库理论的主要部分：关系代数、数据依赖理论、规范化理论、符号表追踪理论及无环超图理论。

第 1 章　关系和关系模式

关系是数学上集合论中的一个概念。1970 年，E. F. Codd 发表了题为“大型共享数据库数据的关系模型”的论文，将关系的概念引入数据库，开创了数据库关系方法和关系数据理论的研究，在层次和网状数据库系统之后，形成了以关系数据模型为基础的关系数据库系统。本章介绍关系数据模型中的基本概念：关系和关系模式、关键字以及在关系上的基本操作。

为了对数据模型有一个较全面的了解，下面先介绍数据模型的基本概念及数据库系统中采用的基本数据模型。

1.1　数据模型

数据模型是数据库中描述实体和实体间联系以及有关语义约束的一种方法，是现实世界数据抽象的主要工具。数据模型是数据库系统的一个核心问题，不同类型的数据库系统其主要区别是所支持的数据模型不同。通常一个数据模型由三部分组成：数据结构、数据操作和数据的完整性约束。

1. 数据结构

数据结构用于描述数据的静态结构，包括应用所涉及的对象类和对象类所具有的特征以及他们之间的联系。如网状模型中的数据项、记录，系型，关系模型中的域、关系等。

2. 数据操作

数据操作是施加在对象上的一组操作，是对数据动态特性的描述。通常对数据的操作有检索、插入、删除和修改，支持不同数据模型的数据库系统对数据操作定义有不同的操作符和操作所遵循的规则。

3. 数据的完整性

数据的完整性约束是对数据静态和动态特性的限定，它定义相容的数据库状态的集合和所允许的状态改变。如关系模型中规定一个数据记录必须有一个确定的关键字。现实世界中的实体是按一定方式相互制约、相互依存的，完整性约束应反映数据模型中对象间的这种制约和依存关系。

数据模型给出了在计算机系统上描述和动态模拟现实数据及其变化的一种抽象方法，数据模型不同，描述和实现方法亦不相同，相应的支持软件即数据库管理系统也就不同。严格地讲，一个数据模型应由上述三部分组成，但数据结构的不同是区别数据模型最本质的部分。因此，一般所说的数据模型主要指其表示的不同数据结构。

基本的数据模型有层次、网状和关系数据模型。

1.1.1　层次数据模型

层次和网状模型是最早用于数据库系统的数据模型。层次模型的基本数据结构是层次结构，也称树型结构，树中每个结点表示一个实体类型，这些结点应满足：

(1) 有且仅有一个结点无双亲，这个结点称为根结点；

(2) 其它结点有且仅有一个双亲结点。

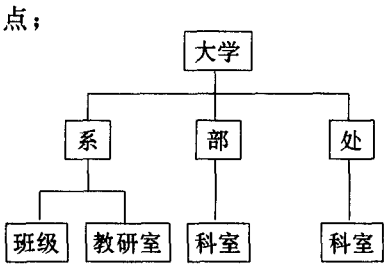


图1 1 大学组织机构的层次模型

在层次结构中，结点之间的连线表示实体间的联系。现实世界中实体间的联系有一对一、一对多和多对多三种联系。层次结构中结点间的连线表示了实体间的一对多联系，即一个父结点可以有一个或多个子结点。实现世界中许多实体间存在着自然的层次关系，如组织机构、家庭关系、物品分类等。图 1.1 所示是一个层次模型，该模型描述了一个大学的组织机构及其联系。

层次模型中的主要概念有：片段、字段、层次序列、层次路径等。

1.1.2 网状数据模型

网状模型的数据结构是一个网络结构。在网状模型中，允许：

(1) 一个结点可以有多个双亲结点；

(2) 多个结点无双亲结点。

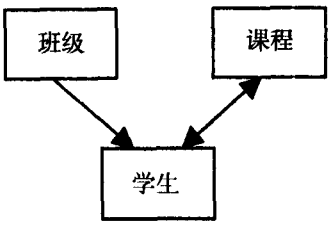


图 1.2 网状模型的例子

在网状模型中每个结点表示一个实体类型，结点间的连线表示实体间的联系。与层次模型不同，网状模型中的任意结点间都可以有联系，而且可以表示多对多的联系。因此，与层次模型相比网状模型更具有普遍性。在图 1.2 中班级、学生和课程实体间组成了一个网状数据模型，其中班级和学生间为一对多的联系而学生与课程间为多对多的联系。

网状模型中的主要概念有：记录、数据项、系、域等。

网状模型虽然可以表示实体间的复杂关系，但它与层次模型间没有本质的区别，它们都用连线表示实体间的联系，在物理实现上也有许多相同之处，如都用指针表示实体间的联系。层次模型是网状模型的特例，它们都称为格式化的数据模型。

1.1.3 关系数据模型

在关系模型中基本的数据结构是二维表，由行和列组成。一张二维表称为一个关系。在关系模型中，实体和实体间的联系都是用关系表示的。在二维表中存放了两类数据：

(1) 实体本身的数据；

(2) 实体间的联系。

如图 1.3 是一个表示学生和教师任课情况的关系模型，图 1.3 (a) 和 (b) 分别表示学生关系和教师任课关系。这两个关系也表示了学生和任课教师间的多对多联系，他们之间的联系是由在两个关系中的同名属性“班级”表示的。

关系模型中的主要概念有：关系、属性、元组、域、关键字等。

与层次和网状模型相比，关系模型有下列优点：

(1) 数据结构单一，不管实体还是实体间的联系都用关系来表示；

(2) 建立在严格的数学概念基础上，具有坚实的理论基础；

(3) 将数据定义和数据操纵统一在一种语言中，使用方便，易学易用。

需要全本请访问 www.ertongbook.com

学号	姓名	班级
909901	王鸣	90991
909902	李丽	90992
909801	刘敏	90981
909802	陆川	90982
909803	辛力	90981

(a)

职工姓名	系别	课程	班级
吴云峰	数　学	离散数学	90991
赵　伟	外　语	英　　语	90992
马小路	计算机	数据结构	90981
曹跃岩	计算机	数 据 库	90981

(b)

图 1.3 学生和教师任课情况的关系模型

由于关系数据模型具有许多优点，因而在 80 年代之后的商品化数据库系统几乎都是关系型的。当然，关系模型也有不足之处，主要是其结构单一，缺乏语义信息，不能更好地模拟现实世界中的复杂对象，因而在新的应用领域受到限制。

1.1.4 面向对象数据模型

面向对象数据模型中的基本数据结构是对象，一个对象由一组属性和一组方法组成，属性用来描述对象的特征，方法是描述对象的操作。一个对象的属性可以是另一个对象，另一个对象的属性还可以用其它对象描述，以此来模拟现实世界中的复杂实体。

图 1.4 所示是一个对购车管理对象的描述。购车管理对象为一个复杂对象，其描述包括属性和方法两个部分，其中属性纳税情况和违章情况也都是对象，分别有自己的属性和方法。

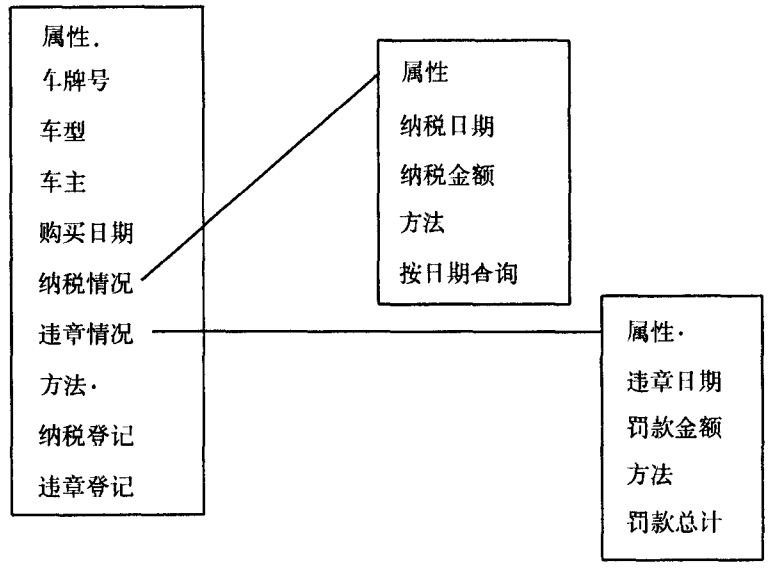


图 1 4　购车对象的数据结构

在面向对象的数据模型中对象是封装的，对对象的操作通过调用其方法来实现。面向对象数据模型中的主要概念有：对象、类、方法、消息、封装、继承、多态等。

面向对象的数据模型有许多优点，主要有：

(1) 可以表示复杂对象，精确模拟现实世界中的实体；

(2) 模块化的结构，便于管理和维护；

(3) 具有定义抽象数据类型的能力。

面向对象的数据模型是新一代数据库系统的基础，是数据库技术发展的方向。本书第八章将较详细地介绍面向对象的数据模型和面向对象数据库技术。

下面介绍关系模型中的主要概念及其形式化描述。

1.2 关系和关系模式

1.2.1 关系

在关系模型中惟一的数据结构是关系，一个关系对应一张二维表。下面在给出关系的定义之前，先定义笛卡尔积（Cartesian Product）。

定义 1 给定一组集合 D_1，D_2，…，D_n，它们可以是相同的。定义 D_1，D_2，…，D_n 的笛卡尔积为：

$D_1 \times D_2 \times \ldots \times D_n = \{(d_1,d_2,\cdots,d_n) | d_i \in D_i, i = 1,2,\cdots n\}$。

其中每一个元素$(d_1,d_2,\cdots,d_n)$叫做一个 n 元组(n-tuple)，元素中的每一个值 d_i 叫做第 i 个分量。

例如：设 $D_1=\{1,2,3\}$，$D_2=\{a,b\}$则

$D_1\times D_2=\{(1,a),(1,b),(2,a),(2,b),(3,a),(3,b)\}$

定义 2 $D_1\times D_2\times\cdots\times D_n$ 的任一个子集称 D_1，D_2，…，D_n 上的一个关系(Relation)。集合 D_1，D_2，…，D_n 是关系中元组的取值范围，称关系的域(Domain)，n 叫做关系的度(Degree)。

度为 n 的关系称 n 元关系。如 $n=1$ 的关系称一元关系，$n=2$ 的关系称二元关系。

关系中的列称为属性，每一列用属性名表示。我们用符号 dom(A_i)表示属性 A_i 的域，这些域是有限的非空集合。

下表 1 是一组火车时刻表。表中的每一行是一个五元组，给出了每列火车的车次、始发站、终到站、开车时间及到达时间。

表 1 火车时刻表

NUMBER	*FROM*	*TO*	*DEPARTS*	*ARRIVES*
565	BeiJing	XuZhou	20：40	7：54
523	XuZhou	LuoYang	21：30	6：06
532	LuoYang	BeiJing	21：45	9：40
K95	WuChang	ShenZhen	16：55	7：18
K96	ShenZhen	WuChang	17：13	7：37

表中每一属性名对应的域为：

dom（*NUMBER*）＝{565,523,532,K95,K96}

dom(*FROM*)＝dom(*TO*)＝{BeiJing,XuZhou,LuoYang,WuChang,ShenZhen}

dom(*DEPARTS*)＝dom(*ARRIVES*)＝一组时间。

从表 1 可以看出，表中没有相重的行，列 *FROM* 和 *TO* 及列 *DEPARTS* 和 *ARRIVES* 分别具有相同的域，将时刻表中列的次序互换不影响表的内容。因此，数据库中的关系虽然是

一个二维表，但具有某些特定的性质。

数据库中的关系具有如下的性质：

(1) 每一列中的值是同类型的数据，来自同一个域；

(2) 不同的列可以有相同的域，每一列称为属性，用属性名标识；

(3) 列的次序是无关紧要的，

(4) 关系中的每个分量是不可分的数据项；

(5) 元组的次序是无关紧要的；

(6) 关系中的各个元组是不同的，即不允许有重复的元组。

1.2.2 关系模式

从关系的定义得出关系是元组的集合。但在关系数据库中，关系表示为两个部分，一部分是对关系的描述，称为关系模式(Relation Shema)，另一部分是关系中元组的集合，称为关系。关系模式又称为关系的内涵(intension)，而关系中元组的集合又称为关系的外延(extension)。

定义 3 关系模式可形式化地表示为一个三元组 $R(U,D)$，其中 R 是关系名，U 是组成关系 R 的属性名集合，D 是属性集 U 中每个属性所对应的域。

关系模式一般表示为 $R(A_1,A_2,\cdots,A_n)$。模式 R 上的一个关系 r 是从 R 到 D 的映像 t 的有限集，通常写为 $r(R)$。元组 t 的各个分量用 $t[A_i]$表示，$t[A_i]\in D_i(1\leqslant i\leqslant n)$，即 $t[A_i]$的值是受限于 D_i 的。

如表 1 的关系可表示为 $TRIAN(NUMBER,FROM,TO,DEPARTS,ARRIVES)$。该关系有五个元组，第一个元组 t_1 的分量为：$t_1[NUMBER]=565$，$t_1[FROM]=$BeiJing，$t_1[TO]=$XuZhou，$t_1[DEPARTS]=12:40$，$t_1[ARRIVES]=7:54$。

定义 4 设属性集 U 和 U 的属性所关联的域 D，U 上的关系数据库模式 $\boldsymbol{R}$ 是关系模式 R_1，R_2，…，R_p 的集合，且 $R_1\cup R_2\cup\cdots\cup R_p=U$。一个关系数据库模式 $\boldsymbol{R}$ 对应的所有关系的集合$\{r_1,r_2,\cdots,r_p\}$称关系数据库模式 $\boldsymbol{R}$ 上的一个关系数据库 d，其中，r_i 是对应 R_i 上的一个关系，$1\leqslant i\leqslant p$。

在关系数据库中，关系模式一经确定是不会随意改变的，是稳定的。但关系中的元组将随着时间的改变是动态变化的，即关系中的元组可以增加、删除和修改。如表 1 中若增开一列火车，则在 $TRAIN$ 关系中将增加一个元组，而停开一列火车，将删除一个元组。若某列火车的运行时间有变化，则需修改相应元组的 $DEPARTS$ 和 $ARRIVES$ 的值。所以，某一时刻数据库中的关系处在某一种状态，不同时刻数据库的状态是不同的。

从以上叙述看出，关系模式和关系分别表示关系的结构和关系的内容，但常常把关系模式和关系统称为关系，一般用大写字母表示关系模式，用小写字母表示关系。

1.3 键

键(Key)是关系数据库中的一个重要概念，它能惟一地标识一个元组，使不同元组表示客观世界中的不同个体。

关系模式 R 上的键 K 是 R 的子集，$K=\{B_1,B_2,\cdots,B_m\}$。对 $r(R)$中任意两个不同的元组

t_1、t_2 满足 $t_1[K]\neq t_2[K]$，即任意两个不同的元组其 K 的值是不同的。这就意味着要惟一标识一个元组，只要知道其 K 的值就足够了。下面给键下一个确切的定义。

定义 5 设关系模式 $R(U)$，$K\subseteq U$，r 是 R 上的任一关系，若对 r 中的任意两个不同的元组满足.

(1) $t_1[K]\neq t_2(K)$，

(2) 若 $K'\subset K$ 但 $t_1[K']\neq t_2[K']$不成立，

则称 K 是 R 的键。若条件(2)不成立，称 K 是 R 的超键(superkey)。

由键的定义得出，键是能惟一标识元组的最小属性集。如关系 *TRAIN* 中 *NUMBER* 是一个键，而属性集 *NUMBER*、*FROM* 则是一个超键。我们在关系模式中用带下划线的属性表示关系的键，即 *TRAIN*(*NUMBER*, *FROM*, *TO*, *DEPARTS*, *ARRIVES*)表示关系 *TRAIN* 的键是 *NUMBER*。有的关系具有多于一个的键。如关系 *TRAIN* 中，属性集 *FROM*、*TO*、*DEPARTS* 也是一个键，因为一般不会从同一始发站同一时间发两列火车到同一终到站。在这种情况下，我们指派其中的一个键为主键(primary key)，简称为关系的键，用带下划线的属性表示，其余的未被指派的键称隐含键(implicit key)，而主键和隐含的键统称为候选键(candidate key)。

由于键能惟一标识一个元组，因此在元组中作为键的属性值不能是空值。

1.4 关系的更新

在前面提到，数据库的状态是随时间变化的，经常要对关系中的元组进行插入、删除、修改操作，这些操作统称为关系的更新。下面形式化地描述这些操作。

1. 插入

如果要在关系中增加元组，可以用插入操作。

对关系 $r(A_1, A_2, \cdots, A_n)$，插入操作形式为：

ADD(r; $A_1=d_1, A_2=d_2, \cdots, A_n=d_n$)。

若按关系模式中给定的属性名序，插入操作可简化为：

ADD(r; $d_1, d_2, \cdots, d_n$)。

例如，要增开一趟从北京到天津的 Y15 次列车，可对表 1 中的关系 *TRAIN* 执行以下插入操作：

ADD (*train*, *NUMBER*=Y15, *FROM*=BeiJing, *TO*=TianJin,

DEPARTS=10.05, *ARRIVES*=12:43)

以上操作可简化为：

ADD(*train*; Y15, BeiJing, TianJin, 10:05, 12 : 43)

插入操作要加一个元组到一个指定的关系中必须符合对该关系的定义。若出现下列情况之一，插入操作将被拒绝，即操作无效。

(1) 描述的元组不符合所指定的关系模式；

(2) 元组的某些值不属于对应的域；

(3) 元组的键已在关系中存在。

例如，以下插入操作是不允许的。因第一个插入操作中 *DATE* 不是关系 *TRAIN* 的属性，

而第二个插入操作中，键 *NUMBER* 为 565 次列车的信息已在表 1 中了，且属性 *ARRIVES* 的值 25：45 不在时间域中。因此，以下操作将被拒绝。

ADD(*train*;*NUMBER*=117,*FROM*=JiNan,*TO*=TaiYuan,*DATE*=6 March)

ADD (*train*;*NUMBER*=565,*FROM*=BeiJing,*TO*=XuZhou,

DEPARTS=18.50,*ARRIVES*=25：45)

2 删除

删除操作是从关系中撤消元组。对关系 $r(A_1,A_2,\cdots,A_n)$，删除操作形式为：

DEL (r; $A_1=d_1$, $A_2=d_2$, …, $A_n=d_n$)

当按关系模式给出的属性名序时，删除操作可简化为：

DEL($r;d_1,d_2,\cdots,d_n$)

例如，对 *train* 关系作以下操作，*NUMBER* 为 523 的一行元组将被删除。

DEL(*train*;*NUMBER*=523,*FROM*=XuZhou,*TO*=LuoYang,

DEPARTS=21：30,*ARRIVES*=06：06)

以上操作可简化为：

DEL(*train*;523,XuZhou,LuoYang,21:30,06：06)

实际上，为了识别被删除的元组并不需要所有元组的信息，只需要指定键的值就足够了。如果 $K=\{B_1,B_2,\cdots B_m\}$是键，那么，删除操作可简化为下面的形式

DEL($r;B_1=e_1,B_2=e_2,\cdots,B_n=e_m$)

如关系 *train* 的键是 *NUMBER*，简化为. DEL(*train*;*NUMBER*=523)。甚至还可以简化为：DEL(*train*;523)。

删除操作的结果将使被指定删除的元组从关系中消除。如果被删除元组在关系中不存在，这个关系将保持不变，但需给出一个出错提示。

在关系中删去最后一个元组是不受限制的，即空关系是允许的。

3. 修改

除了插入和删除一个完整的元组外，我们还可以只修改一个元组的一部分值。修改元组用修改操作来完成。如上述关系 $r(A_1,A_2,\cdots,A_n)$，有$\{C_1,C_2,\cdots,C_p\}\subseteq\{A_1,A_2,\cdots,A_n\}$，则修改操作形式为：

CH($r;A_1=d_1,A_2=d_2,\cdots,A_n=d_n;C_1=e_1,C_2=e_2,\cdots,C_p=e_p$)

修改操作把元组$\{d_1,d_2,\cdots,d_n\}$中$\{c_1,c_2,\cdots,c_p\}$的值改为$\{e_1,e_2,\cdots,e_p\}$。如果 $K=\{B_1,B_2,\cdots B_m\}$为键，那么可简化为：

CH($r,B_1=d_1,B_2=d_2,\cdots,B_m=d_m;C_1=e_1,C_2=e_2,\cdots,C_p=e_p$)

例如，要将 523 次列车的发车时间改为 22：40，需要对表 1 中的关系 *train* 作以下操作。

CH(*train*,*NUMBER*=523;*DEPARTS*=22：40,*ARRIVES*=07：16)

修改操作实际上可用删除操作后跟一个插入操作实现。对插入和删除操作的限制可运用到修改操作中。如指定的元组不在关系中；修改后的值不在规定的域中；被修改元组的键和已在关系中元组的键值相同，这些情况都应使修改操作出错。

例 1.4.1 施行下列操作于表 1，则操作结果如表 2 所示。

1. ADD(*train*;33,TianJin,ShangHai,17：20,10：36),

2. ADD(*train*;Y15,BeiJing,TianJin,10：05,12：43);

3. DEL(*train*;523);

4. CH(*train*;*NUMBER*=532;*DEPARTS*=22：45,*ARRIVES*=10：42)。

表 2　更新后的火车时刻表

NUMBER	*FROM*	*TO*	*DEPARTS*	*ARRIVES*
33	TianJin	ShangHai	17·20	10　36
Y15	BeiJing	TianJin	10·05	12：43
565	BeiJing	XuZhou	20·40	7：54
532	LuoYang	BeiJing	22·45	10：42
K95	WuChang	henZhen	16·55	7：18
K96	ShenZhen	WuZhang	17：13	7：37

习　题

1. 设关系 R（货号，品名，规格型号，重量，单价，数量，生产单位，入库日期）表示一仓库入库单，其数据如下表所示。

仓库入货单

货　号	品　名	规格型号	重　量	单　价	数　量	生产单位	入库日期
FS－40N	电风扇	400mm	20kg	450 00	150	吴县电机厂	1998 2 16
DV－330	影碟机	VC－N30	13 5kg	1145 00	200	厦门电子公司	1998 4.21
YU－110	彩　电	29E8DX	38 5kg	5956.00	80	大连电视机厂	1998 5 28
ASL－17	微波炉	WD800	16.5kg	1439 00	500	顺德电器厂	1998 8.5
NV－900	录放机	G－30	11 5kg	2398 00	50	松下电器公司	1998 9 25

按要求完成以下操作：

（1）入库单中少了一项，数据为：TR－668、彩电、34N2PD、43kg、9245、60、长虹电视机厂、1998.11 21，请把该数据加入入库单中。

（2）将货号为 ASL－17，1998 年 8 月 5 日入库的微波炉的单价改为 1 200 元。

2. 试给出 1 题中入库单关系 R 的键。

3. 设关系模式 $R=\{FIGHT-NUMBER,DATE,GATE,TIME,DESTINATION\}$。$r(R)$中一个元组为$\{d_1,d_2,d_3,d_4,d_5\}$，其含义为："在日期 d_2 的时刻 d_4，航班 d_1 的飞机从 d_3 门出发，到达目的地 d_5"。试求 R 的键。

4. 回答下列问题：

（1）两个键的并集是否为一个键？

（2）两个超键的交集是否必定成为一个键？

5. 已知关系模式 $R(A_1,A_2,\cdots,A_n)$，R 最多能够有多少种键？R 的超键最多能有多少？

6. 如果键 K 的值为空，应如何解释？

7. 已知 $K=\{b_1,b_2,\cdots,b_m\}$为关系模式 $R(A_1,A_2,\cdots,A_n)$的一个键，r 是 R 上的一个关系，$\langle d_1,$

需要全本请访问 www.ertongbook.com