

网络数据库 技术应用

本书配有
教学课件

周玲艳 张希 等编著

1.138SQ
5



机械工业出版社
China Machine Press

计算机基础课程系列教材

TP311.13852

zh/85

网络数据库 技术应用

周玲艳 张希 等编著



机械工业出版社
China Machine Press

建立网络数据库应用程序需要数据库系统的使用、数据库的设计和网路应用程序的设计及开发等方面的知识。本书探讨了网络数据库系统的实现理论、技术和方法。

本书共 7 章,介绍了数据库的概念模型、数据模型、关系数据库的特点等基本知识,以选课系统数据库为背景,介绍了 MySQL 数据库中支持的 SQL 语句的语法、JDBC 相关知识、通过 JDBC 访问数据库、事务操作等内容,以及如何进行 JSP 编程实现在页面上访问数据,数据库系统设计的全过程。

本书适用于大专院校网络数据库课程的教材,也可以作为网络数据库应用系统开发人员的参考书。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

网络数据库技术应用/周玲艳等编著. —北京:机械工业出版社, 2008. 8
(计算机基础课程系列教材)

ISBN 978-7-111-24609-1

I. 网… II. 周… III. 关系数据库—数据库管理系统—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2008)第 110572 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:王 玉

三河市明辉印装有限公司印刷·新华书店北京发行所发行

2008 年 8 月第 1 版第 1 次印刷

184mm × 260mm · 13.75 印张

标准书号: ISBN 978-7-111-24609-1

定价: 25.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010)68326294

前 言

随着计算机应用的普及和互联网技术的迅速发展,数据量急剧地增加,各种信息系统应运而生。这些系统绝大部分都是用数据库作为后台技术支撑,借助数据库技术科学地保存和管理大量的、复杂的数据。

网络数据库不仅可以提供数据库的所有服务,还可以直接为网络服务,以便方便而充分地利用宝贵的数据、信息资源。所以,建立网络数据库应用程序需要数据库管理系统的使用、数据库的设计和应用程序的设计及开发等方面的知识。本书主要探讨网络数据库系统的实现理论、技术和方法。

本书共7章。第1章介绍数据库基本知识,包括数据库、数据库管理系统的基本概念、概念模型和数据模型、关系数据库的特点、数据库保护、网络数据库应用技术等内容。第2章介绍MySQL数据库的基本概念、安装方法、MySQL支持的数据类型、函数和运算符。第3章以选课系统数据库为背景,介绍MySQL数据库中支持的SQL语句的语法。第4章讲述JDBC相关知识,包括JDBC驱动、JDBC API,以及如何通过JDBC访问数据库,事务的基本概念和事务操作等内容。第5章介绍JSP的基本语法和运行环境,仍以选课系统数据库为背景,介绍如何进行JSP编程实现在页面上访问数据。第6章介绍数据库设计的全过程,重点介绍如何进行需求分析、概念结构设计和逻辑结构设计,并讲述选课系统数据库是如何使用这些理论设计出来的,最后通过一个图书管理系统数据库设计实例再次将介绍的理论知识加以应用。第7章的内容为实训指导,首先是MySQL数据库实训,通过建立小型电子商务系统数据库进行数据库的相关练习;其次是JDBC和JSP使用实训,有关JDBC的基本练习和使用JSP编程访问数据库;最后建立了小型电子商务系统应用程序。

本书由周玲艳主笔,张希参与了本书部分章节的编写和资料搜集、整理工作。在写作过程中,孙建华教授给予了精心的指导,并提出了许多宝贵的意见。在此向她表示由衷的感谢。感谢给予我帮助和指导的陈军老师、袁玫老师、朱熹福老师、刘总路老师和各位同事同仁。

由于时间紧张,学识有限,错误之处在所难免,欢迎广大读者和专家批评指正,以期修订更新。

作者

目 录

前言	23
第1章 数据库概述	1
1.1 数据库系统基本概念	1
1.1.1 数据库	1
1.1.2 数据库管理系统	1
1.1.3 数据库系统	2
1.2 概念模型与数据模型	2
1.2.1 概念模型	3
1.2.2 数据模型	7
1.3 关系数据库	10
1.3.1 关系模型概述	10
1.3.2 关系的完整性	12
1.3.3 关系数据语言	13
1.4 与数据库沟通——SQL 简介	13
1.4.1 SQL 语言概述	13
1.4.2 SQL 语言的特点	14
1.4.3 SQL 语言的组成	15
1.5 数据库保护	15
1.5.1 数据库的安全性	15
1.5.2 并发控制	16
1.5.3 故障恢复	16
1.6 网络数据库应用技术	17
1.6.1 数据库应用系统的体系结构	17
1.6.2 Web 数据库访问技术	18
本章小结	20
练习	20
第2章 MySQL 基础	22
2.1 MySQL 简介	22
2.1.1 MySQL 的基本概念和特点	22
2.1.2 几种常用数据库的特点	22
2.2 MySQL 的安装和使用	23
2.2.1 MySQL 的获得	23
2.2.2 MySQL 的安装	24
2.2.3 目录结构	30
2.2.4 MySQL 的基本使用	31
2.3 MySQL 数据类型	32
2.3.1 数值类型	32
2.3.2 字符串类型	35
2.3.3 日期和时间类型	38
2.3.4 字段类型的选择	39
2.4 MySQL 函数	40
2.4.1 字符串函数	40
2.4.2 日期时间函数	43
2.4.3 聚合函数	46
2.4.4 其他函数	47
2.5 MySQL 运算符	47
2.5.1 分组运算符	47
2.5.2 算术运算符	47
2.5.3 比较运算符	48
2.5.4 逻辑运算符	49
2.5.5 模式匹配运算符	50
本章小结	51
练习	51
第3章 MySQL 的 SQL 语法	52
3.1 MySQL 中的命名规则	53
3.1.1 字符、长度规则	53
3.1.2 大小写规则	53
3.2 数据库操作	54
3.2.1 创建数据库	54
3.2.2 使用数据库	55
3.2.3 删除数据库	55
3.3 表操作	56
3.3.1 创建表结构	56

3.3.2 修改表结构	60	4.3.6 处理结果	95
3.3.3 删除表	61	4.3.7 关闭连接	96
3.4 记录操作	61	4.4 事务操作	96
3.4.1 新增记录	61	本章小结	97
3.4.2 删除记录	63	练习	97
3.4.3 更改记录	63	第5章 使用 JSP 编程访问数据库	99
3.4.4 查询记录	64	5.1 JSP 编程基础	99
3.5 信息查询	70	5.1.1 JSP 概述	99
3.6 索引	72	5.1.2 JSP 运行环境——Tomcat 的安装 和配置	99
3.6.1 索引简介	72	5.1.3 JSP 语法简介	105
3.6.2 建立索引	73	5.2 选课系统的开发	109
3.6.3 删除索引	74	5.2.1 选课系统数据库	109
3.7 视图	74	5.2.2 开发及运行环境	112
3.7.1 创建视图	75	5.2.3 JSP 页面	114
3.7.2 更改视图	75	5.2.4 开发注意事项	126
3.7.3 删除视图	75	本章小结	127
3.8 导入、导出数据	75	练习	127
3.8.1 导出数据	76	第6章 数据库设计理论及实例	128
3.8.2 导入数据	76	6.1 数据库设计理论	128
3.9 综合实例	77	6.1.1 数据库设计概述	128
本章小结	81	6.1.2 数据库设计方法	128
练习	81	6.2 数据库设计的基本步骤	129
第4章 使用 JDBC 连接数据库	83	6.2.1 需求分析	130
4.1 JDBC 概述	84	6.2.2 概念结构设计	134
4.1.1 JDBC 简介	84	6.2.3 逻辑结构设计	136
4.1.2 JDBC 应用程序的体系 结构	85	6.2.4 物理结构设计	137
4.1.3 JDBC 驱动	86	6.2.5 数据库实施	138
4.2 JDBC API	88	6.2.6 数据库运行和维护	138
4.2.1 DriverManager 类	88	6.3 选课系统数据库设计实例	139
4.2.2 Connection 类	90	6.3.1 选课系统需求分析	139
4.2.3 Statement 类	91	6.3.2 选课系统概念结构设计	141
4.2.4 ResultSet 类	92	6.3.3 选课系统逻辑结构设计	143
4.3 通过 JDBC 访问数据库	92	6.3.4 选课系统数据库结构设计	143
4.3.1 加载驱动程序	92	6.4 图书管理系统数据库设计实例	143
4.3.2 定义连接的 URL	93	6.4.1 图书管理系统需求分析	143
4.3.3 建立连接	93	6.4.2 图书管理系统概念结构 设计	153
4.3.4 创建 Statement 对象	94	6.4.3 图书管理系统逻辑结构 设计	154
4.3.5 执行 SQL 查询或者其他 SQL 操作	94		

6.4.4 图书管理系统数据库结构	155
设计	155
6.4.5 图书管理系统数据库的	
生成	158
本章小结	160
练习	160
第7章 综合实训——构建小型电子	
商务系统	161
7.1 小型电子商务系统简介	161
7.2 MySQL 实训	161
7.2.1 连接和退出 MySQL 数据库管理	
系统	161

7.2.2 创建数据库和表	162
7.2.3 查询表	166
7.3 JDBC 和 JSP 的使用实例	167
7.3.1 JDBC 的使用	167
7.3.2 使用 JSP 编写页面程序	167
7.4 使用 JSP + JDBC + MySQL 构建小型	
电子商务系统	168
7.4.1 总体设计	168
7.4.2 主要 JavaBean 的编写	169
7.4.3 客户端主要功能模块设计	189
7.4.4 管理端主要功能模块设计	201
参考文献	212

第1章 数据库概述

随着社会的迅速发展、科技的不断进步，人们对各种数据的需求越来越多。日益庞大的各种数据已成为国家、社会的重要资源和财富，再用纸和笔来记录这些数据已经完全跟不上时代发展的脚步。此时，数据库技术作为信息系统的基础和核心应运而生，并且得到了越来越广泛的应用。

1.1 数据库系统基本概念

1.1.1 数据库

对于数据库这个概念，当人们从不同的角度来描述时，就有不同的定义。例如，称数据库是一个“记录保存系统”（该定义强调了数据库是若干记录的集合）；又如称数据库是“人们为解决特定的任务，以一定的组织方式存储在一起的相关数据的集合”（该定义侧重于数据的组织）；更有甚者，称数据库是“一个数据仓库”。这些说法虽然形象，但并不严谨。

严格地说，数据库(DataBase, DB)是长期存储在计算机内、大量的、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型进行组织、描述和储存，有以下特点：

- 最小冗余：数据库中的数据尽可能不重复。
- 数据独立性：数据库中的数据与应用程序没有依赖关系。
- 安全性：保护数据库，以防止不合法使用。
- 完整性：存取数据库中数据的过程要确保其正确性、一致性和有效性。
- 数据共享：数据库中的数据可以同时为多个用户和多个应用程序服务。

1.1.2 数据库管理系统

数据库管理系统(Database Management System, DBMS)是一个用来管理数据库的软件系统，它能科学地组织和存储数据，以高效地获取和维护数据。

数据库管理系统是位于用户与操作系统之间的一层数据管理软件，主要用来定义和管理数据库，处理数据库与应用程序之间的联系。它的主要功能包括以下几个方面：

- 数据定义功能：数据库管理系统提供数据定义语言(Data Definition Language, DDL)，用户通过它可以方便地对数据库中的数据对象进行定义。
- 数据操纵功能：数据库管理系统提供数据操纵语言(Data Manipulation Language, DML)，用户可以使用DML操纵数据，实现对数据库的基本操作，如查询、插入、删除和修改等。
- 数据库的运行管理：数据库在建立、运行和维护时，由数据库管理系统统一管理、控制，以保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

- 数据库的建立和维护功能：包括数据库初始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重组功能和性能监视、分析功能等。

1.1.3 数据库系统

数据库系统(DataBase System, DBS)是指基于数据库的计算机应用系统, 主要包括数据库、数据库管理系统。与其相关的其他内容包括: 应用系统、数据库管理员和用户。

数据库系统可以用图 1-1 表示。

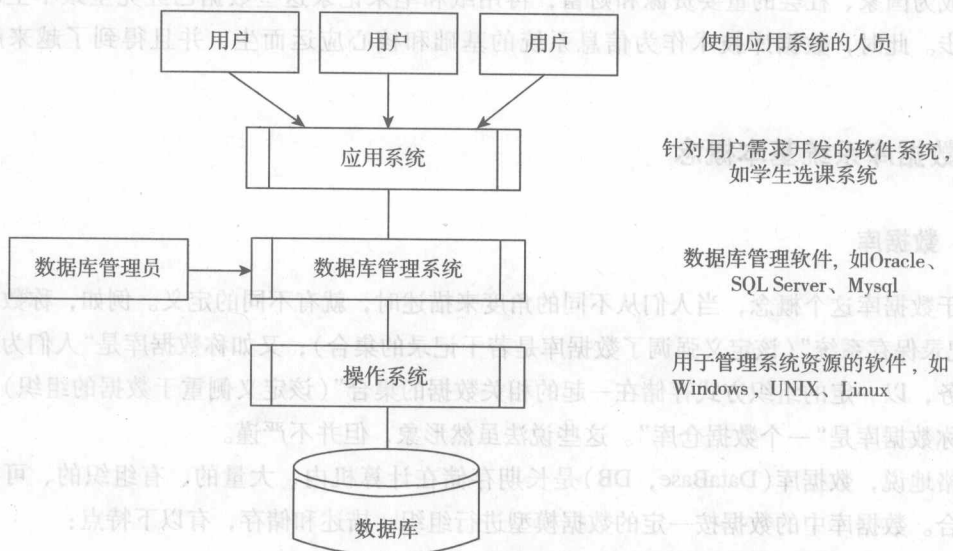


图 1-1 数据库系统组成

数据库系统的出现使信息系统从以加工数据的程序为中心转向围绕共享的数据库为中心的新阶段。这样既便于数据的集中管理, 又有利于应用程序的研制和维护, 提高了数据的利用率和相容性, 提高了决策问题的可靠性。

1.2 概念模型与数据模型

模型是对现实世界中信息、内容的模拟和抽象表达, 人们通过模型就可以联想到现实生活中的事物。比如, 买房时, 通过楼房的模型图可以想象房屋建成后的结构、布局和周围情况等。数据模型也是一种模型, 它是对现实世界数据特征的抽象。

数据库是数据的集合, 它不仅反映数据本身的内容, 而且反映数据之间的联系。由于计算机不可能直接处理现实世界中的具体事物, 所以必须把具体事物转换成计算机能够处理的数据。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。通俗地讲, 数据模型就是对现实世界的模拟。

现有的数据库系统均是基于某种数据模型的。因此, 了解数据模型的基本概念是学习数据库的基础。不同的数据模型提供给我们不同的模型化数据和信息的工具。根据模型应用的不同目的, 可以将这些模型划分为不同层次的两类。

第一类是概念模型, 也称信息模型, 它是按用户的观点对数据和信息建模, 主要用于数据库

的设计。另一类是数据模型，主要包括网状模型、层次模型和关系模型等，它是按计算机系统的观点对数据建模，主要用于数据库的实现。

为了把现实世界中的具体事物抽象、组织为某一数据库管理系统支持的数据模型，人们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，首先将现实世界中的客观对象抽象为不依赖于任何具体机器的信息结构，这种信息结构不是某一个数据库管理系统支持的数据模型，而是概念级的模型；然后再把概念模型转换成具体机器上数据库管理系统支持的数据模型。

现实世界、信息世界、机器世界这三个领域是由客观到认识、由认识到使用管理的三个不同层次，后一领域是前一领域的抽象描述。在这三个世界中，有着各自领域的术语，其对应关系如表 1-1 所示。

表 1-1 现实世界、信息世界、机器世界术语的对应关系

现实世界	信息世界	机器世界
事物	实体	记录
特征	属性	数据项
事物集合	实体集	数据或文件
事物间的联系	实体间的联系	数据间的联系
人的认识	概念模型	数据模型

1.2.1 概念模型

概念模型对现实世界进行建模，通过概念模型可生成机器世界的数据模型。概念模型可以看成是现实世界到机器世界的一个过渡的中间层次，如图 1-2 所示。

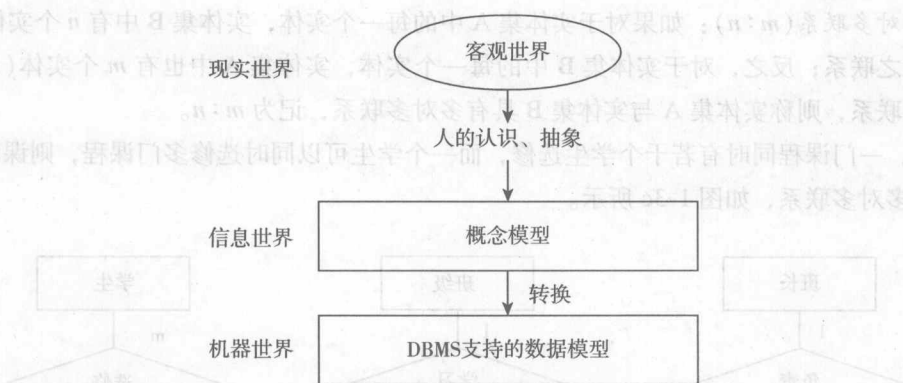


图 1-2 概念模型与数据模型的关系

概念模型不依赖计算机及任何数据库管理系统，是对现实世界的真实、全面的反映。下面介绍概念模型涉及的基本概念和表示方法。

1. 概念模型中的基本概念

概念模型中涉及的概念主要有：实体、属性、实体集、实体间的联系。

(1) 实体(Entity): 在现实世界中客观存在并可相互区别的事物。实体可以是具体的人、事、物, 也可以是抽象的概念, 如一个学生、一个职工、一部汽车、一个部门、一个班级、一门课等都是实体。

(2) 属性(Attribute): 实体所具有的某一特性。一个实体可以由若干个属性来描述。例如学生实体可以由学号、姓名、性别、出生年份、系、入学时间(如: 00001, 李丽, 女, 1980, 计算机系, 2002)等属性组成, 这些属性组合起来描述了一个学生。

- 码(Key): 唯一标识实体的属性集。例如, 学号是学生实体的码。

- 域(Domain): 属性的取值范围称为该属性的域。例如, 学号的域为5位整数, 姓名的域为字符串集合, 性别的域为男、女。

(3) 实体集(Entity Set): 同型实体的集合称为实体集。例如, 全体学生就是一个实体集。

(4) 联系(Relationship): 在现实世界中, 事物内部以及事物之间是有联系的, 这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系。实体之间的联系通常是指不同实体集之间的联系。

1) 两个实体集之间的联系可以分为三类。

- 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(也可以没有)实体与之联系, 反之亦然, 则称实体集A与实体集B具有一对一联系, 记为1:1。

例如, 学校里一个班级只有一个班长, 而一个班长只在一个班中任职, 则班级与班长之间具有一对一联系, 如图1-3a所示。

- 一对多联系(1:n): 如果对于实体集A中的每一个实体, 实体集B中有n个实体($n \geq 0$)与之联系; 反之, 对于实体集B中的每一个实体, 实体集A中至多只有一个实体与之联系, 则称实体集A与实体集B有一对多联系, 记为1:n。

例如, 一个班级中有若干名学生, 而每个学生只在一个班级中学习, 则班级与学生之间具有一对多联系, 如图1-3b所示。

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

例如, 一门课程同时有若干名学生选修, 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, 如图1-3c所示。

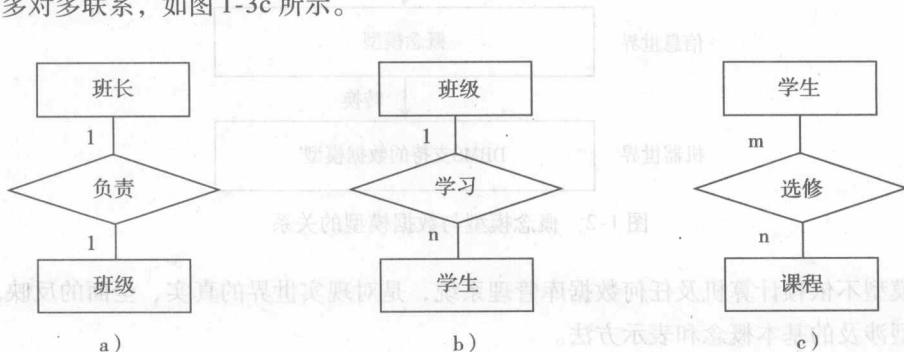


图 1-3 实体集间联系的实例

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。那么, 用图形来表示两个实体集之间的这三类联系, 如图 1-4 所示。

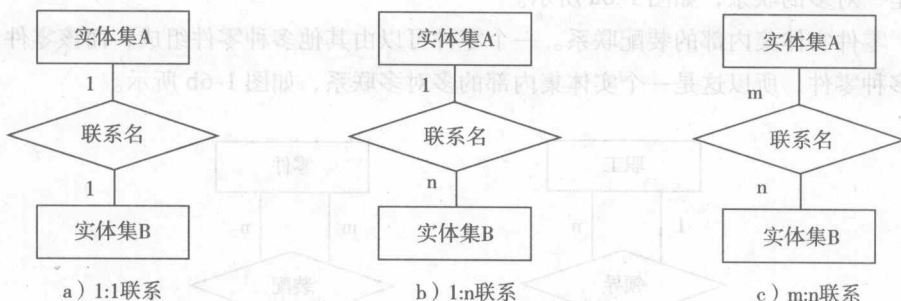


图 1-4 实体集间的联系

2) 多个实体集之间的联系有两类。

两个以上的实体集之间也会存在联系, 其联系类型一般为: 一对多和多对多。

- 一对多联系: 设实体集 $E_1, E_2, \dots, E_n$, 如果 $E_j (j=1, 2, \dots, n)$ 与其他实体集 $E_1, E_2, \dots, E_{j-1}, E_{j+1}, \dots, E_n$ 之间存在一对多的联系, 则对于 E_j 中一个给定实体, 可以与其他实体集 $E_i (i \neq j)$ 中一个或多个实体联系, 而实体集 $E_i (i \neq j)$ 中一个实体最多只能与 E_j 中的一个实体联系, 则称 E_j 与 $E_1, E_2, \dots, E_{j-1}, E_{j+1}, \dots, E_n$ 之间的联系是一对多的。

例如, 一门课程可以有若干教师讲授, 使用若干本参考书; 一个教师只讲授一门课程, 每本参考书只供一门课程使用, 则课程与教师、参考书之间的联系是一对多的, 如图 1-5a 所示。

- 多对多联系: 在两个以上的多个实体集之间, 当一个实体集与其他实体集之间均存在多对多联系, 而其他实体集之间没有联系时, 这种联系称为多实体集间的多对多联系。

例如, 有三个实体集: 供应商、项目和零件。一个供应商可以供给多个项目、多种零件; 每个项目可以使用多个供应商供应的零件; 每种零件可由不同供应商供给。因此, 供应商、项目和零件三个实体集之间是多对多的联系, 如图 1-5b 所示。

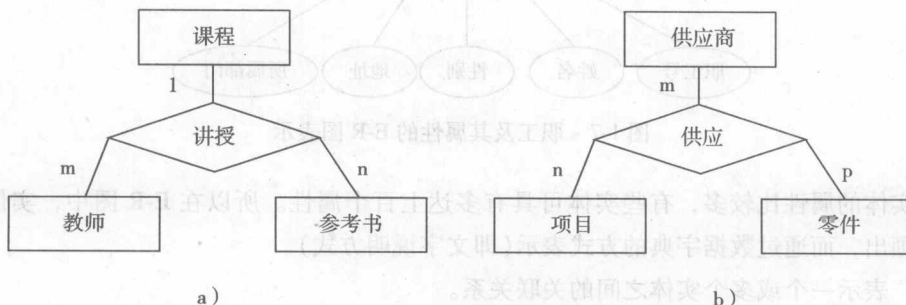


图 1-5 三个实体集间联系的实例

3) 实体集内部的联系。

在一个实体集内的实体之间也可以存在一对多或多对多的联系。

例如, 职工是一个实体集, 职工中有领导, 而领导自身也是职工。职工实体集内部具有领导与被领导的联系, 即某一个领导级职工领导若干名职工, 而一个职工仅被一个领导级职工所管, 这种联系是一对多的联系, 如图 1-6a 所示。

又如, 零件实体集内部的装配联系。一个零件可以由其他多种零件组成, 而该零件又可以组装成其他多种零件, 所以这是一个实体集内部的多对多联系, 如图 1-6b 所示。

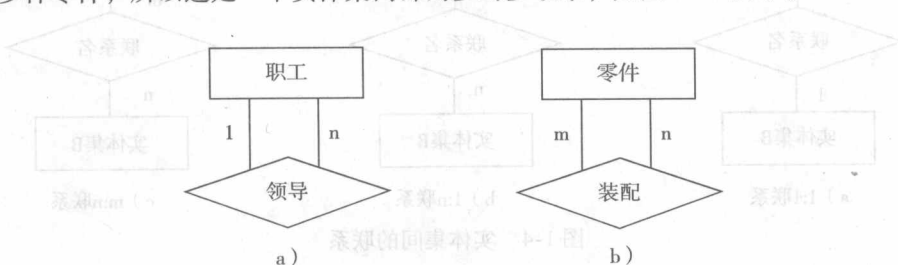


图 1-6 实体集内部联系的实例

2. 概念模型的表示方法

概念模型的表示方法很多, 其中最为著名和使用最为广泛的是 P. P. Chen 于 1976 年提出的实体-联系方法(Entity-Relationship Approach), 简称 E-R 图法。该方法用 E-R 图来描述现实世界的概念模型, 提供了表示实体、属性和联系的方法。E-R 图也称为 E-R 模型。

E-R 图包括 3 个基本要素:

实体: 一个数据对象, 指应用中可以区别的客观存在的事物, 如人、部门、商品等。

在 E-R 图中, 实体用矩形框表示, 在矩形框内写明实体名称。

属性: 实体的某一特性称为属性, 如职工有职工号、姓名、性别、年龄、所属部门等属性。

在 E-R 图中, 属性用椭圆表示, 并用无向边(即两端没有箭头的边)将其与相应的实体连接起来, 如图 1-7 所示。

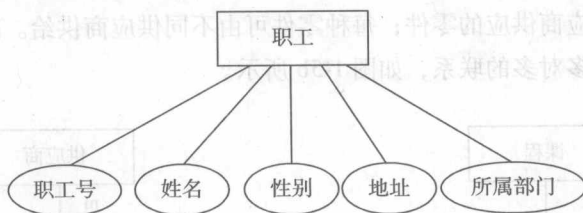


图 1-7 职工及其属性的 E-R 图表示

由于实体的属性比较多, 有些实体可具有多达上百个属性。所以在 E-R 图中, 实体的属性可不直接画出, 而通过数据字典的方式表示(即文字说明方式)。

联系: 表示一个或多个实体之间的关联关系。

在 E-R 图中, 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, 同时在无向边上表明联系的类型(如: 1:1 或者 1:n, 或 m:n)。如果联系具有属性, 则该属性仍用椭圆框表示, 仍需要用无向边将该属性与其联系连接起来。联系的属性必须在 E-R 图上标出, 不能通过数据字典说明。例如, 供应商、项目和零件之间存在有供应联系, 该联系有供

应量属性,如图 1-8 所示。

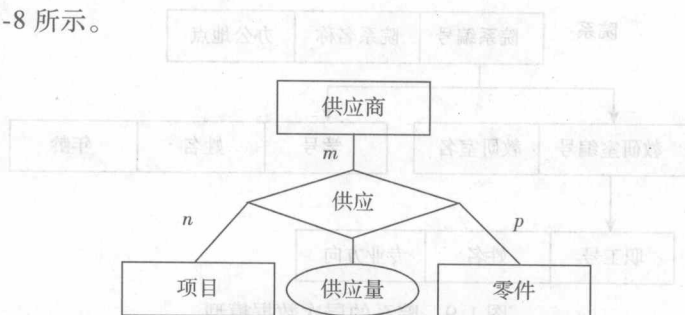


图 1-8 联系及其属性的 E-R 图表示

1.2.2 数据模型

数据模型是对数据的特点、数据之间关系的一种抽象表示,它与具体的数据库管理系统有关,与数据库管理系统支持的数据、联系的表示方式和存储方式有关。

数据模型包括数据结构、数据操作和完整性约束三个部分。数据结构是对数据、数据类型、数据之间的关系的抽象描述。数据操作是对数据模型中各种对象的操作。数据完整性约束是对数据模型中数据的约束规则。其中数据结构是刻画模型性质的最基本的方面。

数据库领域中常用的数据模型是层次模型、网状模型和关系模型,一般将层次模型和网状模型统称为非关系模型。不同的数据模型以不同的方式表达客观世界实体的关系。

本节简要介绍层次模型、网状模型和关系模型的数据结构。

1. 层次模型

层次模型是数据库系统中最早出现的数据模型,层次数据库系统采用层次模型作为数据的组织方式。

层次模型用树形结构来表示各类实体以及实体间的联系。在数据库中,满足下面两个条件的基本层次联系的集合为层次模型:

- 1) 有且只有一个结点没有双亲结点,这个结点称为根结点;
- 2) 根以外的其他结点有且只有一个双亲结点。

在层次模型中,每个结点表示一个记录类型(实体),结点之间的连线表示记录类型间一对多的父子联系。每个记录类型可包含若干个字段(属性),记录类型描述的是实体,字段描述的是实体的属性。

图 1-9 是一个院系的层次数据结构,有四个记录。院系记录是根结点,它有院系编号、院系名称和办公地点 3 个数据项;其两个子女结点是教研室和学生记录;教研室记录是院系的子女结点,它还是教师的双亲结点,教研室记录由教研室编号、教研室名两个数据项组成;学生记录由学号、姓名、年龄 3 个数据项组成;教师记录由职工号、姓名和专业方向 3 个数据项组成。学生与教师是叶结点,它们没有子女结点。在该层次数据结构中,院系与教研室、教研室与教师、院系与学生的联系均是一对多的联系。

层次模型具有一个基本特点:对于任何给定的记录值,只有按其路径查看时,才能显出它的全部意义,没有一个子女记录能够脱离双亲记录值而独立存在。

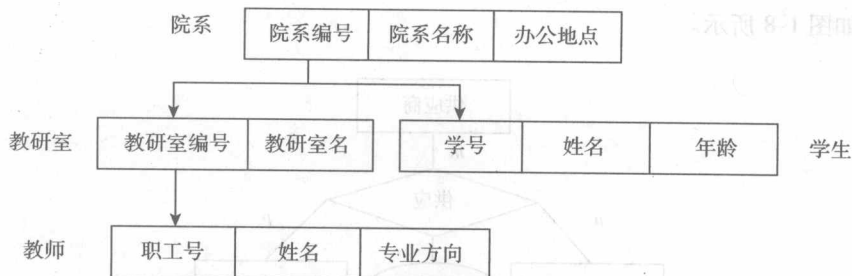


图 1-9 院系的层次数据模型

层次模型适合于表现具有比较规范的层次关系的系统，对这类层次结构清晰的系统，用层次模型描述非常自然、直观、容易理解。但是对于非层次性的，用层次模型就难以表达了。

2. 网状模型

在现实世界中，事物之间的联系更多是非层次关系的，用层次模型表示非树形结构是很不直接的。网状模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限制，允许多个结点没有双亲结点，允许结点有多个双亲结点，此外它还允许两个结点之间有多种联系（称之为复合联系）。因此，网状模型可以更直接地描述现实世界。网状数据库系统采用网状模型作为数据的组织方式。

在数据库中，满足下面两个条件的基本层次联系的集合为网状模型：

1) 允许一个以上的结点无双亲；

2) 一个结点可以有多个双亲。

与层次模型一样，网状模型中每个结点表示一个记录类型（实体），每个记录类型可包含若干个字段（实体的属性），结点间的连线表示记录类型之间一对多的父子联系。

层次模型中子女结点与双亲结点的联系是唯一的，而在网状模型中这种联系可以不唯一。层次模型实际上是网状模型的一个特例。

图 1-10 是一个简单的网状模型，描述学生与课程两个实体间多对多的联系。一个学生可以选修若干门课程，某一课程可以被多个学生选修，因此，学生与课程之间是多对多联系。网状模型不能表示记录之间多对多的联系，为此引进一个学生选课的连接记录，它由三个数据项组成，即学号、课程号和成绩，表示某个学生选修某一门课程及其成绩。这样，学生与课程记录都是选课记录的双亲，通过学生与选课之间的一对多联系及课程与选课之间的一对多联系表示学生与课程之间的多对多联系。

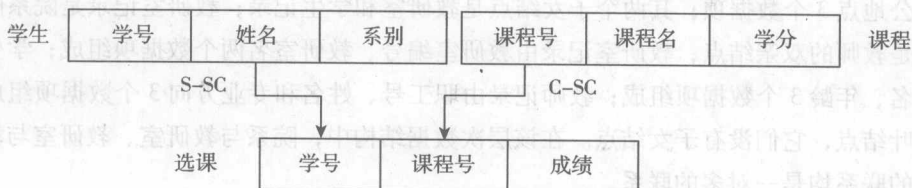


图 1-10 学生/课程/选课的网状数据模型

网状模型能够更为直接地描述现实世界，但是网状模型结构比较复杂，不利于用户掌握和使用。

3. 关系模型

关系模型是目前最重要的一种数据模型。关系数据库系统采用关系模型作为数据的组织方式。

1970 年美国 IBM 公司 San Jose 研究室的研究员 E. F. Codd 首次提出了数据库系统的关系模型, 开创了数据库关系方法和关系数据理论的研究, 为数据库技术奠定了理论基础。

20 世纪 80 年代以来, 计算机厂商新推出的数据库管理系统几乎都支持关系模型, 非关系系统的产品也大都加上了关系接口。数据库领域当前的研究工作也都是以关系方法为基础。

(1) 关系模型的数据结构

关系模型不同于层次模型和网状模型, 它建立在严格的数学概念的基础上。关系模型中数据的逻辑结构是一张二维表, 由行和列组成。如表 1-2、表 1-3 所示即为一个关系模型的数据结构。

表 1-2 学生表

学号	姓名	性别	出生日期	所在系	电话
00001	李丽	女	1980-4-10	D01	56897896
00002	张强	男	1981-2-9	D02	45678912
00003	孙坤	男	1981-10-10	D03	56984512

表 1-3 选课表

学号	课程号	成绩
00001	C01	85
00001	C02	90
00002	C01	80
00003	C02	85

关系模型把数据看成二维表中的元素, 而这个二维表在关系数据库中称为关系。将在下一节中详细讨论关系。

用关系(表格数据)表示实体和实体之间的联系的模型称为关系模型。

在关系模型中, 实体本身以及实体和实体之间的联系都用关系来表示, 实体之间的联系不再通过指针实现。

表 1-2、表 1-3 中学生和选课的联系是靠“学号”字段实现的。

在关系数据库中, 记录值仅构成关系; 关系之间的联系是靠连接字段值表达的。理解关系和连接字段的思想在关系数据库中是非常重要的。例如, 要查询“李丽”的考试情况, 首先要在“学生”关系中得到“李丽”的学号值, 然后根据这个学号值在“选课”关系中找出此学生的所有考试记录值。

对于用户来说, 关系的操作应该是很简单的, 但关系数据库管理系统本身是很复杂的。关系操作之所以对用户很简单, 是因为它把大量的工作交给了数据库管理系统来实现。尽管在层次数据库和网状数据库诞生之时就有了关系模型数据库的设想, 但研制和开发关系数据库管理系统却花费了比人们想象要长得多的时间。关系数据库管理系统真正成为商品并投入使用比层次

数据库和网状数据库晚十几年。但关系数据库管理系统一经投入使用，便显示出了强大的活力和生命力，并逐步取代了层次数据库和网状数据库。现在常见的数据库管理系统几乎都是关系数据库管理系统，比如 Microsoft SQL Server、Oracle、IBM DB2、Access、MySQL 等，都是关系数据库管理系统。

(2) 关系模型的优缺点

关系数据模型具有下列优点：

- 关系模型与非关系模型不同，它是建立在严格的数学概念的基础上的。
- 关系模型的概念单一。无论实体还是实体之间的联系都用关系表示。对数据的检索结果也是关系（即表）。所以，其数据结构简单、清晰，用户易懂易用。
- 关系模型的存取路径对用户透明，从而具有更高的数据独立性、更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作。

所以，关系数据模型诞生以后发展迅速，深受用户的喜爱。

当然，关系数据模型也有缺点，其中最主要的缺点是，由于存取路径对用户透明，查询效率往往不如非关系数据模型。因此，为了提高性能，必须对用户的查询请求进行优化，增加了开发数据库管理系统的难度。

1.3 关系数据库

关系数据库系统是支持关系模型的数据库系统。

1.3.1 关系模型概述

关系数据库中数据的基本结构是二维表（Table），即数据按行、列有规则地排列、组织。数据库中每个表都有唯一的表名。

下面介绍关系模型中的基本术语。

关系（Relation）：一个关系在逻辑上对应一个按行、列排列的二维表，表名就是关系名，如表 1-2 所示。

属性（Attribute）：表中的一列称为一个属性（或称一个字段），每个属性起一个名字，即为属性名，如表 1-2 所示，六列表示六个属性（学号、姓名、性别、出生日期、所在系、电话）。

域（Domain）：属性的取值范围称为域。例如，性别的域是男、女；所在系的域是学校所有系的集合。

元组（Tuple）：表中的每一行称为元组，又称记录（Record）。如在表 1-2 的“学生”关系中，元组有：

- (00001, 李丽, 女, 1980-4-10, D01, 56897896)
- (00002, 张强, 男, 1981-2-9, D02, 45678912)
- (00003, 孙坤, 男, 1981-10-10, D03, 56984512)

候选码（Candidate Key）：如果一个属性或属性集的值能够唯一标识一个关系的元组，又不包含多余的属性，则称该属性或属性集为候选码，又称候选关键字或候选键。在一个关系上可以有多个候选码。