

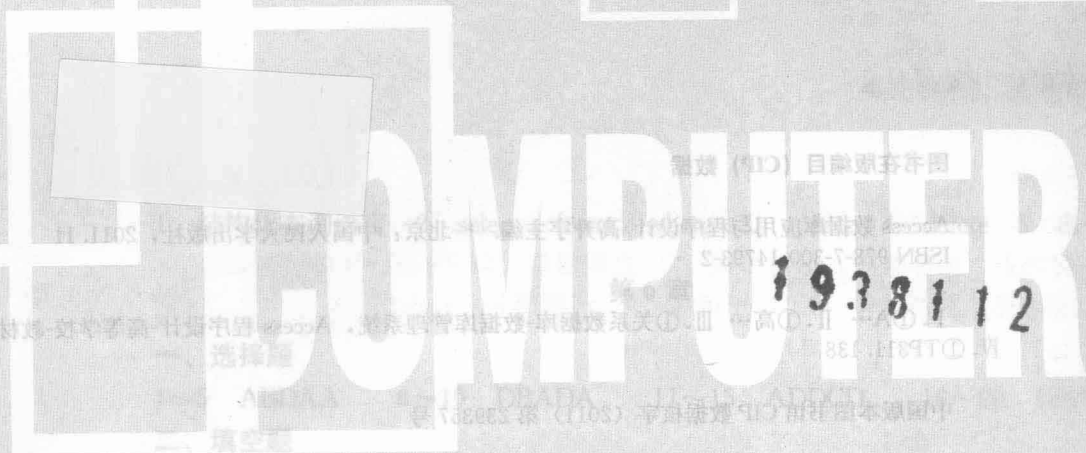
COMPUTER

Access 数据库应用与 程序设计

主 编/高升宇

主 审/张郭军





Access 数据库应用与 程序设计

主 编/高升宇

副主编/奚建荣 索红军

习贵民 刘龙飞

主 审/张郭军



中国人民大学出版社

· 北京 ·

图书在版编目 (CIP) 数据

Access 数据库应用与程序设计/高升宇主编. —北京: 中国人民大学出版社, 2011. 11
ISBN 978-7-300-14793-2

I. ①A… II. ①高… III. ①关系数据库-数据库管理系统, Access-程序设计-高等学校-教材
IV. ①TP311. 138

中国版本图书馆 CIP 数据核字 (2011) 第 239357 号

Access 数据库应用与程序设计

主编 高升宇

主审 张郭军

Access Shujuku Yingyong yu Chengxu Sheji

出版发行 中国人民大学出版社

社 址 北京中关村大街 31 号

邮政编码 100080

电 话 010-62511242 (总编室)

010-62511398 (质管部)

010-82501766 (邮购部)

010-62514148 (门市部)

010-62515195 (发行公司)

010-62515275 (盗版举报)

网 址 <http://www.crup.com.cn>

<http://www.ttrnet.com> (人大教研网)

经 销 新华书店

印 刷 中煤涿州制图印刷厂

规 格 185 mm×260 mm 16 开本

版 次 2011 年 12 月第 1 版

印 张 20.25

印 次 2011 年 12 月第 1 次印刷

字 数 480 000

定 价 35.00 元

版权所有 侵权必究

印装差错 负责调换

内容简介

本教材是根据教育部计算机基础课程教学指导分委员会提出的《关于进一步加强高校计算机基础教学的几点意见》及原人事部、原信息产业部《信息处理技术员考试大纲》的考试要求并结合信息技术教育的现状而编写的。

全书以 Microsoft Office Access 2003 中文版为平台，内容共分 11 章，先后介绍了关系数据库的基本理论与应用技术，主要包括数据库的概念、Access 数据库及数据库对象、数据表的创建和操作、查询、SQL 结构化查询语言、窗体、报表、数据访问页、宏、模块与 VBA 程序程序、学生管理系统实例。全书以理论联系实际的方式讲解知识、介绍 Access 数据库的应用与操作技能，叙述详尽，概念清晰。读者可以通过一边学习、一边实践的方式，达到掌握 Access 数据库应用的目的。

本书结构严谨、可操作性和实用性强，既可以作为高等院校非计算机专业的教材，同时也可以作为全国计算机等级考试考生的培训辅导参考书。

前 言

Access 是由微软公司开发的 Office 办公软件系统中的一个重要组件，是一个功能强大且易于实现和使用的关系型数据库管理系统，拥有良好的图形用户界面和丰富的应用向导。Access 可以用来直接开发一个小型的数据库应用系统，也可以作为网站的后台数据库，具有很好的应用前景。所以，掌握应用数据库技术进行信息处理，已成为一个新时代大学生所必备的基本文化素质之一。

在组织、编写过程中，除考虑讲授 Access 的基本概念和应用之外，还充分考虑了《全国计算机等级考试（NCRE）二级 Access 考试大纲》和《信息处理技术员考试大纲》考试科目的内容。书中采用以案例驱动方式讲解 Access 知识，使学生能从应用角度出发，带着问题学，学以致用，达到提高数据库技术应用能力的目的。

本书在内容的选取上既考虑到大学新生计算机知识的起点明显提高这一现状，又兼顾了学生生源多样性及地区教育的不平衡性所引起的计算机基础知识与操作技能上的差异。

本书共分 11 章，第 1 章介绍了关系数据库的基本概念、数据模型、数据库设计概念等基础知识。第 2 章介绍了 Access 数据库系统功能、特点及应用环境等基本概念。第 3 章介绍了 Access 数据库表的创建、操作等基本操作方法。第 4 章介绍了 Access 中查询的概念、查询的创建及应用的基本方法。第 5 章介绍了 SQL 语言的语法及应用方法。第 6 章介绍了窗体的概念、作用、不同类型的窗体创建方法等。第 7 章介绍了报表概念与用途，报表的创建，报表数据的排序与分组方法。第 8 章介绍了数据访问页的概念及创建方法。第 9 章介绍了宏的概念、设计方法及应用。第 10 章介绍了 VBA 的概念、编程过程及应用方法等。第 11 章以设计“学生管理系统”为例，介绍了应用 Access 数据库系统解决实际应用的方法和步骤。大部分章节后附有上机实验内容，在附录部分附有各章的部分答案。此外，还为教师提供电子课件，老师可登录（<http://jpkc.wntc.edu.cn/ec/C156/Course/Index.htm>）免费下载。

本书由高升宇主编并统稿，索红军、奚建荣、习贵民、刘龙飞任副主编，由张郭军主审并定稿。第 1、2、8 章由索红军编写，第 3、4、5 章由习贵民编写，第 6、7 章由刘龙

飞编写,第9章由高升宇编写,第10、11章以及附录由奚建荣编写。

为便于以后教材的修订,恳请读者、专家及教师多提宝贵意见,以帮助我们不断改进和完善。

在本书的编写过程中,始终得到中国人民大学出版社的关心、支持与帮助,并在本书编写和审阅过程中,提出了许多宝贵意见,且给予了具体指导,在此致以诚挚的谢意。

编者

2011年10月

目 录

第 1 章 数据库的概念	1
1.1 数据库基础知识	1
1.2 数据模型	3
1.3 数据库设计基础	10
本章小结	16
习题	16
第 2 章 Access 数据库及数据库对象	18
2.1 Access 数据库简介	18
2.2 Access 2003 数据库的启动与退出	20
2.3 Access 2003 数据库的创建及其对象	20
本章小结	27
习题	27
第 3 章 数据表的创建和操作	28
3.1 数据表的建立	28
3.2 表结构编辑与深化设计	46
3.3 表数据的编辑、排序与筛选	56
3.4 数据表操作	61
本章小结	65
习题	66
第 4 章 查询	70
4.1 查询概述	70
4.2 选择查询	73
4.3 交叉表查询	86

4.4 操作查询	89
本章小结	96
习题	96
第5章 SQL 结构化查询语言	99
5.1 SQL 语言简介	99
5.2 SQL 数据查询语句	100
5.3 SQL 数据定义	110
5.4 SQL 数据操纵	113
本章小结	114
习题	114
第6章 窗体	116
6.1 窗体概述	116
6.2 快速创建窗体	121
6.3 使用窗体设计视图设计窗体	136
6.4 窗体的美化	155
本章小结	159
习题	160
第7章 报表	165
7.1 报表概述	165
7.2 快速创建报表	169
7.3 使用报表设计器制作报表	178
7.4 编辑报表	183
7.5 创建子报表	192
7.6 报表的预览与打印	195
本章小结	196
习题	196
第8章 数据访问页	201
8.1 数据访问页概述	201
8.2 快速创建数据访问页	202
8.3 用数据访问页设计器创建数据访问页	208
8.4 数据访问页的使用	214
本章小结	215
习题	215

第 9 章 宏	217
9.1 宏的概念	217
9.2 宏的创建	219
9.3 运行宏	228
9.4 宏的调试	229
本章小结	230
习题	230
第 10 章 模块和 VBA 程序设计	233
10.1 模块概念	233
10.2 模块分类	234
10.3 创建模块	237
10.4 VBA 程序设计基础	239
10.5 VBA 流程控制语句	254
10.6 过程调用和参数传递	262
10.7 VBA 程序的调试	266
10.8 VBA 程序运行错误处理	269
10.9 VBA 应用编程实例	272
本章小结	274
习题	274
第 11 章 学生管理系统实例	277
11.1 数据库应用系统的开发过程	277
11.2 系统需求分析	279
11.3 系统数据库设计	279
11.4 系统功能设计	283
习题	289
附录 I 全国计算机等级考试二级 Access 考试大纲	290
附录 II 全国计算机等级考试二级 Access 笔试模拟题	294
附录 III 全国计算机等级考试二级 Access 上机模拟题	300
附录 IV 全国计算机等级考试二级 Access 真题 (2011 年 3 月)	302
附录 V 各章习题参考答案	312

第 1 章

数据库的概念

数据库是一个关于特定主题或用途的、具有一定结构的数据的集合，它与一般的数据文件（其中的数据是无结构的，是一串文字或数字流）不同。数据库中的数据可以是文字、图像、声音等。从最初的数据库发展到今天的大型数据库管理系统，数据库已经成为我们日常生活中重要的组成部分。假如不借助数据库的帮助，许多简单的工作将会变得冗长乏味，甚至会难以实现。

本章作为本书的开篇，将向读者介绍有关数据库的一些预备知识，主要包括数据库基础知识、基本概念、数据模型、数据库设计基础等有关方面的内容。

1.1 数据库基础知识

数据库是在 20 世纪 60 年代兴起的一种数据管理技术。所谓数据库，顾名思义就是数据的仓库，但它与现实生活中存放物品的仓库不完全相同。数据库中的数据必须有一定的联系，具体地讲，数据库是具有统一的结构形式并存放于统一的存储介质内，并可被各个应用程序共享的数据的集合，是多种应用数据的集成。

1.1.1 信息处理的发展

数据库中的数据有一定的结构形式，而且一般都有确定的意义，也就是包含一定的信息。数据实际上就是描述事物的符号记录，数据经过加工处理之后就成为信息。

信息和数据既有联系又有区别。数据是描述客观事实和概念的一组文字、数字或符号等，它是信息的素材，是信息的载体和表达形式。信息是加工后的数据，是从数据中加工、提炼出来的关于客观事物的知识，能够减少不确定性，对使用者的决策有意义。数据只有经过处理和解释并赋予一定的意义后才成为信息。信息既是客观事物的特征、运动变化的反映，又是事物之间相互作用、相互联系的反映。

信息处理也称为数据处理,它是指将数据转换成信息以及对信息进行再加工的过程,包括信息的筛选、清理、检索、存储、共享等。随着信息类型的多样化、信息数量的庞大化以及人们对信息要求的精细化等,信息处理变得越来越复杂化和多样化。比如对庞大、多样信息的存储与检索必然涉及信息的结构,必然要研究信息的组织形式等。而数据库正是有组织地存储大量信息的仓库,应用数据库可以方便地对信息进行检索、共享与分析等,为相关的处理、决策提供相应的支持。另外,各个信息之间必然也存在一定的联系,如何正确地表达信息之间的联系,处理信息之间的相互关系,也需要对信息进行有组织的管理、分析等,这同样要用到数据库。信息处理最终将与数据库技术结合进行研究与分析。

数据处理是计算机应用的主流,特别是关系数据库管理系统 DBaseⅢ 推出之后,数据库技术就成为计算机应用者必须掌握的技能。数据和信息是数据处理中两个最基本的概念。

数据处理是从某些已知的数据出发,推导加工出一些新的数据,这些新的数据又表示了新的信息。数据处理包括对数据进行收集、存储、传送、整理、检索、计算、输出等各种加工、管理。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,这些操作是数据处理业务的基本环节,而且是任何数据处理业务中必不可少的共有部分。数据处理能力与数据管理方式有着密切的关系,随着数据量的剧增以及计算机技术的发展,数据管理技术的发展大致经历了人工管理、文件管理、数据库管理及分布式数据库管理等 4 个阶段。

1.1.2 数据库系统

通俗地讲,数据库就是数据的集合。例如,每个人都有很多亲戚和朋友,为了保持与他们的联系,我们常常用一个笔记本将他们的姓名、地址、电话等信息都记录下来,这样要查谁的电话或地址就很方便了。这个“通讯录”就是一个最简单的“数据库”,每个人的姓名、地址、电话等信息就是这个数据库中的“数据”。我们可以在笔记本这个“数据库”中添加新朋友的个人信息,也可以由于某个朋友的电话变动而修改他的电话号码这个“数据”。不过说到底,我们使用笔记本这个“数据库”最主要的目的还是为了能随时查到某位亲戚或朋友的地址、邮编或电话号码这些“数据”。如果我们将笔记本中的这些数据信息转移存储到计算机中,并按一定的结构组织起来,就形成了我们现在所说的数据库。

另外借助数据库的帮助,可以将复杂甚至难以实现的工作变得简单、容易。例如,不应用数据库,假设供应商电话号码存储在以下不同位置:在包含供应商电话号码的卡文件中、在文件柜内的产品信息文件中、在包含订单信息的电子表格中。如果供应商的电话号码发生了变化,则必须在所有这三个地方更新该信息。但是,应用数据库后,在精心设计的数据库中,只存储一次电话号码,就可在多处使用,所以只需在一个地方更新该信息,便可实现在数据库中任何使用该电话号码的地方自动更新此电话号码。

数据库(DataBase, DB)就是为了实现一定的目的而按某种规则组织起来的“数据”的“集合”。与文件系统比较,数据库系统有以下特点:

(1) 数据的结构化。在文件系统中,各个文件不存在相互联系,因此从单个文件来看,数据是有结构的,但从整个系统来看,数据又是没有结构的。而数据库中的数据存储

是按同一结构进行的。

(2) 数据共享。在文件系统中,数据一般是由特定的用户专用的。数据库系统中,数据共享是它的主要目的,数据库系统提供一套有效的管理手段,保持数据的完整性、一致性和安全性,使数据具有充分的共享性。

(3) 数据独立性。在文件系统中,数据结构和应用程序相互依赖,一方的改变总会影响另一方的改变。数据库系统则力求减少相互依赖,实现数据的独立性。

(4) 可控冗余度。数据专用时,每个用户拥有并使用自己的数据,难免有许多数据相互重复,即冗余。数据实现共享后,不必要的重复将全部消除。当然在某些情况下为了提高查询效率,也可保留少量冗余,其冗余度由设计人员控制。

在信息社会里,数据库的应用非常广泛,如银行业、通信行业用数据库存储客户信息;企业用数据库管理原料、生产、产品等信息;经销行业用数据库存储生产、库存、销售信息;学校用数据库管理学生的个人信息、课程成绩等。

数据库系统(DataBase System, DBS)是一个计算机应用系统,它由计算机硬件、数据库管理系统、数据库、应用程序和用户等部分组成。数据库管理系统(DBMS)是处于用户和数据库之间的一种系统软件。DBMS提供对数据库中数据资源进行统一管理和控制的功能,它是数据库系统的核心,其功能的强弱是衡量数据库系统性能优劣的主要指标。

1.2 数据模型

数据模型是指数据库中数据与数据之间的关系,它是数据库系统中的一个关键概念。数据模型不同,相应的数据库系统就完全不同,任何一个数据库系统都是基于某种数据模型的。不同的数据模型提供了模型化数据和信息的不同工具,根据模型应用的不同目的,可以将模型分为两类或两个层次:一是概念模型,二是数据模型。前者是按用户的观点来对数据和信息建模,后者是按计算机系统的观点对数据和信息建模。

美国国家标准协会(American National Standard Institute, ANSI)的数据库管理系统研究小组于1978年提出了标准化的建议,将数据库结构分为3级:面向用户或应用程序员的用户级、面向建立和维护数据库人员的概念级、面向系统程序员的物理级。用户级对应外模式,概念级对应模式,物理级对应内模式,使不同级别的用户对数据库形成不同的视图。所谓视图,就是指观察、认识和理解数据的范围、角度和方法,是数据库在用户“眼中”的反映。很显然,不同层次(级别)用户所“看到”的数据库是不相同的。

模式又称概念模式或逻辑模式,对应于概念级。它是由数据库设计者综合所有用户的数据,按照统一的观点构造的全局逻辑结构,是对数据库中全部数据的逻辑结构和特征的总体描述,是所有用户的公共数据视图(全局视图)。它是由数据库管理系统提供的数据模式描述语言(Data Description Language, DDL)来描述、定义的,体现、反映了数据库系统的整体观。

外模式又称子模式,对应于用户级。它是某个或某几个用户所看到的数据库的数据视图,是与某一应用有关的数据的逻辑表示。外模式是从模式导出的一个子集,包含模式中允许特定用户使用的那部分数据。用户可以通过外模式描述语言来描述、定义对应于用户

的数据记录（外模式），也可以利用数据操纵语言（Data Manipulation Language, DML）对这些数据记录进行处理。外模式反映了数据库的用户观。

内模式又称存储模式，对应于物理级，它是数据库中全体数据的内部表示或底层描述，它描述了数据在存储介质上的存储方式及物理结构，对应着实际存储在外存储介质上的数据库。内模式由内模式描述语言来描述、定义，它是数据库的存储观。

在一个数据库系统中，只有唯一的数据库，因而作为定义、描述数据库存储结构的内模式和定义、描述数据库逻辑结构的模式，也是唯一的，但建立在数据库系统之上的应用则是非常广泛、多样的，所以对应的外模式不是唯一的，也不可能是唯一的。

1. 概念模型

概念模型是对客观事物及其联系的抽象，用于信息世界的建模，它强调其语义表达能力，以及能够较方便、直接地表达应用中各种语义知识。这类模型概念简单、清晰、易于被用户理解，是用户和数据库设计人员之间进行交流的语言。概念模型的表示方法很多，其中最著名的是 E-R 表示方法（实体—联系方法），它用 E-R 图来描述现实世界的概念模型。用 E-R 图表示概念模型是 1976 年由 Peter Chen 首先提出的，该模型将现实世界的要求转换成实体、联系、属性等几个概念，以及它们间的两种基本连接关系可以用一种图非常直观地表示出来。E-R 图的主要成分是实体、联系和属性。在概念模型中主要有如下一些概念：

实体：实体是现实世界中可区别于其他对象的“事件”或物体。如学生是一个实体。

实体集：实体集是具有相同类型及共享相同性质（属性）的实体集合。如全班学生就是一个实体集。

属性：实体通过一组属性来表示，属性是实体集中每个成员具有的描述性性质。将一个属性赋予实体集，表明数据库为实体集中每个实体存储相似信息，但每个实体在自己的每个属性上都有各自的值。如学生实体有学号、姓名、年龄、性别等属性。

关键字和域：实体的某一属性或属性组合，其值能唯一标识出某一实体，称为关键字，如学号是学生实体集的关键字，学号可以区分每一个学生。每个属性都有一个可取值的集合，称为该属性的域，如性别的域是“男”、“女”。

联系：客观事物之间的关系即信息世界中实体之间的联系。常见的实体联系有 3 种：一对一联系（1:1），如学校与校长间的联系，一个学校与一个校长间相互一一对应；一对多联系（1:n），如公司与员工间的联系，一个公司可以有多个员工，一个员工只能属于一个公司；多对多联系（n:m），如教师与学生的联系，一个教师可以为多个学生上课，而一个学生也可以受教于多个教师。

2. 数据模型

数据模型是人们运用数学方法描述数据库技术所研究的对象，即客观事物（如人、物、工作、效果、概念等）以及反映这些客观事物之间相互联系的数据。

数据库中的数据是结构化的，是按某种数据模型来组织的。当前常用的数据模型有 3 类：层次模型、网状模型和关系模型。它们之间的根本区别在于数据之间联系的表示方式不同：层次模型用树型结构来表示数据之间的联系；网状模型用图形结构来表示数据之间的联系；关系模型用二维表来表示数据之间的联系。

(1) 层次模型。

层次模型是将数据元素分为若干层，最高层只有一个元素，称为根。上一层与下一层发生关系，下一层只与再下一层发生关系，它是一个定向的有序树，表示了一对多的联系，如图 1—1 所示。例如家谱、行政隶属等各数据元素之间的关系。

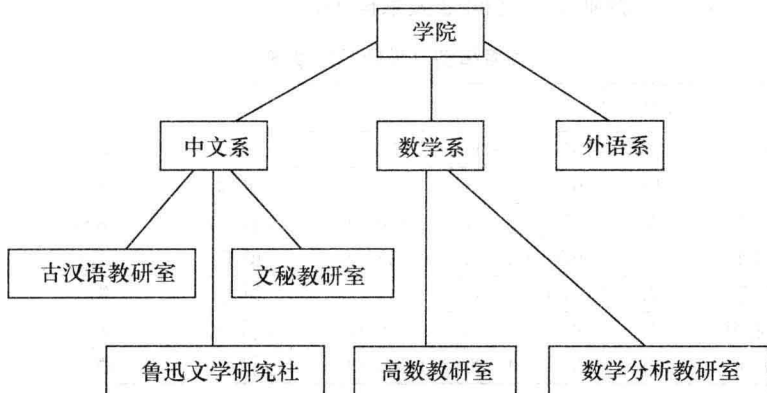


图 1—1 层次模型

层次模型的基本结构是树形结构，具有以下特点：

- ① 每棵树有且仅有一个无双亲结点，称为根；
- ② 树中除根外所有结点有且仅有一个双亲。

(2) 网状模型。

层次数据模型中的任意一个基本层次互相连通的集合，就是一个网状数据模型。它能表示多对多的联系，数据之间不分层次，每个数据元素都和任意一个或多个其他数据元素相连接，形成网络，如图 1—2 所示。例如售票系统、城市交通等。从图论上看，网状模型是一个不加任何条件限制的无向图。

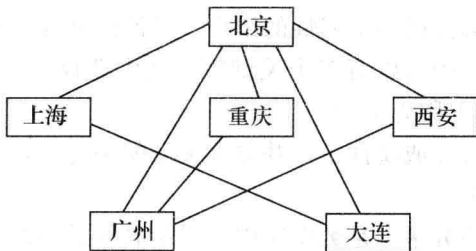


图 1—2 网状模型

(3) 关系模型。

关系模型是由若干行、若干列构成的二维表（表格）的结构。其中，每一列为一个数据项，是数据库文件中的最基本单位，表示了实体的一个属性，叫做字段。每一列给定一个名称，叫做字段名。每一行通过各个属性表示了一个完整的实体，叫做元组或记录，如表 1—1 所示。在二维表格中，第一行标明了各字段的名称，表明该关系中实体所具有的属性，体现了二维表格的结构。除了第一行外的其他任一行的数据就是一个元组（表 1—1 中第二行开始）。关系模型是使用最广泛的数据模型，目前大多数数据库系统都是关系模

型的。

1.2.1 关系数据模型

关系模型是用二维表格结构来表示实体与实体之间联系的数据模型,表 1—1 就是一张教师基本情况表,每个教师实体之间是关系模型的。

表 1—1 关系模型 (教师基本情况)

编号	姓名	性别	出生年月	工资	职称	学历	所属院系
T001	蒋孔	男	1950-07-05	2 350.00	教授	硕士	中文系
T002	李华	男	1968-04-08	1 568.00	讲师	博士	经济学院
T003	孔艳	女	1955-05-04	2 100.00	副教授	本科	中文系
T004	戴红利	女	1970-05-03	1 368.00	助教	硕士	数学系
T005	杜凯	男	1970-04-09	1 759.00	讲师	本科	中文系
T006	王珊珊	女	1975-05-06	2 300.00	副教授	博士	计算机系
T007	东风	男	1965-04-25	1 566.00	副教授	硕士	计算机系

关系模型是建立在关系代数基础上的,因而具有坚实的理论基础。与层次模型和网状模型相比,具有数据结构单一、理论严密、使用方便、易学易用的特点,因此,目前大多数数据库管理系统的数据模型都是采用关系数据模型,成为数据库应用的主流,如 Access 就是一种关系型的数据库管理系统。

1. 关系数据模型的基本概念

关系:一个关系就是一张二维表,每个关系有一个关系名,即 Access 中的数据表。

关系模式:对关系的描述称为关系模式,例如表 1—1 所示的关系被描述为:教师基本情况(编号,姓名,性别,出生年月,工资,职称,学历,所属院系)。

属性:表中的列称为属性,每一列有一个属性名,且属性名唯一,对应 Access 中的字段。

关键字:关系中一个属性或多个属性的组合,其值能够唯一地标识一个元组。

主关键字:在一个关系中可以有多个关键字,从中选择一个来与其他关系建立联系,称为主关键字,在 Access 中称为主键。

外关键字:关系中的属性或属性组,并非该关系的关键字,但它们是另一个关系的关键字,称其为该关系的外关键字。

数据项:数据项也称为分量,是数据库中可以命名的最小逻辑数据单位,指某个元组对应的属性值,用来描述属性的数据。

元组:二维表中的一行,对应 Access 中的记录,指的是关系中的一行数据,用它描述实体。它是数据项的有序集,即一个记录是由若干个数据项组成。

索引:为了加快数据库的访问速度,所建立的一个独立的文件或表格。

2. 关系规范化

不是所有的二维表都能称为关系。一个二维表要称为关系或合理的关系,还应满足一定的限制,即关系要规范化。关系规范化是指关系模型中的每一个关系模式都必须满足一定的要求,这些要求可分为最基本要求和高级要求两大类。满足最基本要求的二维表才能称之为关系,最基本的要求有三个:

(1) 属性不可再分或多值。

最基本的要求是,关系中的每个属性都必须是不可再分的数据单元且属性不得多值,即通常人们讲的表中不能再含表,属性值仅一个。这称为关系的一级范式:1NF (first normal form)。通常表示为: $R \in 1NF$ 。例如表 1—2 所示的表,由于在成绩数据项中,又包含了四个“子数据项”,因此就不是一个关系。

表 1—2 具有组合数据项的非规范化表

学号	姓名	成绩			
		语文	数学	物理	化学
001	蒋孔	78	95	89	90
002	李华	89	90	67	78
003	孔艳	93	89	91	62
...	...	...	...	...	...

又例如表 1—3 所示的表,由于在学历数据项中,孙丹的学历及毕业年份中包含了两栏数据,属于属性多值,因此也不是一个关系。

表 1—3 具有多值数据项的非规范化表

编号	姓名	性别	职称	学历	毕业年份
T007	东风	男	副教授	硕士	1990
T008	葛优	男	讲师	本科	1992
T009	孙丹	女	副教授	大学 研究生	1990 1996
...	...	...	...	...	...

(2) 属性不得同名。

同一关系中不能有相同的属性名出现,但属性的左右位置可以任意。

(3) 元组不得完全相同。

同一关系中不允许有完全相同的两个元组,但元组的先后次序可以任意。

需要说明的是,符合最基本要求的关系并不是好关系,它存在着冗余大、插入异常、删除异常、修改异常等危险。

关系数据库的规范化理论认为:关系数据库中的每个关系都要满足一定的规范。根据满足规范的条件不同,可以将规范分为 5 个等级:第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、第四范式(4NF)、第五范式(5NF)。一般情况下,只要把数据规范到第三范式标准就可以满足需要了。

第一范式:在一个关系中消除重复字段,且各字段都是最小的逻辑存储单位。

第二范式:关系模型属于第一范式,关系中每一个非主关键字段都完全依赖于主关键字段,不能只部分依赖于主关键字段的一部分。

第三范式:关系模型属于第二范式,任何字段不能由其他字段派生出来,它要求字段没有冗余,即要求去除传递依赖。

并不是规范越高越好,例如,满足第三范式的关系数据中没有冗余。但是,没有冗余的数据库未必是最好的数据库,有时为了提高运行效率,就必须降低范式标准,适当保留冗余数据。

1.2.2 关系运算

关系数据库系统的特点之一是它建立在数据理论的基础之上。在对数据库进行查询时,人们总是希望尽快找到所需要的数据,这就需要对关系进行一定的运算。有很多数据理论可以表示关系模型的数据操作,其中最为著名的是关系代数与关系演算。已经证明两者在功能上是等价的。关系的基本运算分为两类:一类是基于传统的集合运算的关系运算,另一类是专门的关系运算。

1. 关系模型的基本操作

关系由若干个不同的元组所组成, n 元关系是一个 n 元有序组的集合。设有一个 n 元关系 R ,它有 n 个域,分别是 $D_1, D_2, \dots, D_n$,此时它们的笛卡儿积是:

$$D_1 \times D_2 \times \dots \times D_n$$

该集合的每个元素都具有如下形式的 n 元有序组:

$$(d_1, d_2, \dots, d_n), d_i \in D_i (i=1, 2, \dots, n)$$

该集合与 n 元关系 R 有如下联系:

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

即 n 元关系 R 是 n 元有序组的集合,是它的域的笛卡儿积的子集。

关系模型有插入、删除、修改和查询四种操作,它们又可以进一步分解成6种基本操作:

① 关系的属性指定。指定一个关系内的基本属性,用它确定关系这个二维表中的列,主要用于检索和定位。

② 关系的元组选择。用一个逻辑表达式给出关系中满足此表达式的元组,用它确定关系这个二维表的行,主要用于检索和定位。

③ 两个关系的合并。将两个关系合并成一个关系。用此操作可以不断合并从而可以将若干个关系合并成一个关系,以建立多个关系间的检索与定位。

用上述三个操作可以进行多个关系的定位。

④ 关系的查询。在一个关系或多个关系间查询,查询的结果也为关系。

⑤ 关系元组的插入。在关系中添加一些元组,用它完成插入与修改。

⑥ 关系元组的删除。在关系中删除一些元组,用它完成删除与修改。

2. 关系模型的基本运算

由于操作是对关系的运算,而关系是有序组的集合,因此,可以将操作看成是集合的运算。

(1) 插入。

设关系 R 需要插入若干元组,若要插入的元组组成的关系为 R' ,则插入可用集合并运算表示为:

$$R \cup R'$$

(2) 删除。

设关系 R 需要删除一些元组,若要删除的元组组成的关系为 R' ,则删除可用集合差