

· 计算机信息服务职业教育系列教材.

SQL Server 2000 数据库项目案例开发 理论篇

王维平 杨波/主编



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

计算机信息服务职业教育系列教材

SQL Server 2000 数据库 项目案例开发

理论篇

主 编 王维平 杨 波

副主编 郁 云 张文雯

参 编 张 永 魏 瑾 夏月平

东南大学出版社

· 南京 ·

内 容 提 要

本书从实用的角度出发,系统、完整地讲述了当前数据库技术的原理和应用实践,主要包括:数据库系统绪论、关系数据库、SQL 语言、关系数据理论、数据库设计等,主要章节后附有习题。

本书内容丰富、结构清晰,概念和关键技术讲解清楚,同时配套教材《SQL Server 2000 数据库项目案例开发·实践篇》,以丰富的示例展示的具体应用。两者结合具有很强操作性和实用性,可作为高职院校“SQL Server 数据原理与应用”的课程教材,或作为社会培训班“SQL Server 2000”的基础入门教材,同时对从事数据开发和管理人员也具有较高参考价值。

图书在版编目(CIP)数据

SQL Server 2000 数据库项目案例开发·理论篇 /
王维平,杨波主编. —南京:东南大学出版社,2009.9
(计算机信息服务职业教育系列教材)
ISBN 978-7-5641-1824-2

I. S… II. ①王… ②杨… III. 关系数据库—数据库
管理系统,SQL Server 2000—职业教育—教材
IV. TP311.138

中国版本图书馆 CIP 数据核字(2009)第 153623 号

东南大学出版社出版发行
(南京四牌楼 2 号 邮编 210096)

出版人:江 汉

网址: <http://press.seu.edu.cn>

电子邮件: press@seu.edu.cn

江苏省新华书店经销 南京京新印刷厂印刷

开本:787mm×1092mm 1/16 印张:7.5 字数:192 字

2009 年 9 月第 1 版 2009 年 9 月第 1 次印刷

ISBN 978-7-5641-1824-2

印数:1~4000 册 定价:18.00 元

(凡有印装质量问题,可直接向读者服务部调换。电话:025-83792328)

前 言

SQL Server 2000 是 Microsoft 公司推出的一个高性能大型关系数据库管理系统,它为企业客户和创建商业应用程序的独立软件供应商提供了数据库管理平台,它使用方便、功能强大。使用 SQL Server 2000 可以开发不同类型的应用程序,其中包括:数据仓库、数据复制、分布式数据库应用程序、Internet 和 Intranet 应用,以及管理工具等。

数据库技术是大型计算机领域中应用最广泛的技术之一,几乎遍及计算机应用的各个方面,是计算机本科专业及其他许多专业的必修课。对本科院校的学生来说,数据库原理中的大部分内容是比较容易掌握的,但对于国家示范性软件职业技术学院、高职高专学校、成人教育学院的学生来说,由于他们原先学习基础相对薄弱,理论的学习比较困难,而对这类学生来说更应强调应用技术的掌握,本科院校所使用的教材却往往偏重原理,不太适合。本书就是针对读者的特点而编写的数据库原理类教材,书中通过各种案例来解释数据库的原理及应用技术,通俗易懂,易教易学。

本书共 5 章,第 1 章阐述了数据库和数据库系统的基本概念,第 2 章主要介绍了关系数据库系统所普遍使用的关系模型,第 3 章以具体的实例详细介绍了 SQL 语言常用语句的语法、含义,第 4 章详细说明了关系数据库的规范化理论,第 5 章结合具体的事例详细分析了关系数据库的设计过程。

本书适合于作为示范性软件职业技术学院、高职高专学校、成人教育学院的数据库原理与应用课程的教材,也很适合作为数据库原理自学教材和参考书。对于非计算机专业的本科学生,如期望学到关键知识而只是使用的数据库,也可以采用本书作为教材。

本书由王维平、杨波、郁云、夏月平、张文雯、魏瑾共同编著而成。参与本书编写的还有张永老师。本书在编写过程中得到了南京信息职业技术学院信息服务学院各位同仁的大力支持,在此一并感谢。

由于编者水平所限,加之时间仓促,书中不足之处在所难免,恳请读者批评指正。

编 者

2009 年 9 月

信息服务职业教育系列丛书

编委会

主任：王维平

副主任：陈 宁 王 岚

编委：（按姓氏笔画排序）

丁 勇	丁 涛	丁 睿	王红梅
王文宁	杨 波	汤欣怡	陆兰华
李欣坤	李 谦	陈 诚	张 永
张文雯	郁 云	魏 瑾	夏月平



目 录

1 绪论	1
1.1 数据库系统概述	1
1.2 数据管理技术的产生和发展	2
1.2.1 人工管理阶段	2
1.2.2 文件管理阶段	3
1.2.3 数据库系统阶段	4
1.2.4 分布式数据库管理阶段	5
1.3 数据模型	5
1.3.1 数据模型的组成要素	5
1.3.2 概念模型	6
1.4 最常用的数据模型	9
1.4.1 层次模型	9
1.4.2 网状模型	11
1.4.3 关系模型	13
1.5 数据库系统结构	15
1.6 数据库系统的组成	17
1.7 数据库技术的研究领域	18
1.7.1 DBMS	18
1.7.2 数据库设计	19
1.7.3 数据库理论	19
1.8 本章小结	19
1.9 习题	19
2 关系数据库	20
2.1 关系模型概述	20
2.1.1 单一的数据结构——关系	20
2.1.2 关系操作	20
2.1.3 关系的三类完整性约束	20
2.2 关系数据结构及形式化定义	21
2.2.1 关系的基本术语	21
2.2.2 关系模式	22
2.2.3 关系数据库	22



2.3 关系的完整性	23
2.3.1 实体完整性规则	23
2.3.2 参照完整性规则	23
2.3.3 用户自定义完整性规则	24
2.4 关系代数	24
2.4.1 传统的集合运算	25
2.4.2 专门的关系运算	26
2.5 本章小结	29
3 关系数据库标准语言 SQL	30
3.1 SQL 概述	30
3.2 数据定义	31
3.3 数据查询	39
3.3.1 单表查询	40
3.3.2 连接查询	47
3.3.3 嵌套查询	50
3.4 数据更新	55
3.4.1 插入数据	55
3.4.2 修改数据	56
3.4.3 删除数据	57
3.5 视图	57
3.5.1 定义视图	58
3.5.2 查询视图	59
3.5.3 更新视图	59
3.5.4 视图的删除	60
3.5.5 视图的作用	60
3.6 数据控制	61
3.6.1 授权	61
3.6.2 收回权限	61
3.7 本章小结	62
3.8 习题	62
4 关系数据理论	63
4.1 关系模式设计基础	63
4.2 函数依赖	64
4.2.1 函数依赖的基本性质	65
4.2.2 函数依赖的导出性质	65
4.2.3 关系键的形式定义	66
4.3 规范化理论	67

4.3.1	规范化的概念	67
4.3.2	第二范式定义 2NF	67
4.3.3	第三范式定义 3NF	68
4.3.4	BCNF 范式定义	69
4.3.5	多值依赖	70
4.3.6	规范化小结	72
4.4	函数依赖公理	72
4.4.1	函数依赖公理	72
4.4.2	属性集 X 关于 F 的闭包	73
4.4.3	函数依赖的等价(覆盖)	75
4.4.4	函数依赖最小集	75
4.5	模式分解	77
4.5.1	模式分解定义	77
4.5.2	模式分解特性	77
4.5.3	分解的无损连接性和保持函数依赖的分解	77
4.5.4	模式分解算法	79
4.5.5	实例	80
4.6	本章小结	83
4.7	习题	83
第 5 章	数据库设计	85
5.1	数据库设计概述	85
5.1.1	数据库与信息系统	85
5.1.2	数据库设计的特点	85
5.1.3	数据库设计方法简述	86
5.1.4	数据库设计的基本步骤	86
5.2	需求分析	87
5.2.1	需求分析的任务	88
5.2.2	需求分析的方法	88
5.2.3	数据字典	90
5.3	概念结构设计	91
5.3.1	概念结构	91
5.3.2	概念结构设计的方法与步骤	92
5.4	逻辑结构设计	95
5.4.1	E-R 图向关系模型的转换	95
5.4.2	数据模型的优化	98
5.4.3	设计用户子模式	98
5.5	数据库的物理设计	98
5.5.1	数据库的物理设计的内容和方法	99



5.5.2	关系模式存取方法选择	99
5.5.3	确定数据库的存储结构	101
5.5.4	评价物理结构	102
5.6	数据库的实施和维护	103
5.6.1	数据的载入和应用程序的调试	103
5.6.2	数据库的试运行	104
5.6.3	数据库的运行和维护	104
5.7	本章小结	105
5.8	习题	105
参考文献		108



1 绪 论

数据库技术是计算机应用领域中非常重要的技术,它产生于 20 世纪 60 年代末,是数据管理的最新技术,也是软件科学的一个重要分支。本章节内容介绍了数据库系统的相关内容、数据库的发展过程以及数据库系统的基本概念,在此基础上着重介绍常用数据库系统的体系结构以及它们的特点以及数据库管理系统的体系结构。

1.1 数据库系统概述

数据库系统本质上是一个用计算机存储记录的系统。在介绍数据库的基本概念之前,首先介绍一些数据库常用的术语,主要是:数据、数据库、数据库管理系统。

(1) 数据是人们生活离不开的被视为人类社会中一种极其重要的资源。如企业的生产管理和产品销售数据,国家人口普查数据和国家自然资源的数据等,都是企业生产和国家建设中不可缺少的。这些数据之所以有价值,是因为这些数据是表现信息的,是载荷信息的物理符号。信息向人们提供关于现实世界的新的事实的知识,它反映了客观物的物理状态。例如:在学校的学生信息管理档案中,如果人们需要了解学生的学生的姓名、性别、出生年月、家庭地址、所在班级、所在的系别、入学年限等信息,那么可以这样来描述这些信息(张三,男,1984,南京市中山东路 100 号,G40511,计算机系,2009.9)。

而这样的信息就是数据,可以通过这个数据了解到该学生的一些基本信息。如该张三学生是南京人,1984 年出生,2009 年 9 月份入学,在计算机系就读,班级编号是 G40511。

(2) 数据库是长期存储在计算机中的、有组织、统一管理的相关数据的集合。通俗的讲数据库就是存储数据的“仓库”。它具有一般仓库的特点:如可以存放现实世界中的一些东西;这些东西是有组织的存入的,由专人保管。但除此之外又有所不同,数据库中的数据是存放在计算机的存储器(如磁盘)上;数据管理和使用是通过数据库系统来完成的。

所谓的数据库是有规则的存储在一起的、相关的、去掉了不要的无用的信息,用以供各种用户共享的数据集合。数据库可以为多种应用提供服务,不存在有害的或不必要的冗余,数据间联系紧密但数据的存储独立于使用它的程序。

(3) 数据库管理系统(Database Management System,DBMS)是专门用于建立和管理数据库的一套软件,介于应用程序和操作系统之间。它对数据库进行统一的管理和控制,以保证数据库的安全性和完整性。用户通过 DBMS 访问数据库中的数据,数据库管理员也通过 DBMS 进行数据库的维护工作。它提供多种功能,可使多个应用程序和用户用不同的方法在同时或不同时刻去建立、修改和询问数据库。

数据库管理系统提供以下主要功能:

① 数据定义, DBMS 提供数据定义语言。DDL, 可以方便的定义数据库中的各种对象。

② 数据操纵。DBMS 提供数据操纵语言。通过 DML, 可以实现数据库中数据的基本操作, 比如向表中插入一行数据, 删除表中的某些行数据, 修改表中的某些数据以及查询表中的数据等

③ 安全控制和并发控制。DBMS 提供数据控制语言(DCL)。通过 DCL, 可以控制什么情况下谁可以执行什么样的数据操作。另外, 由于数据库是共享的, 多个用户可以同时访问数据库, 这可能会引起访问冲突, 从而导致数据的不一致。为此, DBMS 还提供了并发控制的功能, 以避免并发操作时可能带来的数据不一致性问题。

④ 数据库备份与恢复。DBMS 提供了备份数据库和恢复数据库的功能。

数据库系统, 是采用了数据库技术的计算机系统, 它能够按照数据库的方式存储和维护数据, 并且能够向应用程序提供数据。狭义地讲, 数据库和数据库管理系统加在一起就构成了数据库系统; 广义地讲, 数据库系统通常由数据库、硬件、软件和人员四部分组成。

由于数据模型是数据库系统的基础, 因此人们就按数据模型来命名数据库系统, 如数据模型为层次模型、网状模型和关系模型, 则相应的数据库系统就称为层次数据库系统、网状数据库系统和关系数据库系统。

1.2 数据管理技术的产生和发展

数据库管理系统是一个最典型且使用最为广泛的树管理系统。目前, 几乎所有数据管理指的都是如何对数据进行分类、组织、储存、检索及维护, 它是数据处理的核心。随着计算机软硬件的发展, 数据管理技术经过不断的完善, 经历了如下三个阶段:

- (1) 人工管理阶段。
- (2) 文件管理阶段。
- (3) 数据库系统阶段。

1.2.1 人工管理阶段

20 世纪 50 年代中期以前, 计算机主要用于科学计算。那时的计算机硬件方面, 外存只有卡片、纸带及磁带, 没有磁盘等直接存取的存储的设备; 软件方面, 只有汇编语言, 没有操作系统和高级语言, 更没有管理数据的软件; 数据处理的方式就是批处理, 这些决定了当时的数据管理只能依赖人工来进行。

人工管理阶段的特点是:

(1) 数据不保存。当时的计算机主要用于科学和工程计算, 一个程序对应一组数据。在计算某一个问题时, 把需要处理的少量数据以二进制的形式穿孔在程序代码之后, 上机运算时同程序一起输入到内存中, 运行程序时读取数据并处理, 最后把运算结果输出出来, 计算完就退出, 没有将数据长期保存的必要。

(2) 没有专门的数据管理软件。数据需要由应用程序自己管理, 因此应用程序的设计者不仅要考虑数据的逻辑结构, 还要考虑数据的物理结构, 比如存储结构、存储方式、输入输出方式等等。一旦存储结构发生变化, 应用程序也要做相应的修改, 程序员的负担非常重,

数据的独立性也很差。

(3) 数据不共享。一组数据对应一个程序。当多个程序使用相同的数据时,必须各自定义,不能共享。因此程序与程序之间有大量的冗余数据。

(4) 数据不具独立性。数据逻辑结构或物理结构发生变化后,必须对应用程序做相应的修改,这样加重了程序员的负担。

人工管理阶段中应用程序和数据相互结合成一个整体,互相依赖,其关系如图 1-1 所示。

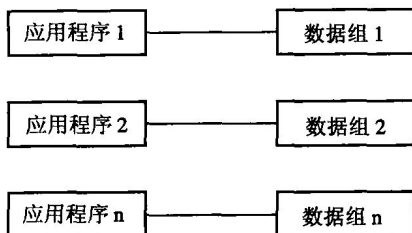


图 1-1 应用程序与数据的关系

1.2.2 文件管理阶段

20 世纪 50 年代末到 60 年代中期是数据管理的文件系统管理阶段。这时计算机硬件已经有了磁盘、磁鼓等直接存取的外存设备;软件则有了操作系统、高级语言,还有一个称为文件系统的软件。它将数据按照一定的规则组织起来,称为一个文件并起了一个名字存在磁盘上。应用程序通过文件名访问该文件,这时用户不必过多的考虑数据物理存储的细节,这一任务是由文件系统来完成,大大减轻了用户编程的劳动强度。文件系统充当应用程序和数据之间的接口,其关系如图 1-2 所示,这一阶段比人工管理阶段有了很大的改进,此时文件管理系统阶段的具有如下特点:

(1) 数据可以长期保存在磁盘上。用户可以反复对文件进行查询、修改、插入和删除等操作。

(2) 文件系统提供了数据与程序之间的存取方法。应用程序和数据有了一定的独立性。数据物理结构的改变也不一定反映在程序上,大大减轻了程序员的负担。

(3) 数据冗余量大。文件系统中,文件仍然是面向应用的,一个文件基本上对应于一个应用程序。即使多个程序使用了一部分相同的数据,也必须建立各自的文件,不能对数据项进行共享,因此冗余大存储空间浪费。由于数据可能有多个副本,对其中之一进行修改时还容易造成数据的不一致性。

(4) 文件之间是分离的。在文件系统中,各数据文件之间是彼此分离、分开存放的,数据之间的联系必须通过编程来实现。

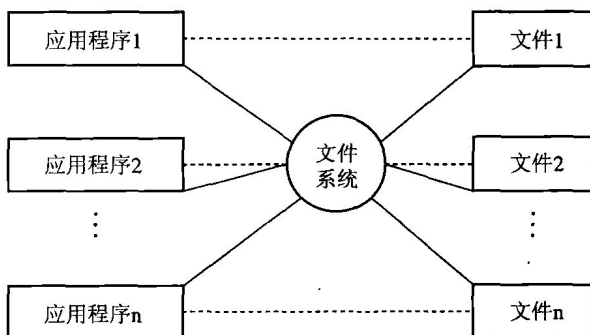


图 1-2 文件系统管理

1.2.3 数据库系统阶段

从 20 世纪 60 年代后期在计算机领域就开始进入数据库系统管理阶段。这一阶段计算机硬件价格大幅度下降,可靠性增强,磁盘技术已经成熟,为数据管理技术的发展奠定了物质基础。随着社会和生产的发展,需要处理的数据量急剧增加,数据处理的规模愈来愈大,并要求对数据进行集中控制管理,提供数据共享的要求也日益迫切,文件系统远不能满足这些要求。用数据库系统管理数据的基本思想是:将所有的数据按数据模型进行集中存放,即存放在数据库中,由一个软件(即 DBMS)实行统一的数据管理,应用程序通过 DBMS 访问数据。数据与程序之间的关系如图 1-3 所示。这种数据管理方法克服了以往所有管理方法存在的缺点,与以往的系统相比,数据库管理系统具有如下优点:

(1) 数据是集成的、共享的。在图书馆数据库系统中所有的数据都集中存储在一个数据库中,应用程序可以要求 DBMS 访问学生的信息和借书情况。应用程序员只需要指定需要的数据,不需编写程序来处理不同的文件。多个用户可以同时访问数据库,共享使用数据库。

(2) 数据很少重复。在数据库中,数据是很少重复的,如在图书馆管理系统中,学生信息表只存放学生的信息;图书信息只放在图书表中,而当学生借阅图书时,可以通过借阅信息中的信息了解到学生的信息,不需要在借阅信息中存放太多的学生信息,避免了数据的重复。

(3) 数据的独立性好。数据的独立性指应用程序不因物理存储格式和访问技术的改变而改变。数据的物理存储结构有 DBMS 来维护,应用程序不关心数据的物理存储结构,通过 DBMS 来访问数据库。

(4) 数据结构化。数据库系统中,实现了整体数据的结构化,把文件系统中简单的记录结构变成记录和记录之间的联系所构成的结构化数据。在描述数据的时候,不仅要描述数据本身,还要描述数据之间的联系。数据之间的联系通过存取路径来实现,把相关的数据有机组织在一起。

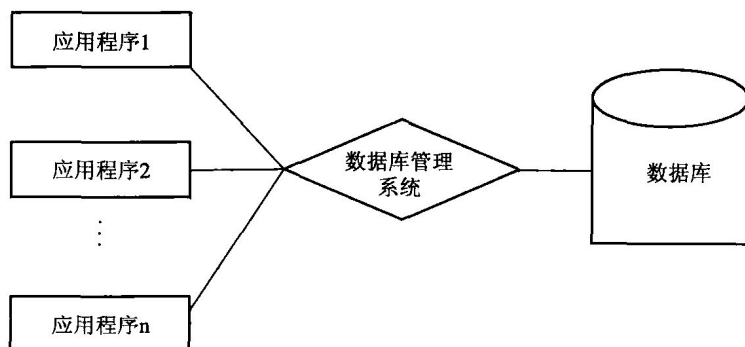


图 1-3 数据库系统管理阶段

DBMS 是运行在操作系统之上的数据库管理系统软件,由它实施对外存上的数据库进行统一管理,并负责执行在 DBMS 之上开发的应用程序对数据库的全部操作。

DBMS 除了同操作系统配合按照用户的要求存取数据库中的数据外,主要还具有以下



四个方面的管理控制功能:

- (1) 安全性控制。
- (2) 一致性控制。
- (3) 并发性控制。
- (4) 数据库恢复。

1.2.4 分布式数据库管理阶段

分布式数据库系统通过计算机网络和通信线路可以把分布在不同地域的、不同局域网环境下的、不同类型的数据库系统连接和统一管理起来。分布式数据库系统既支持客户的局部应用,又支持客户的全局应用。

1.3 数据模型

数据库是基于数据模型建立的。通俗地讲,数据模型就是数据的组织方式,并可以用某一种数据结构来描述。数据模型是数据特征的抽象,它不是描述个别的数据,而是描述数据的共性。它一般包括两个方面:一是数据的静态特性,包括数据的结构和限制;二是数据的动态特性,即在数据上的所定义的运算或操作。数据模型,第一要能够真实反映现实世界,否则就没有实际意义了;其次,要易于理解,必须和人们对外部事务的认识相一致;最后,因为最终还是需要由计算机来处理,所以还要能够便于实现。在数据库技术的发展过程中,主要出现过三种数据模型,即层次数据模型、网状数据模型和关系数据模型。数据模型通常由数据结构、数据操作和完整性约束三要素组成。

1.3.1 数据模型的组成要素

通俗地讲,数据模型是数据的组织方式,并可以用某种数据结构来描述,任何一种数据模型都必须是严格定义的概念的集合。这些概念必须能够精确地描述系统的静态特性、动态特性和完整性约束条件。因此数据模型通常都是由数据结构、数据操作和完整性约束三个要素组成。

1. 数据结构

数据结构用于描述系统的静态特性

数据结构是所研究的对象类型(object type)的集合。这些对象是数据库的组成成分,它们包括两类,一类是与数据类型、内容、性质相关的对象,例如关系模型中的域、属性、关系等;一类是与数据之间联系有关的对象,例如关系模型中反应联系的关系,网状模型中的系型(set type)。

数据结构是对系统静态特征的描述,是刻画一个数据模型性质最重要的方面。因此在数据库系统中,人们通常按照其数据结构的类型来命名数据模型。例如,层次结构、网状结构和关系结构的数据模型分别命名为层次模型、网状模型和关系模型。

2. 数据操作

数据操作是对系统动态特性的描述,是对数据库中各种数据对象及实例允许执行的操作的集合。数据操作包括对象的创建、删除和修改,对数据的检索和更新(包括插入、删除、

修改)及其他的相关操作。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。

3. 数据的约束条件

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约束和储存规则,用来限定符合数据模型的数据库的状态以及状态的变化,以确保数据的正确性、有限性和相容性。

数据模型必须反映和规定本数据模型应该遵守基本的通用的完整性约束条件。例如,在关系模型中,任何关系必须满足实体完整性和参照完整性两个条件(第2章将详细讨论这两个完整性约束条件)。此外,数据模型还应该提供定义完整性约束条件的机制,以反映具体应用所涉及的数据必须遵守的特定的语义约束条件(称为用户自定义完整性约束)。例如,在企业的数据库中规定员工的年龄不得小于18周岁,并且不能大于60周岁。

1.3.2 概念模型

数据模型是数据库系统的核心和基础,是对现实世界数据特征的抽象,用来描述数据的一组概念和定义。各种机器上实现的DBMS软件都是基于某种数据模型的。为了把现实世界中的具体事务抽象、组织为某一DBMS支持的数据模型,人们常常首先将现实世界抽象为信息世界,然后将信息世界转换为机器世界。也就是说,首先把现实世界中的客观对象抽象为某一种信息结构,这种信息结构并不依赖于具体的计算机系统,不是某一个DBMS支持的数据模型,而是概念级的模型;然后再把概念模型转换为计算机上某一个DBMS支持的数据模型,这一过程如图1-4所示。不难看出,概念模型实际上是现实世界到机器世界的一个中间层次。

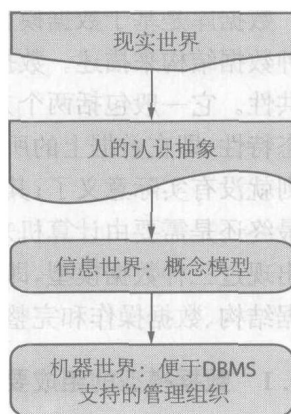


图 1-4 三个世界的抽象认识

概念模型是对信息世界各类对象、属性及联系等数据的描述,用于信息世界的建模,是现实世界到信息世界的第一层抽象。概念模型是用户与数据库设计人员之间进行交流的语言,因此一方面概念模型应该具有较强的语义表达能力,能够方便、直接地表达客观应用中的各种语义知识,另一方面它还应该简单、清晰、易于理解。

概念模型是对信息世界建模,所以模型应该能够方便、准确地表示出上述信息世界中的常用概念。概念模型的表示方法很多,其中的实体-联系方式最为常用的,是由美籍华人 P. P. S. Chen 于 1976 年提出的。该方法用 E-R 图来描述现实世界的概念模型。

信息世界涉及的概念主要有:

(1) 实体(entity)

实体实例(entity instance)简称实例,是现实世界中客观存在的并可相互区别的事物,实体实例可以是具体的人、事、物。例如,一个叫“李明”的学生,一个称为“数据库原理”的课程,都是一些实体实例。而实体是客观存在并可相互区别的事物。具有相同属性的实例必然具有共同的特征和性质。例如,学生(学号,姓名,性别,出生年份,系,入学时间)就用实体名及其属性名集合来抽象和刻画同类实体。

实体型：用矩形表示，矩形框内写明实体名称。

(2) 属性(attribute)

实体所具有的某一特性称为属性。一个实体可以由若干个属性来刻画。属性(attribute)用来描述实体的特征。例如，学生实体可以由学号、姓名、性别、出生年份、班级、系部、入学时间等属性组成(4051101, 张山, 男, 1976, 40511s, 计算机系, 1994)。这些属性组合起来表征了一个学生实例。

属性：用椭圆形表示，椭圆内写明属性名，并用无向边将其与相应的实体连接起来。

(3) 码(key)

唯一标识实体的属性集称为码。例如，学号是学生实体的码。

(4) 联系(relationship)

在现实世界中，事务不是孤立的，事务内部以及事务之间是有联系的，这些联系在信息世界中反映为实体和实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系。

联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型(1:1, 1:n 或 m:n)。

需要注意的是，联系本身也是一种实体，也可以有属性。如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来。

两个实体型之间的联系可以分为三类。

① 一对一联系(1:1)

如果对于实体集 A 中的每一个实体，实体集 B 中至多有一个实体与之联系；对于实体 B 中的每一个实例，实体 A 中也至多有 1 个实例与之联系，则称实体 A 与实体 B 具有一对一联系。记为 1:1(读着 1 对 1)。在 1:1 联系中，一种类型实体的一个实例至多与另一种类型实体的一个实例关联。图 1-5(a)“拥有”联系将单个学生与每本借书证关联，表明一个学生有且只拥有一本借书证，一本借书证只能被一个学生拥有。

② 一对多联系(1:n)

如果对于实体 A 中的每一个实例，实体 B 中多有一个实例($n \geq 0$)与之联系，反之，对于实体 B 中的每一个实例，实体 A 中至多只有一个实例与之联系，则称实体 A 与实体 B 有一对多联系。记为 1:n。例如图 1-5(b)一个班级中有若干名学生，而每个学生只在一个班

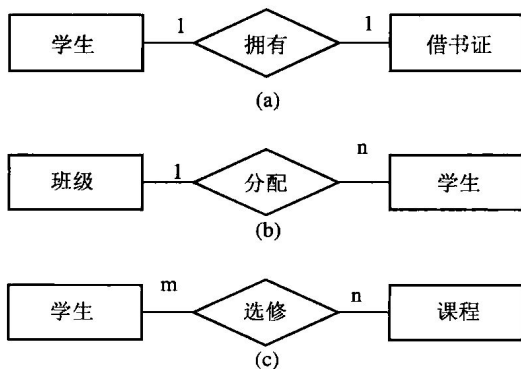


图 1-5 三种类型的二元联系

级中学习,班级与学生之间的联系我们称为“分配”,是一对多联系。

③ 多对多联系($m:n$)

如果对于实体 A 中的每一个实例,实体 B 中有 n 个实例($n \geq 0$)与之联系,反之,对于实体 B 中的每一个实例,实体 A 中也有 m 个实例($m \geq 0$)与之联系,则称实体 A 与实体 B 具有多对多联系。记为 $m:n$ 。

例如,图 1-5(c)一门课程同时有若干个学生选修,而一个学生可以同时选修多门课程,则课程与学生之间具有多对多联系。

实际上,一对一联系是一对多联系的特例,而一对多联系又是多对多联系的特例。

实体型之间的这种一对一、一对多、多对多联系不仅存在于两个实体型之间,也存在于两个以上的实体型之间。两个以上实体之间的联系称为多元联系。图 1-6 给出了三元联系的实例。

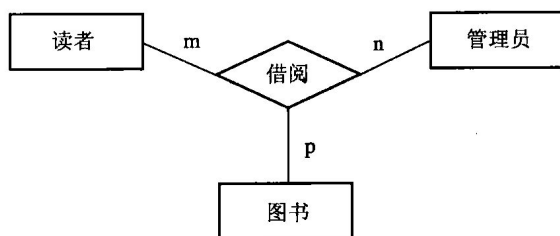


图 1-6 管理员管理图书借阅

多实体型之间一对一、多对多联系的定义及其例子请读者自行给出。

同一个实体集内的各实体之间也可以存在一对一、一对多、多对多的联系。例如,学生实体集内部具有领导与被领导的联系,即某一学生(班干部)“领导”若干名学生,而一个学生仅被另外一个学生直接领导,因此这是一对多的联系。如图 1-7 所示。

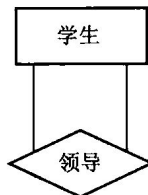


图 1-7 递归联系

下面用 E-R 图表示学生、课程的概念模型,一个学生可以选修多门课程,一门课程可以被多个学生选修,学生与课程之间由多对多的选课联系。各个实体由下列属性:

- 学生: 学号,姓名,性别,年龄。
- 课程: 课程号,课程名,学分。

这两个实体的属性用 E-R 图表示,如图 1-8 所示,选修这个联系又都分别具有各自的属性,如选修时间、成绩。

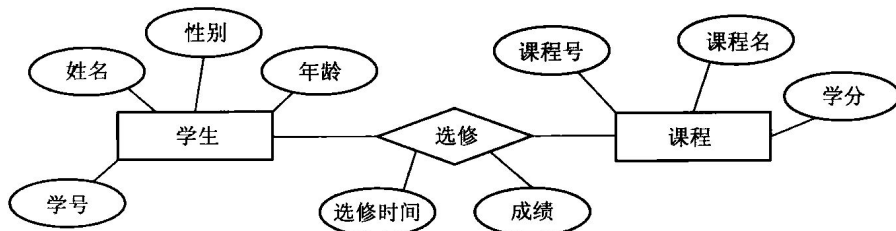


图 1-8 学生选修 E-R 图