

普通高校系列教材·信息技术

数据库原理及应用

—FoxPro for Windows

普通高校系列教材(信息技术)编委会组编

郑德忠 编

邹华跃 审

学习参考



11.13-42

南京大学出版社

内
容
简
介

本书是普通高校系列教材《数据库原理及应用——FoxPro for Windows》的配套辅导材料。本书共分为五大部分:第一部分是学习方法指导,结合课程的特点,概括出各章的基本内容、重点难点;第二部分是习题解答,详尽地分析了教材中的所有习题,并给出正确答案;第三部分是典型题解与分析,给出了大量具有代表性的例题;另外在本书的第四部分有针对性地给出两套模拟测试卷,使读者在学完整个课程后进行自我测试和检查。为了培养大家的实际应用能力,该书的第五部分安排了四个具有代表性的实验。通过这些实验,使读者能够举一反三,提高设计应用程序的能力。

本书可作为大、中专学校学生学习《数据库原理及应用——FoxPro for Windows》首选辅导书。

图书在版编目(CIP)数据

《数据库原理及应用——FoxPro for Windows》学习参考/郑德忠编. —南京:南京大学出版社,2003.5

ISBN 7-305-02676-X

I.数... II.郑... III.关系数据库—数据库管理系统,FoxPro—高等教育—教学参考资料 IV.TP311.138

中国版本图书馆 CIP 数据核字(2003)第 038040 号

书 名 《数据库原理及应用——FoxPro for Windows》学习参考
编 者 郑德忠
出版发行 南京大学出版社
社 址 南京市汉口路 22 号 邮编 210093
电 话 025-3596923 025-3592317 传真 025-3303347
网 址 www.njupress.com
电子函件 njupress1@public1.ptt.js.cn
经 销 全国新华书店
印 刷 合肥学苑印刷厂
开 本 787×1092 1/16 印张:9.375 字数:225 千
版 次 2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷
书 号 ISBN 7-305-02676-X/TP·99
定 价 11.00 元

* 版权所有,侵权必究。

* 凡购买南大版图书,如有印装质量问题,请与所购图书销售部门联系调换。

普通高校系列教材(信息技术)编委会

主 任:孙钟秀 中国科学院院士

副主任:张福炎 南京大学教授

陈国良 中国科学技术大学教授

钱洲胜 中国计算机函授学院教授

委 员(按姓氏笔画排序):

于学锋	中国计算机函授学院	王佩珠	西安交通大学
王文兰	桂林电子工业学院	王蔚韬	重庆建筑大学
伍良富	成都电子科技大学	成松林	东南大学
刘存书	郑州信息工程大学	朱大奇	安徽工业大学
朱宝长	西安电子科技大学	孙德文	上海交通大学
杜象元	上海交通大学	李茂青	厦门大学
李学干	西安电子科技大学	杨来利	兰州大学
何淑兰	北京科技大学	张凤祥	华中科技大学
张绍林	河北行政学院	张维勇	合肥工业大学
张民坤	云南工业大学	张景书	哈尔滨工程大学
赵良全	新疆大学	皇甫正贤	东南大学
洪志全	成都理工学院	姚君遗	合肥工业大学
高 平	浙江大学	陶世群	山西大学
曹翊旺	湖南省计算机高等专科学校	梁文康	山东大学
韩国强	华南理工大学	舒 洪	南昌大学
葛 燕	中国科学技术大学	解世耀	辽宁大学
谭耀铭	南京大学	黎庆国	合肥工业大学

出版前言

近些年来,我国的高等教育事业有了长足的发展,高校招生人数年年递增,越来越多的年轻人有机会接受正规的高等教育,这一举措无疑对我国的社会进步和经济发展有着重要的意义。但是人们也深刻的认识到,高等教育质量的好坏是一个不容忽视的关键性问题,而保证教育质量的一个重要环节就是抓好教材建设,而教材内容陈旧、教学手段落后的现象一直存在着。尤其像计算机技术这样的新兴领域发展迅猛、知识更新日新月异、教学内容落后于新技术新知识的矛盾显得尤为突出。基于上述两方面考虑,在南京大学出版社的鼎力相助下,一个以组编高校信息、电子类专业教材为主要任务的教材编委会成立了。

针对我国高等教育的现状和信息、电子技术的发展趋势,编委会组织部分高校的专家教授进行了深入的专题研讨。大家一致认为,在当前情况下组编一套紧跟新技术发展、符合高校教学需要、满足大学生求知欲望的系列教材势在必行,这将有助于规范教学体系、更新教学内容、把握教学质量、培养合格人才。专家们还对教材的结构、内容、体例及配套服务等提出了具体要求。为了能使这套教材逐步完善,并促进全国各地高校教学质量的提高,编委会决定在教材之外认真做好三件事:第一,为每本教材配备一本供学生使用的学习参考书,其主体内容为学习方法指导、习题分析与解答、典型题解或课程设计、模拟测试卷及解答、实验指导书;第二,定期对教材内容进行修订,及时补充新技术新知识,并根据具体情况组编新的教材;第三,有计划地组织各地高校教师进行教学交流与研讨,通过这种途径来提高偏远地区的师资水平。我们相信,通过各方面的大力支持和大家的不懈努力,这套教材会逐步被广大师生所接受,并在使用过程中得到完善、充实。

大家都知道,组编这样一套系列教材是个牵涉面很广的大工程。这个工程不仅在起步阶段需要得到各级教育主管部门、各高等院校、出版社的大力支持和协助,而且在使用过程中也离不开各位专家、教授、学生的热心呵护和指导。因此,殷切期待所有的能人志士关心我们,帮助我们,向我们提出好的建议或意见,为我们指出教材中的不足之处。

最后,感谢所有为本套系列教材出版付出辛勤劳动的同志们。

普通高校系列教材(信息技术)编委会

2003年3月

编 者 的 话

随着计算机技术在各个领域的推广普及和应用,数据库管理技术正发挥着越来越重要的作用。数据库应用既有完整的理论体系,又有很强的实践性,是一门非常实用的技术,但学习掌握起来有一定的难度。

为了使大家更好地掌握数据库理论知识与实用技巧,我们编写了本书。该书是教材《数据库原理及应用——FoxPro for Windows》的配套辅导教材,适合于在学完课程每章基本内容后复习时使用。本书内容共分为五个部分:第一部分是学习方法指导,内容包括各章节基本内容的提要、重点难点的讲解、学习的思路和方法等,目的是通过章节讲解,使读者既掌握教材的重点和难点,又能把握学习的思路和方法;第二部分为习题解答,该部分详细地给出了《数据库及应用 FoxPro for Windows》教材中各章后的习题答案,通过解答书后的习题,读者可进一步巩固所学的知识,增强编程能力;第三部分是典型题与分析,这部分是根据原教材的内容,提炼出具有代表性的题目,并加以分析和解答,充分展示了教材内容的主体,以此培养读者分析问题、解决问题的能力;第四部分是模拟测试,这部分内容综合整个教材,总结性地给出两套模拟测试卷,其广度和深度适中,供学完整个课程后进行自我测试和检查;第五部分是实验指导,共安排了四个具有代表性的实验题目,从实验名称、实验要求、实验设备、预习要求、实验步骤、讨论等六个方面进行描述,以此帮助读者举一反三,培养实际应用能力,提高应用程序的设计水平。

本书内容较为全面,指导性较强,既是《数据库及其应用 FoxPro for Windows》课程的学习参考教材,也可作为数据库应用学习人员的辅导材料。

由于编者水平有限,难免有不妥或疏漏之处,请批评指正和提出宝贵意见。

编 者

2003 年 3 月

目 录

第一部分 学习方法指导	(1)
一、课程的性质与任务	(1)
二、课程的基本内容及学习要点	(1)
第二部分 习题解答	(46)
第1章习题解答	(46)
第2章习题解答	(47)
第3章习题解答	(49)
第4章习题解答	(50)
第5章习题解答	(51)
第6章习题解答	(52)
第7章习题解答	(52)
第8章习题解答	(52)
第9章习题解答	(52)
第10章习题解答	(55)
第11章习题解答	(56)
第12章习题解答	(62)
第三部分 典型题解与分析	(63)
第1章	(63)
第2章	(64)
第3章	(67)
第4章	(71)
第5章	(72)
第6章	(76)
第7章	(79)
第8章	(84)
第9章	(88)

第 10 章	(103)
第 11 章	(105)
第 12 章	(110)
第四部分 模拟测试	(112)
模拟试卷(一)	(112)
模拟试卷(一)分析与解答	(116)
模拟试卷(二)	(123)
模拟试卷(二)分析与解答	(128)
第五部分 实验指导	(134)
实验一 数据库的建立和数据记录的输入	(134)
实验二 数据库的使用	(136)
实验三 数据库应用程序设计(I)	(138)
实验四 数据库应用程序设计(II)	(139)

第一部分 学习方法指导

对《数据库原理及应用——FoxPro for Windows》课程的学习,应该领会数据库系统的基本理论和设计思想,掌握操作管理数据库的方法,不断实践,将管理数据库和设计程序应用到实际工作中,只有这样,才能真正掌握这门课。本部分将根据《数据库原理及应用——FoxPro for Windows》课程的性质,理出学习思路和方法。在本书的后几部分中,给出了课程习题解答、典型题解、模拟测试和实验指导等,旨在提高大家的学习效率。

一、课程的性质与任务

《数据库原理及应用——FoxPro for Windows》课程中既阐述数据库系统的基本理论、设计思想方法,又详细介绍对库文件的操作管理及应用程序设计等。本课程的实际应用性很强,尽管教材中数据库系统理论部分介绍了不少的概念和数据库设计思想,但这些都是为后面的程序设计服务的。因为理解了这些概念和设计思想,将对后面的数据库应用管理和程序设计有着重要的指导意义,从而使读者能在一个较高的境界中去进行数据库管理和应用程序设计。当然,在设计应用程序时,还有一些经验和技巧。

本课程的任务是,学习和理解数据库基本理论与设计思想,在基本理论和思想的指导下能顺利进行数据库的管理和应用程序设计。教材的一开始介绍了较多概念和理论,这些对于初学者来说是很抽象的。因此,对于本课程的学习,可以先学习第5章至第8章的内容,当对数据库有了一定的感性认识和操作管理实践经验后,再返回来学习领会第1章至第4章的内容。这样,从学习的角度来说可能更好接受一些。此外,在数据库理论部分用到了少量的“离散数学”和“软件工程”的知识,在教材的引用之处均从最基本的概念讲起,不会影响本课程的学习。当理解了数据库理论,经过一定的数据库管理实践后,然后去进行应用程序设计,将会更加得心应手。

二、课程的基本内容及学习要点

下面将结合教材中每章节的基本内容,总结性地给出学习的重点及难点,并对其进行讲解。

1. 数据库基本知识

教材第1章介绍了数据、信息、数据处理、数据库、数据库系统、DBMS等一系列名词和术语,目的是建立起数据库系统的基本概念。该章的重点是数据库系统的组成,DBMS的功能,数据库的保护;难点为理解数据库的体系结构。理解并区分各个名词的概念,是进一步学习数据库课程的基础。

(1) 数据、信息及数据管理

数据在计算机中是一个具有丰富内涵的概念。凡是计算机能识别和接受的东西都叫数据,如文字、符号、图形、图像、声音等。数据是数据管理的对象。信息是指经过整理或处理,能够反映一定情况和意义的数据。数据与信息既有联系又有区别,数据表示了信息,而信息只有通过数据形式表示出来才能被人理解和接受。数据管理是指数据的收集、整理、组织、存储、维护、检索、传送等操作。对于各方面的数据管理均可以考虑研制出一个通用、高效而又使用方便的管理软件,把数据有效地管理起来,这正是人们研究和使用数据库的目的。

(2) 数据库处理技术发展阶段

用计算机对大量数据进行管理,既方便又快捷。从计算机数据管理技术的发展历程看,大体经历了三个发展阶段:自由管理方式阶段、文件管理方式阶段、数据库方式阶段。

自由管理方式是较低级的计算机管理数据的方式,这种方式迫使程序员直接与存储设备打交道,导致程序与数据紧密相关,程序高度依赖于数据,数据稍有变动,整个程序就必须全部进行修改,编程效率低,程序不灵活而且容易出错。文件管理方式,是把有待于加工处理的数据单独组织成数据文件,在程序中可以通过文件名间接地存取文件中的数据,解脱了程序员直接与存储设备打交道的沉重负担,程序中不再涉及任何具体细节。文件管理方式,是数据管理的一大进步。但是在对相关内容的若干个数据文件的处理中,文件管理方式就显得累赘。针对文件方式的弊端,后来出现了数据库技术。数据库技术的目标,首先是克服程序与文件相互依存,力求数据独立,其次重在表现数据之间的联系,还要尽量克服数据冗余,解决数据安全性和完整性的保护问题。通过以上的比较可以看出,数据库管理方式是管理数据的最佳方式。

(3) 数据库系统组成

数据库就像存放相关数据的仓库,从其整个运行的环境来说,它实际上是个系统,由数据库本身、数据库管理系统软件和支持数据库运行的软、硬件环境以及应用程序和数据库管理员等相互独立而又相互联系的若干部分所组成。

在此有几个名词需要掌握:数据库、数据库管理系统、支持数据库运行的软硬件环境、应用程序、数据库管理员。数据库是一个有一定结构的相关数据集合,它是相对独立的,是数据处理的核心和管理对象。数据库管理系统(简称 DBMS),它是管理、控制数据库的软件,提供数据检索、存取、数据控制和数据库维护等功能。支持数据库运行的软硬件环境是指数据库赖以存在的物理设备、系统软件操作系统、汉字环境等。应用程序是用户针对数据库管理设计的程序。数据库管理员(简称 DBA),是数据库系统中人的因素。DBA 的职责是对数据库进行管理和维护,保持数据库系统始终处于最佳工作状态。

(4) 数据库保护

数据库保护是指保护数据库的数据安全以及数据完整。数据库管理系统中有专门采取的对策进行各种各样的数据保护,大致包括安全性保护、完整性保护、故障恢复等。

2. 数据库系统基本理论

教材第2章介绍了数据模型、实体、属性等概念,着重介绍了在应用领域中最为流行的关系数据模型,阐明用E-R图构造关系数据模型的方法。该章的重点在于理解数据模型,实体、属性及其型与值,实体间三种联系与表示法,E-R图的使用;难点在于如何用关系表示实体间的联系,怎样用E-R图构造关系数据模型。

(1) 数据模型

数据模型是对数据库如何组织的一种模型化表示。数据模型的关键在于数据库不仅要表示存储些什么信息,更重要的是要以一定的数据结构方式表示各种信息的联系。

为了便于理解,可以把数据模型形式化表示为:

$$DM = \{R, L\}$$

其中DM(Data Model)是数据模型英文简称,R是记录型的集合(每个记录型表示一种信息),L是不同记录型联系的集合。

需要注意的是,数据模型重在数据结构,而不是具体装了什么数据,所以数据模型表示的是数据库的框架。

(2) 实体和属性

实体是现实世界任何可被识别事物的抽象命名,不论是有形的还是无形的。例如,一个人、一项设计、一个规划等,都叫实体。任何实体,都可以统一地用属性来描述它,所以属性是事物性质的抽象。例如,在数据库中,属性是描述一系列指定的字段。实体和属性,有所谓型与值之分。型是结构,即数据库的字段;值是在结构约束下的取值,即字段内容。

(3) 实体间的联系

1) 联系方式

实体间的联系情况比较复杂,就联系方式而言,有以下三种:

- 一对一联系(简记 1:1)
- 一对多联系(简记 1:n)
- 多对多联系(简记 m:n)

① 1:1 联系。

若两个不同型实体集合中,任一方的一个实体最多与另一方的一个实体相对应,称 1:1 联系。

② 1:n 联系。

若两个不同型实体集合中,一方一个实体对应另一方的若干实体;反之,另一方的一个实体最多对应本方一个实体,称 1:n 联系。

③ m:n 联系。

若两个不同型实体集合中,两实体集中任一实体,都与另一实体集中一个或多个实体相对应,称 m:n 联系。

以上三种联系方式中,基本的是 1:n 联系,因为 1:1 包含 1:1;而 m:n 可转换成为 1:n。

2) 实体联系的表示方法

从宏观上指出“型”间的联系,然后注明联系方式,这样既简单又能反映实体间的联系情况。具体方法是:

①实体集合用方框代表,方框内填上实体名或者填上标识实体的关键字,也可以具体填上实体的各个属性名。

②联系用有向线段表示。约定:“单”箭头指向有联系的“1”的一方实体,“双”箭头指向有联系的“多”的一方实体。

根据上述约定,图 1-1 所示的是不同型实体间的联系情况。

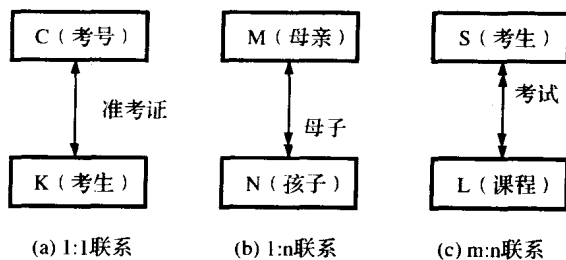


图 1-1 实体联系方式表示法

(4) 数据库类型

DBMS 有不同的类型,是根据反映实体联系方式的数据模型来划分的。目前成熟地应用在数据库技术中的数据模型有:层次模型、网状模型、关系模型。层次模型、网状模型统称为格式化模型。关系模型是用关系表示实体及其联系,建立在严格理论基础上的,是本书的重点。

层次模型,实体及其联系表现为树结构;网状模型,结点间的联系不受层次限制,可以任意发生联系。所谓关系模型就是指用若干关系表示实体及其联系。关系是个数学名词,有严格的数学定义,可以把关系理解成二维表格。一个关系应包括关系名和属性名,前者相当于二维表名,后者相当于二维表头栏目名。

关系模型,有着一体化的特点,不论是实体还是实体间的联系,统一用关系来表示,并且能直接反映各种联系方式,数据表现力强,而且符合人们的习惯,对关系的操作都可以直观地从二维表去理解,又有数学理论为基础,所以现代微型机数据库大多数是关系式的。

(5) E-R 方法介绍

1) E-R 图的基本元素及画法

E-R 方法就是通过所谓的“E-R 图”表示实体及其联系。E-R 图中包括:实体、属性和联系三种基本图素。约定实体用方框表示,属性用扁圆框表示,联系用菱形框表示,并在其内部填上实体名、属性名和联系名,如图 1-2 所示。

E-R 图的具体画法是:把有联系的实体(方框)通过联系(菱形框)连接起来,注明联系方式,实体的属性(扁圆框)连到相应实体上。

2) E-R 图的几种基本形式

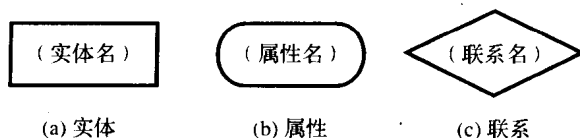


图 1-2 E-R 图基本图素

由于 E-R 图直观易懂,在概念上表示了一个数据库的信息组织情况,所以若能画出 E-R 图,意味着彻底搞清了问题,此后便可根据 E-R 图,结合具体 DBMS 的类型,把它演变为 DBMS 所能支持的数据模型,这种逐步推进的方法如今已普遍用于数据库设计中,成为数据库设计中的一个重要步骤。可以将 E-R 图归结为图 1-3 所示的几种基本形式:

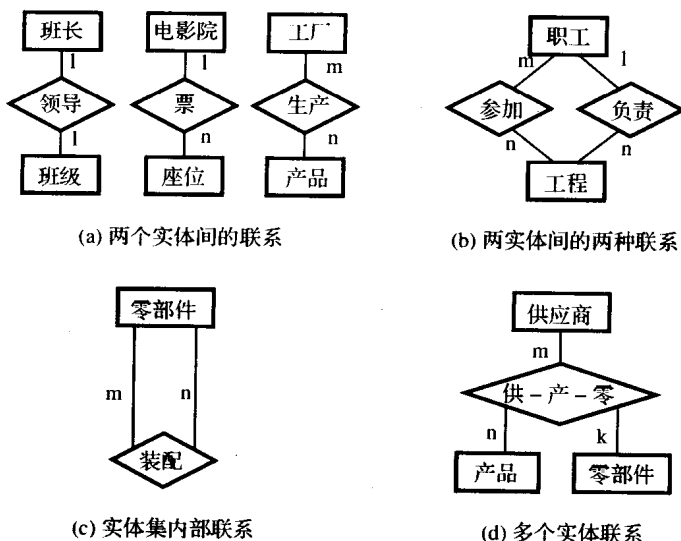


图 1-3 基本联系方式

①两实体间的二元联系,如图 1-3(a)所示,有三种联系方式。

②两实体间的两种以上不同的联系,如图 1-3(b)所示。它表示“职工”与“工程”两实体之间,就“职工”参加“工程”施工而言是多对多的,即一个职工可以参加多项工程建设,反过来一个工程又由很多职工参与施工;而就工程负责人而言,是一对多的,即某一职工可以负责多项工程,反过来一项工程只能由一个职工负责。

③同一实体集内部个体间的二元联系,如图 1-3(c)所示。它表示一个零部件可能由多个其它零部件装配而成,而该零部件又可装配多种其它零部件。

④两实体以上的多元联系,如图 1-3(d)所示。它表示工厂生产的产品向哪个供应商采购零部件用于装配什么产品,其中假定相互间为多对多的联系。

3) 设计 E-R 图的原则

具体设计 E-R 图时,应遵循以下原则:

①首先针对特定用户的应用,确定实体、属性和实体间的联系,作出反映该用户视图的局部 E-R 图。注意,实体和属性是相对而言的,往往一个事物在一种应用环境中可能作为属性,而在另一种环境中却可能以实体出现。例如,“教室”可以作为实体“课程表”的属性,

表示某门课程的上课地点;而在另一种环境下,若需考察“教室”的面积、座位数、设施情况等,又需把它作为实体。原则上能够作为属性就不要作为实体,这有利于 E-R 图的简化。基本原则是属性不能再具有需要描述的性质或与其它事物的联系。

②综合各个用户的局部 E-R 图,产生反映数据库整体概念的总体 E-R 图。在综合时,出现在不同 E-R 图中的同名实体,只能在总体 E-R 图中出现一次,以便消除冗余。要消除那些同名异义或同义异名的现象,以保持数据一致性。具体地说,如果同一事物在不同局部 E-R 图中作为实体而名字又不一致,那就应该把名字统一起来;如果不同事物在不同局部 E-R 图中作为实体而名字又相同,就得重新命名加以区别。实体间的联系也有类似情况,亦应进行类似处理。而在属性这一级,同名异义或同义异名现象更是经常发生。例如,产品编号在不同局部 E-R 图中,可能会有“产品代号”、“产品编号”、“产品代码”等各种名字,这时必须加以统一,消除不一致性。在综合局部 E-R 图时,要注意消除那些冗余的联系,冗余信息会影响完整性,使维护工作变得复杂化,但这要折衷考虑,因为有时必要的冗余对提高数据处理效率有利。也允许在总体 E-R 图中增加新的联系。

总的来说,生成的总体 E-R 图,必须满足以下两点:

第一,能准确地反映原来的局部 E-R 图,包括属性、实体及相互联系。

第二,整体概念一致性,不能存在互相矛盾的表达。

值得注意的是:一个系统的 E-R 图不是唯一的,强调不同侧面和不同联系,作出的 E-R 图可能有很大不同。

4)从 E-R 图导出关系数据模型

遵循以上原则所设计出来的 E-R 图仅是现实世界的纯粹反映,与数据库具体实现毫无关系,但它却是建立数据模型的基础。从 E-R 图出发导出具体 DBMS 所能接受的数据模型,这步工作在数据库设计中称为逻辑设计。教材中着重介绍了关系模型,所以大家要重点掌握由 E-R 图转换为关系式数据模型的方法,即把 E-R 图转换为一个个关系框架,使之相互联系构成一个整体结构化了的数据模型。这里的关键问题是怎样实现不同关系的联系,具体方法可按以下原则进行:

①E-R 图中的每个实体都相应地转换为一个关系,该关系应包括对应实体的全部属性,并应根据该关系表达的语义确定出关键字,因为关系中的关键字属性是实现不同关系联系的主要手段。

②对于 E-R 图中的联系,要根据联系方式的不同,采取不同手段以使被它联系的实体所对应的关系彼此实现某种联系。具体方法是:

- 如果两实体间是 1:n 联系,就将“1”的一方关键字纳入“n”方实体对应的关系中作为外部关键字,同时把联系的属性也一并纳入“n”方对应的关系中。

- 如果同一实体内部存在 1:n 联系,可在这个实体的所对应的关系中多设一个属性,用来表示与该个体相联系的上级个体的关键字。

- 如果两实体间是 m:n 联系,则需对联系单独建立一个关系,用来联系双方实体,该关系的属性中至少要包括被它所联系的双方实体的关键字,并且如果联系有属性,也要归入这个关系中。

- 如果同一实体内部存在 m:n 联系,也需为这个联系单独建立一个关系,该关系中至少应包括被它联系的双方个体的关键字,如果联系有属性,也要归入这个关系中。

• 对于两个以上实体 $m:n$ 的多元联系,也需为联系单独建立一个关系,该关系中最少应包括被它所联系的各个实体关键字,若是联系有属性,也要归入这个关系中。这个情况与两实体间 $m:n$ 联系类似。

3. 关系数据库概论

教材第3章从数学意义上引出了对关系数据库中关系的定义,阐明了关系的性质,对关系数据库操作语言 DML 进行较全面的介绍,阐述建立关系数据库应遵循的规则,即关系规范化的指导思想。该章的重点是关系、关系与二维表、DML 及其特点、关系规范化等有关概念,以及关系规范化的方法与原则,难点是理解关系的定义、关系规范化的方法。

(1) 关系在数学意义上的定义

关系的数学概念可从日常生活上引出。假定:

$M = \{\text{王强, 张伟, 戈华}\}$ 是男性集合

$W = \{\text{李丽, 刘英}\}$ 是女性集合

若 M 和 W 中存在夫妻关系时,那么可能的配对如下:

$M \times W = \{(\text{王强, 李丽}), (\text{王强, 刘英}), (\text{张伟, 李丽}), (\text{张伟, 刘英}), (\text{戈华, 李丽}), (\text{戈华, 刘英})\}$

在数学上把所有这些可能的配对叫“笛卡尔积”,直观意义是诸集合各元素间一切可能的组合。笛卡尔积可看成是一个二维表。

(2) 关系的性质

关系表现为二维表格,因此可以通过二维表格研究关系的性质,具体性质如下:

- ①不允许“表中套表”,即表中元组分量下不可再有分量。
- ②表中各列取自同一个域,因此一列中的各个分量具有相同性质。
- ③列的次序可以任意交换,不改变关系的实际意义。
- ④表中的行叫元组,代表一个实体,表中不允许出现相同的两行。
- ⑤行的次序无关紧要,可以任意交换,也不会改变关系的意义。

根据上述性质,则判断两个关系是否相等,与属性次序无关,与元组次序也无关,与关系的命名也无关,如果仅仅是上述差别,其余完全相同,则认为这是相等的两个关系。

(3) 关系数据操作语言 DML

DML 是应用程序与数据库之间的接口,因此一个好的数据模型,还需要辅以强有力的 DML 供用户操作数据库。DML 特点如下:

①操作对象与结果——关系。因关系可看成元组集合,所以集合运算完全适用于关系。关系 DML,以关系为处理单位,处理后的结果也是关系,这是关系数据操作的最大特点,也是它的优点,使用简捷便当,最适宜处理有规律性的业务信息。

②非过程性强。所谓过程与非过程,以“做什么”和“怎么做”两尺度来衡量。若用户只需提出做什么,而把怎么做的问题交给系统去解决,则称是非过程化的;反之,若需用户编程解决怎么去做,则称是过程化的。从这一角度讲,关系数据操作非过程性强,用户只需告知

系统进行什么操作,勿需指出存取路径,不必设计处理步骤。

③语言一体化。关系式 DML 与 DDL 通常溶于一体,在应用程序中允许出现 DDL 语句,这意味着用户在执行程序过程中,可以根据需要,修改已建立好的属于用户本身的数据模式,并能把它作为模式一部分纳入数据库概念模式中。

④有严密的数学工具。关系数据库上的操作包括:查询、统计和汇总、插入、删除、修改等。所有这些操作中,查询是核心,因为几乎所有别的操作均以查询为前提,所以关系 DML 语言,常被称为查询语言。

在数据库中为了实现数据查询,定义了一组专门的关系运算,包括选取、投影和联接。选取是单目运算,即对一个关系施加的运算,按给定条件从关系中挑选满足条件的元组组成的集合;投影是单目运算,从关系中挑选出指定的属性组成的新关系;联接是双目运算,按给定条件把满足条件的两个关系所有元组,按一切可能拼接后形成的新关系。

(4)关系规范化的实现方法

1)范式和规范化

关系规范化就是研究关系数据库设计中应遵循的原则,以求数据库稳定而又灵活,使数据库好用。在弄懂关系规范化的实现方法之前,必须先弄懂以下几个名词:

关键字:在实体属性中可以用来标识实体集中不同个体的属性。

函数依赖:函数依赖是数学中用的名词,在关系中是用来刻划关系中各个属性之间相互制约而又相互依赖的关系。

范式:简称 NF。要想设计一个好的关系,必须使关系满足一定约束条件,此约束已经形成了规范,分成几个等级,一级比一级要求的严格。满足最低要求的关系称它属于第一范式,在此基础上又满足了某种条件,达到第二范式标准,则称它属于第二范式的关系,如此等等,直到第五范式,显然满足较高条件者必满足较低范式条件。一般情况下,1NF 和 2NF 的关系存在许多缺点,实际的关系数据库一般使用 3NF 以上的关系,但也不是范式越高越好,这取决于应用场合。

关系规范化的实现步骤:

①1NF。

如果关系 R 的所有属性都是不可再分的数据项,则称该关系属于第一范式,记作 $R \in 1NF$ 。1NF 的关系,条件很松,只要满足关系基本定义,不出现表中套表,即满足了条件。1NF 的关系常常存在数据冗余大,修改麻烦,插入异常和删除异常诸多弊端,所以 1NF 的关系不是理想的关系。

②2NF。

若关系 $R \in 1NF$,且它的每一非主属性都完全依赖于关键字,则称 R 属于第二范式关系,记为 $R \in 2NF$ 。

③3NF。

若 $R \in 2NF$,且它的每一非主属性都不传递依赖于关键字,则称 R 属于第三范式关系,记为 $R \in 3NF$ 。

④BCNF。

满足 3NF 的关系数据库,一般情况下能达到满意的效果,但是在某些情况下,3NF 的关

系还会出现麻烦,则出现 BCNF 的定义。

设 R 是一个规范化的关系(不限 R 是第几范式),如果 R 的每一个决定因素项都是候选关键字,则称 R 属于 BCNF 关系,记 $R \in \text{BCNF}$ 。判定一个关系是否属于 BCNF,可以从三个方面来看,即:确定有哪些候选关键字;确定有哪些决定因素;根据定义,每一决定因素都应是候选关键字。

另外,BCNF 并没有规定必须从 3NF 关系出发,即不必遵循 $1\text{NF} \rightarrow 2\text{NF} \rightarrow 3\text{NF}$ 步骤,逐步消除部分依赖和传递依赖,可以直接从 1NF 起步根据定义判定。因 BCNF 约束,除候选关键字外别无其它决定因素,即限定了非主属性无任何决定因素,故 BCNF 排除任何对主关键字的部分与传递依赖属性,所以 $\text{BCNF} \in 3\text{NF}$ 。

2) 分解关系的基本原则

关系规范化过程实质是对关系不断分解的过程,通过分解使关系逐步达到较高范式。但是,分解方法不是唯一的,不同分解方案性能差别往往相当大。分解关系必须遵循以下两个原则:

①分解必须是无损的:即分解后不应丢失信息。

②分解后的关系要相互独立:即避免对一个关系的修改,波及到另一关系。

关系规范化理论很重要,直接触及怎样设计关系,是关系数据库设计的理论依据。下面指出其要点:

①规范化过程是对关系逐步分解过程

利用关系投影运算,每一次分解,都使关系从较低级范式向高级范式变换,这种变换是围绕函数依赖进行的。各范式间的关系如下式所示,是一种全包含关系:

$$5\text{NF} \subseteq 4\text{NF} \subseteq \text{BCNF} \subseteq 3\text{NF} \subseteq 2\text{NF} \subseteq 1\text{NF}$$

②具体对关系规范化时应考虑以下问题:

- 确定关系的各个属性中,哪些是主属性,哪些是非主属性。
- 确定所有的候选关键字。
- 选定主关键字。
- 找出属性间的函数依赖关系。
- 必要时,画出函数依赖图,可采用图示法或直接在关系模式上表示函数依赖关系。
- 根据应用特点,确定规范化到第几范式。
- 分解关系,分解必须是无损的,不得丢失信息。
- 分解后的关系,力求相互独立,即对一个关系内容的修改不要影响到分解出来的别的关系。

值得提出的是,不是范式越高越好。例如,当应用程序着重于查询而很少涉及插入、更新和删除操作,那么宁愿采用较低范式以获得较好的响应速度,因为范式高,关系所表达的信息单纯,查询数据就得作联接运算,而这开销是很大的。

4. 数据库设计

教材第 4 章介绍了数据库设计的内容及特点,数据库逻辑设计和物理设计的步骤, E-R 模型的建立,数据库应用程序投入运行及其维护等事项。该章的重点是数据库设计的内容和步骤、数据流图的画法、E-R 模型的建立、数据库的逻辑设计和物理设计、数据库的

运行和维护方法,难点是数据流图的设计、E-R模型的设计及优化。

(1)数据库设计概述

数据库设计包括结构特性和行为特性的设计。结构特性的设计是指确定数据库的框架和数据结构。行为特性的设计是指确定数据库用户的行为和动作,也就是通过应用程序存取数据和处理数据。从使用方便和改善性能角度考虑,结构特性必须适应行为特性。在数据库设计中,结构特性设计是关键,因为数据库结构框架正是从考察用户的行为所涉及到的数据处理进行汇总和提炼出来的。数据库设计是一种“反复探寻,逐步求精”的过程。

(2)数据库逻辑设计

逻辑设计通常划分成三个设计阶段步骤,每个阶段又可划分若干设计步骤,前一步骤的结果作为后续设计步骤的依据,这样逐步向前推进,最终生成数据库设计的逻辑模式,如图1-4所示。

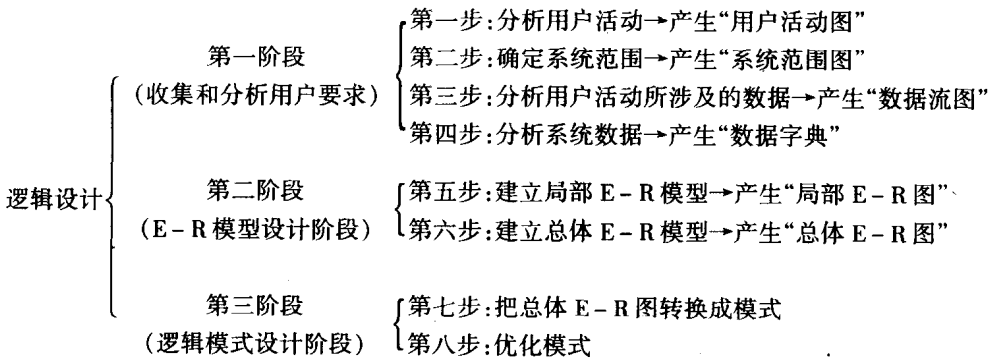


图 1-4 逻辑设计步骤

其中,第一阶段设计任务是收集和分析用户要求,也就是首先要分析处理问题的要求,着手搞清处理流程;画出用户活动图,确定属于系统的处理范围;深入分析用户的业务处理,特别关注用户活动图所处理的数据,并以“数据流图”形式表示出数据的流向和对数据所进行的加工;对数据流图中的每个数据流名、每个文件名、每个加工名,都要给出具体定义,都需要用一个条目进行描述。在第一阶段中,数据流图的作用是从“数据”和“对数据的加工”两方面表达数据处理系统工作过程,画数据流图时应遵循“由外向内,自顶而下”的原则。由外向内是指:先标定系统范围,这个范围就是输入输出之间的部分,该部分的细节暂不考虑,将其当成一个黑箱子。在画数据流图时,往往最外部难以表示出全部数据流,但这不要紧,因为无法表示的内部数据流在以后实现内部各种应用的数据流图时会发现并填补上去。描述系统内部数据流,一般从输入端开始向输出端推进,每当经过使数据流的组成或数值发生变化的地方,就用一个“加工”将其连接起来,这个“加工”正是实现这一数据变化的。注意,不要把相互无关的数据画成一个数据流,也不要作为作为一个处理单位的数据画成二个数据流。

第二阶段设计任务是建立 E-R 模型。经过第一阶段的分析用户要求,已经产生了各种应用的数据流图,它是设计 E-R 模型的依据。第二阶段首先要进行局部 E-R 模型的设计,它是针对具体应用建立的,可从数据流图出发确定实体和属性,并根据数据流图中表示