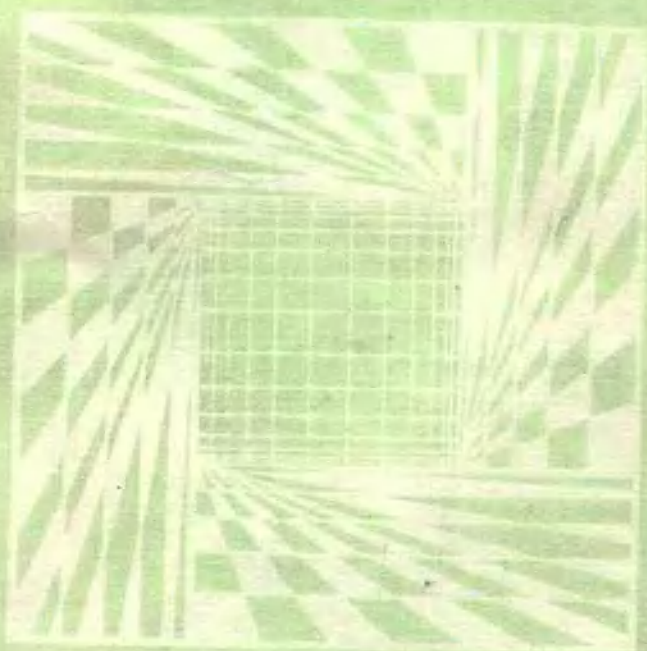


关系数据库技术

[美] 萨德·阿拉基克 著 钱士湘 李昭原 姚晓春 译



北京航空航天大学出版社

2.3

1/1

712 1112-3
AL7/1

关系数据库技术

[美] 萨德·阿拉基克 原著

钱士湘
李昭原 译
姚晓春



北京航空航天大学出版社

内 容 提 要

本书综合介绍了关系数据库技术的原理、方法和技术。主要包括：关系模型的应用环境的模型化及其形式化描述；关系语言；关系数据库逻辑设计、物理设计的理论及其实现技术；以及对多用户和分布式数据库系统的一些重要问题（尤其是与数据完整性有关的问题）进行了系统的阐述。

本书采用了一体化的描述方法。以理论严谨，选材新颖，例子实用，问题丰富，可作为高等院校计算机专业本科生或研究生的数据库课程的教学参考书，对从事计算机工作的科研人员、工程技术人员及其它有关人员也有一定参考价值。

JS20/37

13

关系数据库技术

GUAN XI SHUJUKU JISHU

【英】卢德·阿拉基克 原著

钱士衡 李昭康 姚晓春 译

袁田编辑 樊 景

北京航空航天大学出版社出版

新华书店总店科技发行所发行 各地新华书店经售

北京德云书都印刷厂印装

850×1168 1/32 印张：9.6 字数：255千字

1989年12月第一版 1989年12月第一次印刷 印数：3100册

ISBN 7-81012-140-5/TP·021 定价：2.20元

序 言

本书综合介绍了当今适用于信息系统设计和实现的最重要的原理、工具和技术。为此采用的是关系数据模型的体系以及它在多用户和分布式环境下的应用和实现。主要内容包扣：关系模型的应用环境的概念模型化、该模型性质的形式化描述、关系语言、关系数据库系统的逻辑设计和物理设计及其实现的技术。本书试图运用一种一体化的方法，来展示在关系方法基础上的所有这些问题，以及与这种方法有关的各种研究和实际开发。

这是唯一的一本将关系数据库的原理、工具和技术融于一体的书籍。数据模型的建立、逻辑和物理的数据库设计、数据库应用程序设计、数据库系统的应用和实现等方面存在着多种多样的方法，从而要求我们建立一种通用的体系。近些年来，数据库领域的发展突飞猛进，产生了多种多样的结果，因此必须有一种一致的方法来对待这些结果，否则学习现代数据库技术会困难重重，更不用说讲授这方面的课程了。

很显然，对大多数对关系数据库技术有兴趣的读者来说，应用技术更加重要。同时，有关数据库的基本结构和算法、并发控制协议、以及分布式处理技术的知识，正成为普遍的专业知识。不仅系统开发人员需要了解它们，而且所有那些想了解关系数据库系统实际是怎样工作的读者（尤其是数据库应用程序员）也应该了解它们。另一方面，对于系统开发人员，讲述一下概念和关系模型化、终端用户的需求以及工具，也会是有用的。再者，本书把关系语言、多用户和分布式系统的最重要的实现技术纳入一个体系之内，从而扩充了关系数据模型的种种发展结果所提供的形式化方法。

虽然本书主要根据一些研究论文而写成，但却是作为一本教科书来写的。这是指内容安排的方式，以及书中所用的大量实例和范围广阔的习题。读者除了对现代数学、计算机系统以及它们的一般应用的基本概念要有所了解以外，不再需要更多的背景知识。在介绍关系数据模型的基本理论、内部的关系结构和算法、事务处理模型以及分布式系统时，有些内容是形式化和技术性的，但总是以非常实际的例子加以说明。

除了以上所谈到的一体化目标外，本书还有一个与现有书籍不同的地方，即提供了当前关系模型的概念模型设计之面向“对象-操作”的技术；并对多用户和分布式数据库系统中最重要的问题（尤其是那些与数据完整性有关的问题）进行了深刻的阐述。

作者在此鸣谢BIZ nauke SR Bosne Hercegovine 所提供的研究成果，并对Mario Gaon的评论表示感谢。

萨德-阿拉基克
科学城，日本
一九八五年三月

译 者 的 话

当电子计算机的应用已进入社会各个领域乃至人们的家庭生活中时,数据库系统的重要性就更显得突出,而关系数据库尤为其中的佼佼者。

专门介绍关系数据库的书还不多,而可选为教材者更是鲜见。

本书全面而又系统地介绍了关系数据库技术的概念、工具和技术,理论严谨,选材新颖,例子实用,习题丰富,可以选作高等院校计算机专业本科生或研究生的教材或参考书。基于此,我们翻译了这本书。

序论1、2节、第一章、第五章由钱士湘教授翻译。序论3、4节,第二章、第三章由李昭原副教授翻译。前言、第四章及第五章习题由姚晓春硕士翻译。全文由钱士湘教授作了校对。李昭原副教授参加了部分校对工作。

感谢北航出版社为出这本书所做的努力。

目 录

绪 论

1. 应用环境的模型化 (1)
2. 关系模型 (5)
3. 关系数据库系统 (12)
4. 关系技术 (19)

第一章 数据模型

1. 数据的关系模型 (22)
 - 1.1 实体的关系表示 (22)
 - 1.2 关系代数 (27)
 - 1.3 关系查询语言 (32)
2. 逻辑依赖 (40)
 - 2.1 函数依赖 (40)
 - 2.2 多值依赖 (47)
 - 2.3 连接依赖 (53)
3. 层次及网状模型 (55)
 - 3.1 非规范化关系模型 (55)
 - 3.2 网状模型 (59)

练习: GROUP BY子句; SET函数; 实体联系图;

广义连接; M:N联系; 无损连接 (64)

参考资料注释 (72)

第二章 逻辑设计

1. 范式 (74)
 - 1.1 第二范式 (2NF) (74)
 - 1.2 第三范式 (3NF) (79)

1.3	Boyce-Codd范式 (BCNF)	(84)
1.4	第四范式 (4NF)	(87)
1.5	投影/连接范式 (PJNF)	(93)
2.	抽象	(97)
2.1	非规范化关系模型	(97)
2.2	聚集	(100)
2.3	归纳	(105)
3.	设计方法	(110)
3.1	扩展的关系模型	(110)
3.2	关系数据库程序的环境	(113)
3.3	建立概念模型	(129)
练习:	视图; 数据库实体的型; 归纳; 联系; 聚集; 特征; 冗余聚集; 分类设计方法; 例外模型; 例外处理	(143)
	参考资料注释	(156)
第三章 结构设计		
1.	关系映像	(157)
2.	一元查询的分解	(161)
3.	二元查询的分解	(168)
4.	二元查询的优化	(174)
5.	集合算符的查询分解	(177)
6.	关系及其映像的关系表示	(182)
7.	数据操作语句的分解	(184)
8.	映像的结构	(186)
练习:	链; 网状结构; n 元查询的分解; 查询表达式的 优化; 关系运算符的特性; B*-树	(194)
	参考资料注释	(206)
第四章 数据完整性		
1.	数据的事务和完整性	(207)

2. 事务的并发执行	(205)
3. 封锁协议	(214)
4. 逻辑锁	(225)
6. 恢复某种一致的数据库状态	(237)
练习: 完整性断言; 事务; 触发器; 树协议; 层次封 锁协议	(247)
参考资料注释	(253)
第五章 分布技术	
1. 数据库系统的体系结构	(254)
2. 分布式执行和完整性	(259)
3. 分布式查询处理	(266)
4. 分布式更新	(277)
练习: 分段; 事务结构; 完整性约束和数据分布; 归 纳和分段; 多数据库系统; 目录管理; 对象命 名	(281)
参考资料注释	(292)
参考书录	(293)

绪 论

1 应用环境的模型化

关系数据库技术的各种概念、技巧和工具，可以适用于各种各样的环境。在这种应用环境中，为了尽可能好地满足不同用户对信息的种种需求，引入了关系技术。

为满足用户需要，关系技术要求我们在设计应用环境的模型时，注意掌握那些与满足用户信息需求有关的、该环境的静态与动态特性。此模型不仅要符合设计阶段规定的要求，还必须适应未来的需要。

我们把应用环境看成是有内部联系的对象，和与这些对象有关的活动（或事件）的集合。因而，可以用如下的方式表示：

(i) 采用同样的方法来说明有关的对象、对象的性质以及对象间的关系。

(ii) 采用同样的方法来说明作用在对象和它们之间关系上的操作以及存在于应用环境中的活动或事件。这些操作会引起环境的改变并影响所表示的对象及其之间关系的性质。

(iii) 不适于以对象和操作来表示的应用环境的静态与动态特性，可以把它们说成是施加于对象上的约束或完整性规则。约束往往和应用环境的静态与动态特性有关，因此也表示了该环境的某种特性。

在应用环境中，常提出（启动）能产生所需输出的查询来满足用户对信息的需求。我们规定的能满足用户需求所必需的查询集合，是一种精确说明这些需求的方法。关系技术所提供的方法，

要求把有关对象及它们属性的说明全置于那些查询之中。

为举例说明本书介绍的概念、工具和技术，我们在结论中选择一个典型的大学应用环境。假设在分析了该应用环境的活动和信息需求之后，得到了下列有关对象、它们的属性及所涉及的活动：

对象，STUDENT, COURSE, LECTURER

性质：（如图0-1-1）

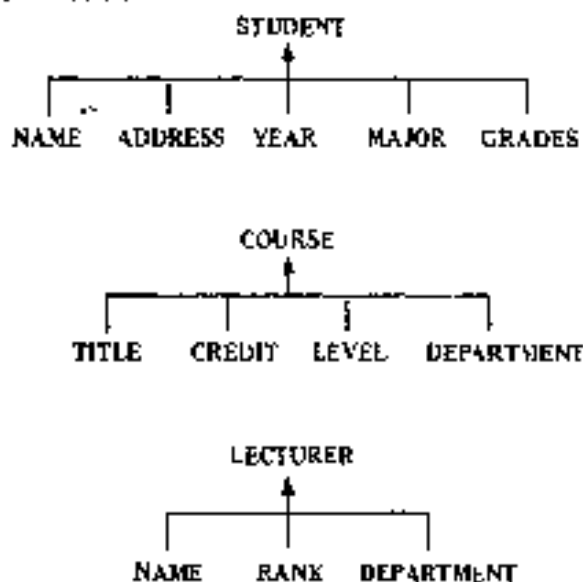


图 0-1-1

涉及特定对象的活动：

招收学生；淘汰学生；改变年级；改变专业。

招聘讲课者；辞退讲课者；改变职称；改变年级（系）。

开设课程；取消课程；改变学分；改变层次；改变部门（系）。

涉及特殊对象的查询举例：

打印某生的全部成绩。

打印计算机科学所有研究生课程的名称。

打印所有计算机科学二年级学生的名单。

打印所有教授名单。

与上述对象有关的完整性约束为：

两门不同的课不得重名。

不能解聘教授。

讲课者的职称，若有改变则只准升不准降。

某个学生所在的年级改变时，新年级应高于原来的年级。

当研究上述应用环境中用户的信息需求时很容易发现，许多查询涉及一个以上对象。例如：

打印某个学生选修的所有课程。

打印选修某门课的所有学生。

打印由某个讲课者所开的所有课。

打印同开一门课的所有讲课者。

在所选择的应用环境中会发现活动也是同样，某些活动涉及一个以上对象。换言之，它们涉及应用环境中诸对象间的联系。类似活动有：

某门课由某生选修。

从某门课中淘汰某生。

为某门课聘请一个讲课者。

改变某门课的讲课者。

应用环境中用户查询所涉及的对象间联系及它们的活动可以被看成是抽象对象。上述的查询及活动确实涉及下列对象，这些对象表达了已被定义的对对象间的联系。

联系：（如图0-1-2）

与上述联系有牵连的完整性约束为：

四年级以下的学生不得选修研究生课。

不得重选已经选过的课程。

若某门课已确定，则其课程、教室、日期、时间必须确定，

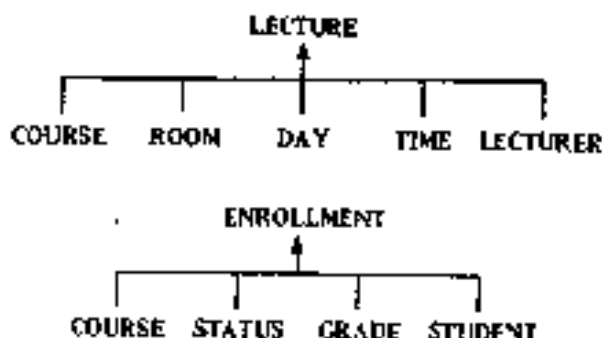


图 0-1-2

而讲师者可随后确定。

低于助理教授以下的讲师者，不得开设研究生课程。

两门课不能在同一教室中同时开设。

任意复杂程度的应用环境所必须满足的不同信息需求，可以按该环境的各个特定用户组方便地划分成不同的视图。视图是应用环境的某种抽象，它适用于共享同一视图的用户组。一般来说，视图可以用对整个应用环境所规定的对象、操作和约束的抽象来加以描绘。

数据模型是某些概念的集合，它可以按照以上惯例对某个应用环境和它的不同视图进行模型化。对应用环境中的对象及它们之间关系的说明叫模式。在某一特定视图中可以见到的对象及其间关系的说明叫子模式。模式中也包含约束以完善数据模型说明中的静态部分。

某个应用环境模型的运行部分包括简单和复合的操作集合。这些操作在特定的约束下施加于所表示的对象之上，以反映出在应用环境中的变化（事件），和（或）产生出满足用户对信息需求的结果（输出，报告）。虽然在设计模式时必须定义一批作用于该模式中所表示的对象上的需预先定义的操作，但是数据模型仍应允许对那些在设计时未能预料的操作和查询加以说明，以满足用户的要求。

2 关系模型

为了开发某个应用环境模型，有必要对其加以抽象，以适应于该应用环境的所有用户之共同视图。数据关系模型所需要的基本抽象形式是把某个应用环境中的对象划分成各种对象类型。当一组对象都具有与用户信息需求有关的相同特性集合时（与那些需求无关的特性则可以不同），我们就可认为它们同属一种类型，某种特性确定了某个对象单独的静态特性，此特性不独立于某个对象而存在。一种对象类型是由一些属性的集合来描述的，这些属性对该类型的所有对象是相同的。

数据的关系模型用同样的方法来表示对象以及对象之间的联系。这种语义相关性的基点是联系亦有其自身的特性，分类抽象亦能适用于关系，因而联系也是一种抽象对象。所以，在本书中用同样的术语来表达对象及它们间的联系，也就是所谓实体。

在数据的关系模型中用该属性值的某个元组来表示一个实体。例如，(Brown, Associate, Mathematics)是代表某个特定的讲课者的元组，它是实体型LECTURER的一个实例。

同一类型的实体集合用诸元组的集合来表示，其中不同的元组表示不同的实体。在形式化的意义上，这样的元组集合就是一个关系。例如LECTURER表明了具有属性NAME、RANK和DEPARTMENT的对象的集合。实体型LECTURER由上述三个属性的各个元组（在此例中为三元组）的值的集合来表示。

关系可以非形式化地用表来表示，表的列相当于属性而表的行相当于元组，二者为关系的元素。例如，关系LECTURER在某个时刻就相当于下列这张表：

LECTURER	NAME	RANK	DEPARTMENT
	Smith	Assistant	Physics
	Brown	Associate	Mathematics
	Jones	Associate	Physics
	Taylor	Full	Statistics
	Sellers	Lecturer	English

在关系模式中用LECTURER(NAME, RANK, DEPARTMENT)这种表示方法来描述关系(表) LECTURER的结构。

数据的关系模型为其实体表示所提供的操作和查询, 可以解释成在表上的操作。更重要的是, 在关系模型中的查询和操作涉及某些元组的集合, 这些元组能满足用元组的属性值所表示的特定的逻辑条件。除了其它优点以外, 仅此一条就可允许我们在关系构架中用与规定查询和更新操作相类似的方法, 来声明模型的第三部分——完整性约束。

作用在实体型LECTURER上的面向集合(关系)的查询之例为:

按字母顺序打印物理系所有副教授的姓名。

上述复杂操作得出的是一张姓名的分类表(关系)。换言之, 该表的唯一属性是NAME。有关讲课者的这一特性的查询用下列逻辑条件来表示:

DEPARTMENT = 物理 AND RANK = 副教授

所以在关系LECTURER的诸元组的集合中满足上述条件的属性值NAME的集合就是查询的结果。

面向集合的操作之例是:

提升所有的副教授。

此操作作用于满足下列条件的关系LECTURER的所有元组上。

RANK = 副教授

所有这样的元组的RANK属性值被改为正教授, 这意味着我们

得到了一个相应于实体型LECTURER的不同的元组集合。

用同样的风格来表示完整性约束的一个例子为：

所有数学系的讲课者的职称必须高于副教授。

在某个应用环境中根据用户对信息需求所构造的关系模式连同相应的子模式、约束、操作和查询被称为该环境的关系模型。关系数据库是关系（具有相当规模的表）的集合，其结构在模式中加以说明。各个元组的实际集合随时间而变化是因为该应用环境中被该用户激活了的不同操作，是随时间而变化的。然而，只有按照完整性约束所做的有关关系数据库的变化（更新）才能被接受。用户通过以下两种方法来满足他们对信息的需求。第一种是启动那些表示复合查询的预先规定了的事务，从而产生相应的报告；另一种是利用用户友好的关系查询语言来表达特定的查询。

数据的全局性关系模型具有概念上简单这一基本优点（它是多张表的集合），适用于应用环境中的不同用户（临时用户、程序员等等）。这些用户可以在他们所充分了解的统一框架的基础上相互之间进行通讯。数据的关系模型还配备有关系数据定义、查询和数据操作（更新）语言。这些也都适用于具有不同技术知识的用户。这些用户有的只懂得极少的基本概念（表和表操作），有的则对现代程序语言的程序设计具有详尽的知识。

数据的关系模型由于在概念上的简单性，有时不足以靠它自身来开发一个充分的、高度面向应用环境且能复盖该环境的大量语义的数据模型，因而在设计关系模式和在此基础上确定不同的视图时引用了大量的抽象技术。

被称为聚合的抽象形式已经在我们的例子中作为两种重要的情况被引用过。首先，每个对象，STUDENT, COURSE 和 LECTURER, 可以被看成具各自属性的聚合。接着，抽象对象——LECTURE和ENROLLMENT——被定义为其它简单的（诸属性）或复合的对象的聚合。事实上，LECTURE被定义

为复合对象COURSE和LECTURER以及简单对象（属性）ROOM、DAY和TIME的聚合。这意味着我们使用聚合把简单或复合对象之间的联系定义成高层次的抽象对象。下面的标志被建议用来表达那种情况（如图0-2-1）：

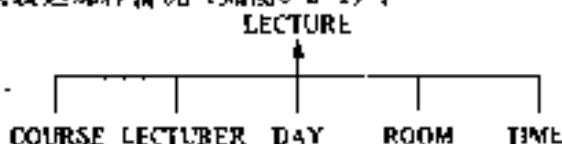


图 0-2-1

如果聚合含有某个复合对象，则在关系框架中该对象的标识符被包含在聚合对象中作为它的某个属性而出现。在上面这个例子中意味着除了ROOM、DAY和TIME是实体型LECTURE的属性外，课程标识符COURSE-ID和讲课者标识符LECTURER-ID也是LECTURE的属性。从而得出抽象对象LECTURE的关系表示：

LECTURE(LECTURE-ID, COURSE-ID,
LECTURER-ID, DAY, ROOM, TIME)

标识符是某个实体型的某种（或一组）属性，它的值确定了在该应用环境中属于该类型的唯一对象。我们把这样的属性称为（主）键。在某个表示相同类型对象集合的表（关系）中，一个元组至多有一个给定的键值。十分明显，每个数据库关系必须至少有一个键。

在为聚合对象LECTURE所规定的完整性约束中，有一个是特别有趣的。它要求每次开课总要有某门现有的课和某位讲课者，这称之为参照完整性约束。注意到为了维护这一完整性约束，作用于参加到这一联系中的实体上的操作必须适当地定义。若取消了某门课或解聘了某位讲课者则相应的开课属性也必须适当更新。同样，在计划开某门课时（往关系LECTURE中插入时），其属性COURSE-ID及LECTURER-ID必须适当选择。