

Access 数据库应用

Access Shujuku Yingyong

许国柱 编著



华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

014009945

TP311.138AC
200

Access

数据库应用

Access Shujuku Yingyong

许国柱 编著



北航

C1696501

TP311.138AC

200



华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

·广州·

图书在版编目(CIP)数据

Access 数据库应用/许国柱编著. —广州: 华南理工大学出版社, 2013. 8
ISBN 978-7-5623-4001-0

I. ①A… II. ①许… III. ①关系数据库系统 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2013)第 177699 号

Access 数据库应用

许国柱 编著

出 版 人: 韩中伟

出版发行: 华南理工大学出版社

(广州五山华南理工大学 17 号楼, 邮编 510640)

http://www.scutpress.com.cn E-mail: scutc13@scut.edu.cn

营销部电话: 020-87113487 87111048 (传真)

责任编辑: 兰新文

印 刷 者: 广东省农垦总局印刷厂

开 本: 787mm × 1092mm 1/16 印张: 18.5 字数: 450 千

版 次: 2013 年 8 月第 1 版 2013 年 8 月第 1 次印刷

印 数: 1~3000 册

定 价: 37.00 元

版权所有 盗版必究 印装差错 负责调换

前 言

随着信息技术的不断发展,数据库系统已成为现实生活中不可或缺的一部分,数据库技术的应用遍及各行各业、各种各样的系统中,可以说无处不在,数据库技术是使事务或信息数据化、电算化、网络化、自动化的基础。无论是一个小企业的 OA 系统,还是大到在互联网或专用网支持下的全球或全国系统,都离不开数据库的应用,因此,数据库技术的基本知识和基本技能已成为各类专业人员必须学习和掌握的技术。

本书以应用为目的,以案例为引导,详细介绍 Access 的主要功能和使用方法。主要内容包括数据库基本概念、关系运算、Access 基本操作、结构化查询语言、规范化设计、实体联系模型、数据库系统设计等基本知识;表、查询、窗体、报表、数据访问页、宏与模块的创建与使用。本书选取典型的项目为载体,采用项目驱动的方式,带着任务和问题学知识、练技能,具有情景真实性、过程可操作性、结果可检验性的特点,主要培养目标是通过课程的学习,可以使学生尽快掌握 Access 的基本功能和操作,掌握 Access 的编程功能和技巧,具备应用小型数据库管理应用系统的能力。

本书主要针对非计算机专业的数据库原理与应用课程,重点突出基础性、先进性、实用性和可操作性,注重对学生创新能力、自学能力和动手能力的培养。本书适合作为普通高等学校非计算机专业本、专科学生的计算机课程教材,也可作为相关培训班的培训教材,还可作为读者自学提高的参考书。

本书配有网络课程,网址是:<http://jpkc.gdsdxy.cn/2010/sjkyljyy/>,网络课程的内容(包括课程设计、教学方案)与教材完全对应,网络资源丰富,实训、实操、习题库、试题库、动画教学、视频教学、在线测试、认证考试等一应俱全。由于编者水平所限,书中不足之处,敬请读者批评指正。

编 者

2013 年 5 月

目 录

项目 1 认识 Access 数据库	(1)
任务 1.1 掌握数据库基础知识	(1)
1.1.1 数据与信息	(1)
1.1.2 数据库、数据库系统与数据库管理系统	(2)
1.1.3 数据库技术的产生与发展	(3)
1.1.4 数据库技术的特点	(5)
1.1.5 关系模型	(6)
1.1.6 关系数据库	(13)
任务 1.2 认识 Access	(14)
1.2.1 Access 特点	(14)
1.2.2 Access 系统的启动与退出	(15)
1.2.3 Access 系统界面	(16)
1.2.4 Access 数据库的对象	(20)
任务 1.3 掌握 Access 基本运算	(24)
1.3.1 Access 数据库的运算量	(24)
1.3.2 Access 数据库的常用函数	(25)
1.3.3 Access 数据库的数据运算符与表达式	(30)
思考题	(34)
项目 2 创建数据库和表	(39)
任务 2.1 数据库设计与创建	(39)
2.1.1 数据库设计	(39)
2.1.2 在 Access 系统中创建新数据库	(41)
2.1.3 数据库的基本操作	(50)
任务 2.2 创建表	(53)
2.2.1 Access 数据类型	(53)
2.2.2 在 Access 系统中建立表结构	(54)
2.2.3 设置字段属性	(61)
任务 2.3 建立表间关系	(70)
2.3.1 表间存在的关系	(70)
2.3.2 建立表间的关系	(72)
任务 2.4 维护表	(75)
2.4.1 表结构的维护	(75)

2.4.2	表数据的维护	(76)
2.4.3	调整表格式	(77)
任务 2.5	表记录的编辑、排序与筛选	(81)
2.5.1	查找数据	(81)
2.5.2	替换数据	(83)
2.5.3	排序记录	(85)
2.5.4	筛选记录	(86)
思考题		(90)
项目 3 创建与使用查询		
任务 3.1	认识查询	(95)
3.1.1	查询的概念	(95)
3.1.2	查询的功能	(95)
3.1.3	查询的类型	(96)
任务 3.2	创建选择查询	(97)
3.2.1	使用向导创建选择查询	(97)
3.2.2	使用“设计”视图创建选择查询	(103)
任务 3.3	创建交叉查询	(121)
3.3.1	认识交叉表查询	(121)
3.3.2	使用向导创建交叉表查询	(121)
3.3.3	使用“设计”视图创建交叉表查询	(128)
任务 3.4	创建参数查询	(132)
3.4.1	认识参数查询	(132)
3.4.2	创建参数查询	(134)
任务 3.5	创建操作查询	(138)
3.5.1	创建生成表查询	(138)
3.5.2	创建更新查询	(141)
3.5.3	创建追加查询	(142)
3.5.4	创建删除查询	(143)
任务 3.6	用 SQL 语言创建查询	(144)
3.6.1	SQL 语言	(144)
3.6.2	用 SQL 语言创建查询	(145)
思考题		(147)
项目 4 创建与使用窗体		
任务 4.1	认识窗体	(153)
4.1.1	窗体的功能	(153)
4.1.2	窗体的组成	(153)
4.1.3	窗体的视图	(154)

4.1.4 窗体的类型	(155)
任务 4.2 使用向导创建窗体	(157)
4.2.1 使用“自动创建窗体”创建窗体	(157)
4.2.2 使用“窗体向导”创建窗体	(159)
4.2.3 使用“窗体向导”创建主/子式窗体	(162)
4.2.4 使用“自动窗体”创建数据表/图式窗体	(166)
4.2.5 使用“设计视图”创建窗体	(171)
任务 4.3 向窗体添加常用控件	(175)
4.3.1 窗体中的控件	(175)
4.3.2 标签控件	(177)
4.3.3 文本框控件	(179)
4.3.4 组合框和列表框控件	(182)
4.3.5 命令按钮控件	(184)
4.3.6 选项组控件	(186)
4.3.7 图像、未绑定对象框和绑定对象框控件	(188)
4.3.8 直线、矩形控件	(188)
4.3.9 窗体查询综合应用	(189)
任务 4.4 利用窗体操作数据	(190)
4.4.1 浏览记录	(190)
4.4.2 编辑记录	(190)
4.4.3 查找和替换数据	(191)
4.4.4 排序记录	(192)
思考题	(192)
项目 5 创建与使用报表	(197)
任务 5.1 认识报表	(197)
5.1.1 报表的功能	(197)
5.1.2 报表的结构	(198)
5.1.3 报表的视图	(198)
5.1.4 报表的类型	(199)
任务 5.2 创建报表	(199)
5.2.1 使用“自动创建报表”创建报表	(200)
5.2.2 使用“报表向导”创建报表	(200)
5.2.3 使用“图表向导”创建图表式报表	(209)
5.2.4 使用“标签向导”创建标签式报表	(211)
5.2.5 使用“设计视图”创建报表	(214)
5.2.6 在报表中添加计算控件	(215)
5.2.7 报表排序和分组	(217)
5.2.8 创建子报表	(220)

任务 5.3 报表的打印	(223)
5.3.1 报表的美化	(224)
5.3.2 报表的预览和打印	(226)
思考题	(227)
项目 6 创建与使用数据访问页	(232)
任务 6.1 认识数据访问页	(232)
6.1.1 数据访问页概述	(232)
6.1.2 数据访问页的视图	(232)
任务 6.2 创建数据访问页	(233)
6.2.1 创建数据访问页	(233)
6.2.2 编辑数据访问页	(238)
思考题	(240)
项目 7 创建与使用宏和模块	(243)
任务 7.1 认识宏	(243)
7.1.1 宏的概念	(243)
7.1.2 宏的功能	(244)
7.1.3 宏的设计窗口	(244)
7.1.4 常用的宏操作命令	(245)
任务 7.2 宏的创建与使用	(246)
7.2.1 宏设计工具栏	(246)
7.2.2 创建宏	(247)
7.2.3 调试宏	(249)
7.2.4 编辑宏	(250)
7.2.5 运行宏	(251)
任务 7.3 创建模块	(253)
7.3.1 模块的基本概念	(253)
7.3.2 创建模块	(254)
7.3.3 程序控制语句	(256)
7.3.4 宏与模块之间的转换	(262)
思考题	(263)
项目 8 设计与实现学生管理系统	(267)
任务 8.1 学生管理系统设计	(267)
8.1.1 数据库系统设计流程	(267)
8.1.2 学生管理数据库系统需求分析及模块	(270)
任务 8.2 学生管理系统界面设计	(271)
8.2.1 学生管理数据库“数据维护”窗体	(271)

8.2.2	学生管理数据库“数据浏览”窗体	(272)
8.2.3	学生管理数据库“数据查询”窗体	(272)
任务 8.3	学生管理系统报表设计	(273)
8.3.1	学生管理数据库单表报表	(273)
8.3.2	学生管理数据库多表报表	(273)
8.3.3	学生管理数据库统计汇总报表	(273)
任务 8.4	学生管理系统主控面板设计	(275)
8.4.1	创建学生管理数据库“系统登录”窗体	(275)
8.4.2	创建学生管理数据库切换面板窗体	(275)
任务 8.5	维护学生管理系统	(279)
8.5.1	压缩和修复数据库	(279)
8.5.2	设置数据库的密码	(281)
8.5.3	将“学生管理”数据库系统生成为 MDE 文件	(282)
思考题		(283)
参考文献		(285)

项目 1 认识 Access 数据库

【教学目标】

- (1) 掌握数据库的基本概念；
- (2) 了解关系模型与关系数据库的基本知识；
- (3) 掌握 Access 系统的安装和启动方法；
- (4) 掌握 Access 系统界面各组成部分的含义；
- (5) 掌握 Access 基本运算方法。

任务 1.1 掌握数据库基础知识

1.1.1 数据与信息

1.1.1.1 数据

数据 (data) 是关于自然、社会现象和科学试验的定量或定性的记录, 是对客观世界所存在的事物的一种表征。

数据的概念在数据处理领域中已大大地拓宽了, 不仅仅是指传统意义的由 0~9 组成的数字, 而是所有可以输入到计算机中并能被计算机处理的符号的总称。

在计算机中可表示数据的种类很多, 除了数字以外, 文字、图形、图像、声音都可以通过扫描仪、数码摄像机、数字化仪等具有模/数转换功能的设备进行数字化, 所以这些都是数据。如: 超市商品的价格、学生的基本情况、员工的照片、罪犯的指纹、播音员朗诵的佳作、气象卫星云图……都可以是数据。

数据是数据库中存储的基本对象, 也是数据库用户操作的对象。数据应按照需求进行采集, 并有结构地存入数据库中。

1.1.1.2 信息

所谓信息, 是以数据为载体的对客观世界实际存在的事物、事件和概念的抽象反映。具体说, 信息是一种被加工为特定形式的数据, 是通过人的感官 (眼、耳、鼻、舌、身) 或各种仪器仪表和传感器等感知出来并经过加工而形成的反映现实世界中事物的数据。

例如: 气象部门通过“今年 11 月份武汉的日平均气温为摄氏 20 度”的数据, 分析得出“今年是个暖冬”的信息。

数据和信息是两个互相联系、互相依赖但又互相区别的概念。数据是用来记录信息的可识别的符号, 是信息的具体表现形式。数据是信息的符号表示或载体, 信息则是数据的内涵, 是对数据的语义解释。只有经过提炼和抽象之后, 具有使用价值的数据才能成为信息。

1.1.1.3 数据处理

数据要经过处理才能变为信息。数据处理是将数据转换成信息的过程,是指对信息进行收集、整理、存储、加工及传播等一系列活动的总和。数据处理的目的是从大量的、杂乱无章的甚至是难于理解的原始数据中,提炼、抽取人们所需要的有价值、有意义的数据(信息),作为科学决策的依据。

可用下式简单地表示信息、数据与数据处理的关系:

$$\text{信息} = \text{数据} + \text{数据处理}$$

数据是原料,是输入,而信息是产出,是输出结果。数据处理的真正含义应该是为了产生信息而处理数据。数据、数据处理、信息的关系如图 1-1 所示。

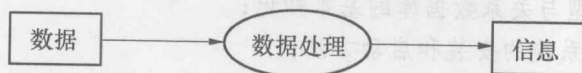


图 1-1 数据、数据处理、信息的关系

数据的组织、存储、检查和维护等工作是数据处理的基本环节,这些工作一般统称为数据管理。

1.1.2 数据库、数据库系统与数据库管理系统

1.1.2.1 数据库

数据库(database, DB)可以直观地理解为存放数据的仓库,只不过这个仓库是建立在计算机的大容量存储器上(如硬盘)。数据不仅需要合理地存放,还要便于经常查找,因此相关的数据及数据之间的联系必须按一定的格式有组织地存储;数据库不仅仅是创建者本人使用,还可以供多个用户从不同的角度共享,即多个不同的用户,为了达到不同的应用目的,使用多种不同的语言,同时存取数据库,甚至同时存取同一块数据。

可以认为:数据库是长期存储在计算机内的、有结构的、大量的、可共享的数据集合。

如教务处学籍管理数据库中有组织地存放了学生的基本情况、课程情况、学生选课情况、开课情况、教师情况等内容,可供教务处、各系教学办、班主任、任课教师、学生等共同使用。

数据库技术使数据能按一定格式组织、描述和存储,并且具有较小的冗余度、较高的数据独立性和易扩展性,并可为多个用户所共享。

数据库是一个企业、组织或机构中各种应用所需要保存和处理的数据集合,各部门应根据工作需要建立符合密级要求、门类齐全、内容准确、更新及时的数据库。

1.1.2.2 数据库系统

基于数据库的计算机应用系统称为数据库系统(Database System, DBS),主要包括:

(1) 支持数据库系统的计算机硬件环境;

(2) 以数据为主体的数据库;

(3) 管理数据库的系统软件 DBMS;

(4) 支持数据库系统的操作系统环境;

(5) 数据库系统开发工具;

(6) 开发成功的数据库应用软件；

(7) 管理和使用数据库系统的人。

它们之间的关系如图 1-2 所示。

1.1.2.3 数据库管理系统

为了方便数据库的建立、运用和维护，人们研制了一种数据管理软件——数据库管理系统（Database Management System, DBMS）。

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，在数据库建立、运用和维护时对数据库进行统一控制、统一管理，使用户能方便地定义数据和操纵数据，并能够保证数据的安全性、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。数据库管理系统是整个数据库系统的核心。

数据库管理系统是对数据进行管理的系统软件，用户在数据库系统中做的一切操作，包括数据定义、查询、更新及各种控制，都是通过 DBMS 进行的，常见的 DB2、Oracle、Sybase、Informix、MS SQL Server、MySQL、FoxPro、Access 等软件都属于 DBMS 的范畴。



图 1-2 数据库系统

1.1.3 数据库技术的产生与发展

数据管理技术的发展经历了人工管理、文件系统和数据库系统 3 个阶段。

1.1.3.1 人工管理阶段

20 世纪 50 年代中期以前为人工管理阶段，是计算机数据管理的初级阶段。

这一阶段计算机主要用于科学计算，硬件中的外存只有卡片、纸带、磁带，没有磁盘等直接存取设备；软件只有汇编语言，没有操作系统，更无统一的管理数据的软件；对数据的管理完全在程序中进行，数据处理的方式基本上是批处理。程序员编写应用程序时，要考虑具体的数据物理存储细节，即每个应用程序中都要包括数据的存储结构、存取方法、输入方式、地址分配等，如果数据的类型、格式或输入输出方式等逻辑结构或物理结构发生变化，必须对应用程序做出相应的修改，因此程序员负担很重。另外，数据是面向程序的，一组数据只能对应一个程序，很难实现多个应用程序共享数据资源，因此程序之间有大量的冗余数据。

1.1.3.2 文件系统阶段

20 世纪 50 年代后期至 60 年代中期，随着计算机软硬件的发展，出现了文件系统，负责对数据进行管理。

这一阶段，计算机已大量用于信息管理。硬件有了磁盘、磁鼓等直接存储设备；在软件方面，出现了高级语言和操作系统；操作系统中有了专门管理数据的软件，一般称为文件系统，用户可以把相关数据组织成一个文件存放在计算机中，由文件系统对数据的存取进行管理，处理方式有批处理，也有联机处理。

文件管理数据的特点如下：

(1) 数据可以长期保存。数据以文件形式存储在计算机的直接存储设备中,可长期保存并反复使用。用户可随时对文件进行查询、修改和增删等处理。

(2) 有专门的数据管理软件——文件系统。由专门的软件即文件系统进行统一的数据管理,文件系统把数据组织成内部有结构的记录,程序员只需与文件名打交道,不必知道数据的物理存储,大大减轻了程序员的负担。

(3) 程序与数据间有一定独立性。数据有两种形式,即用户眼里看到的逻辑结构(称为逻辑文件)和实际存储的物理结构(称为物理文件)。

文件系统中已允许逻辑文件和物理文件有所区别并提供两者之间的转换,数据物理存储结构发生变化时,只要改变文件系统的存取方式,不一定影响程序的运行,从而部分实现了逻辑数据和物理数据的相互独立性。

文件系统阶段对数据的管理有了很大的进步,但一些根本性问题仍没有彻底解决,主要问题如下:

(1) 数据冗余度大。数据冗余度指同一数据重复存储时的重复程度。文件系统阶段各数据文件之间没有有机的联系,一个文件基本上对应于一个应用程序,数据不能共享,因此数据冗余度大。

(2) 数据独立性不高。文件系统文件是为某一特定应用服务的,许多情况下不同的应用程序使用的数据和程序相互依赖,系统不易扩充。一旦改变数据的逻辑结构,必须修改相应的应用程序,而应用程序发生变化(改用另一种程序设计语言来编写程序),也需修改数据结构。

(3) 数据一致性差。由于相同数据的重复存储、各自管理,在进行更新操作时,容易造成数据的不一致性。

1.1.3.3 数据库系统阶段

20 世纪 60 年代末发生了对数据库技术起着奠定作用的 3 件大事,标志着数据管理进入新时代——数据库系统阶段。下面是这 3 件大事:

(1) 1968 年美国 IBM 公司推出了世界上第一个基于层次模型的大型商用数据库管理系统 IMS (Information Management System)。

(2) 1969 年美国数据系统语言研究会 (Conference on Data System Language, CODASYL) 下属的数据库任务组 (Database Task Group, DBTG) 提出了基于网状模型的 DBTG 系统。

(3) 1970 年美国 IBM 公司的高级研究员 E. F. Code 发表的“A Relation Model of Data for Large Shared Data Banks”的论文,提出了关系数据模型,奠定了关系数据库的理论基础。

数据库系统阶段出现了统一管理数据的专门软件系统,即数据库管理系统。数据库系统是一种较完善的高级数据管理方式,也是当今数据管理的主要方式,获得了广泛的应用。

数据库技术在不断地发展。根据数据模型的进展,数据库技术又可以划分为 3 个发展阶段:第一代的网状、层次数据库系统;第二代的关系数据库系统;第三代的以支持面向对象数据模型为主要特征的数据库系统。

第一代数据库系统中,层次数据库的数据模型是有根的定向有序树,网状数据库的数据模型对应的是有向图。这两种数据库具有如下共同点:

(1) 支持三级模式（外模式、模式、内模式）。保证数据库系统具有数据与程序的物理独立性和一定的逻辑独立性。

(2) 用存取路径来表示数据之间的联系。

(3) 有独立的数据定义语言。

(4) 导航式的数据操纵语言。

这两种数据库奠定了现代数据库发展的基础。

第二代数据库的主要特征是支持关系数据模型（数据结构、关系操作、数据完整性）。关系模型具有以下特点：

(1) 关系模型的概念单一，实体和实体之间的联系都用二维表来表示。

(2) 以关系代数为基础，有严格的数学基础。

(3) 数据的物理存储和存取路径对用户不透明。

(4) 关系数据库语言高度非过程化。

第二代数据库是当前主流的数据库系统。

1990 年高级 DBMS 功能委员会发表了“第三代数据库系统宣言”，提出第三代数据库系统主要有以下特征：

(1) 支持数据管理、对象管理和知识管理。

(2) 保持和继承了第二代数据库系统的技术。

(3) 对其他系统开放，支持数据库标准语言，支持标准网络协议，有良好的可移植性、可连接性、可扩展性和互操作性等。

第三代数据库支持多种数据模型（比如关系模型和面向对象的模型），并和诸多新技术相结合（比如分布处理技术、并行计算技术、人工智能技术、多媒体技术等），广泛应用于多个领域，由此也衍生出多种新的数据库技术。

尽管第三代数据库有很多优势，但还是尚未完全成熟的一代数据库系统。

1.1.4 数据库技术的特点

与人工管理、文件系统相比，数据库技术有以下特点。

1.1.4.1 数据结构化

数据结构化是数据库与文件系统的根本区别。在数据库系统中的数据彼此不是孤立的，数据与数据之间相互关联，在数据库中不仅要能够表示数据本身，还要能够表示数据与数据之间的联系，这就要求按照某种数据模型，将各种数据组织到一个结构化的数据库中。

1.1.4.2 数据共享性高、冗余度低

数据库系统从整体角度看待和描述数据，数据不再面向某个应用程序而是面向整个系统，所有用户可以同时存取数据库中的数据，使得数据共享性高，数据的共享减少了不必要的数据库冗余，节约了存储空间，同时也避免了数据之间的不相容性与不一致性。

1.1.4.3 数据独立性高

数据的独立性有两方面的含义，一个指的是数据与程序的逻辑独立性，另一个指的是数据与程序的物理独立性。

数据与程序的逻辑独立性是指当数据的总体逻辑结构改变时，数据的局部逻辑结构

不变, 由于应用程序是依据数据的局部逻辑结构编写的, 所以应用程序不必修改, 从而保证了数据与程序间的逻辑独立性。例如, 在原有的记录类型之间增加新的联系, 或在某些记录类型中增加新的数据项, 把原有记录类型拆分成多个记录类型等均可保持数据的逻辑独立性。

数据与程序的物理独立性是指当数据的存储结构改变时, 数据的逻辑结构不变, 从而应用程序也不必改变。例如, 改变存储设备和增加新的存储设备, 或改变数据的存储组织方式, 改变存取策略等均可保持数据的物理独立性。

在数据库系统阶段有较高的数据与程序的物理独立性和一定程度的数据与程序的逻辑独立性。数据的组织和存储方法与应用程序互不依赖、彼此独立的特性可降低应用程序的开发代价和维护代价, 大大减轻了程序员和数据库管理员的负担。

1.1.4.4 数据由 DBMS 集中管理

数据库为多个用户和应用程序所共享, 对数据的存取往往是并发的, 即多个用户可以同时存取数据库中的数据, 甚至可以同时存取数据库中的同一个数据, 为确保数据库数据的正确有效和数据库系统的有效运行, 数据库管理系统提供下述 4 个方面的数据控制功能:

(1) 数据的安全性控制。数据库要有一套安全机制, 使每个用户只能按规定对某些数据以指定方式进行访问和处理, 以便有效地防止数据库中的数据被非法使用和修改, 以确保数据的安全和机密。例如, 系统提供口令检查或其他手段来验证用户身份, 防止非法用户使用系统; 也可以对数据的存取权限进行限制, 只有通过检查后才能执行相应的操作。

(2) 数据的完整性控制。系统通过设置一些完整性规则以确保数据的正确性、有效性和相容性。即将数据控制在有效的范围内, 或要求数据之间满足一定的关系。

(3) 并发控制。数据库中的数据是共享的, 并且允许多个用户同时使用相同的数据。这就要保证各个用户之间不相互干扰, 对数据的操作不发生矛盾和冲突, 数据库能够协调一致, 因此必须对多用户的并发操作加以控制和协调。

(4) 数据恢复。计算机系统的硬件故障、软件故障、操作员的失误以及故意破坏都会影响数据库中数据的正确性, 甚至造成数据库部分或全部数据的丢失。因而 DBMS 需要有一套备份/恢复机制, 以保证当数据遭到破坏时将数据库从错误状态恢复到最近某一时刻的正确状态, 并继续可靠地运行。

1.1.5 关系模型

人们希望开发一个能对上述类型的表格进行维护、查询的通用程序, 那么要做的第一件事就是把具体的表格抽象为一般的表格。抽象是从一个具体问题到一般问题研究的基本方法。于是人们引入数据模型的概念: 数据模型是对现实世界数据特征的抽象, 数据模型通常由数据结构、数据操作和数据约束三部分组成。

通过对现实的各种需求的抽象和总结, 人们总结出很多适合不同需求的数据模型, 它们有关系模型、网状模型、层次模型和面向对象模型等, 而其中的关系数据模型能满足绝大多数应用问题的需要。相关的理论和技术在以往的几十年里迅速发展, 目前已经相当成熟, 几乎所有的主流数据库产品, 如 Oracle、Informix、SQL Server 都是基于关

系模型的,下面就关系模型的数据结构、数据操作和数据约束做简要介绍。

1.1.5.1 关系模型的数据结构

用二维表格数据来表示实体及实体之间联系的模型叫关系模型,一个表就是一个关系。确定一个二维表的结构就是要确定以下两个内容。

(1) 列的组成以及每一列的数据类型。与收集学生信息之前必须设计一个包含表头的空白表格一样,在使用二维表之前,必须首先确定二维表由哪些列组成,每个列的最多的字符数或数字长度,又由于在计算机内数字、字符和日期信息其存储方式及提供的运算是不同的,因此还要确定每一列的数据类型。

能为二维表的每一列确定唯一的数据类型,是由二维表每一列必须是同质的要求所保证的。

(2) 能唯一确定行的一个或一组列。在手工制作二维表的过程中,事实上会不自觉地遵守着一个法则,那就是不允许出现相同行。也就是说,如果表格中出现了两个完全相同的行,可能是两种情况造成的,一种情况是同一对象的信息重复输入了,那显然是个错误;另一种情况是不同对象在表格中反映出完全一样的信息,难分彼此,那显然是表格设计的缺陷。所以规定二维表中不能出现完全相同的行是合理的,并且是必须的。

为了确保二维表中不出现完全相同的行,如果在增加和修改每一行时,把该行数据与二维表中其他行的数据逐一比较,当二维表横向数据和纵向数据很多时,其工作量和效率是可想而知的。而事实上,大多数情况下,只需要确保二维表的某些列的组合其值不重复就可以了。

在没有相同行的条件下,可以确保这些能标识整个行的列是存在的,因为不存在相同行的另一个等价的表述是二维表所有列的列值组合能确定并且只能确定二维表中的一行,然后在所有列的组合中用逐个剔除的方法可以得到某些列的组合,使其满足:①这些列值能唯一确定表中的一行。②去掉任何一个列,剩下的列的列值不能唯一确定表中的一行。

我们把这些列的组合称为二维表的候选码,其中要满足的第一个条件称为候选码的唯一性,第二个条件称为候选码的最小性。一个二维表可能有多个候选码,属于任一候选码的列称为主属性,不属于任一候选码的列称为非主属性。从候选码中可以任选一个设定为二维表的主码(或称主键),设定了主码就可以通过确保主码的唯一性而避免表中出现完全相同的行。

以上的阐述事实上同时论证了二维表候选码的存在性和为二维表设定主码的必要性。

注意: 候选码的唯一性是基于语义的,即随语义环境的变化而变化,在不可能出现重名的情况下,学生姓名可以作为候选码;在可能出现重名但能确保重名的学生一定不会同年同月同日生的情况下,姓名虽然不能作为候选码,但列组合(姓名,出生日期)可以作为候选码。

在实际的数据库设计中,出于系统运行效率上的考虑,应该尽可能地避免用过多的列构成主码,在找不到合适的主码时,可以人为地增加一个流水号作为二维表的主码,这是以空间换时间的有效方法。

1.1.5.2 关系模型的数据操作

手工情况下对表格操作可分为两种:表格信息的维护(行的增加、修改和删除)和

表格信息的查询。

关系模型的数据操作事实上就是对表的操作，对表的操作是通过表的行操作来实现的。关系模型的数据操作分两部分内容：一部分是表的查询操作，另一部分是对表的更新操作。更新操作又包括行的插入（Insert）、删除（Delete）和修改（Update）。

更新操作往往依赖于查询操作，删除和修改行时首先要确定删除和修改哪些行，即需要把符合条件的行“查询”出来，而插入操作有时要插入的行就是一个查询的结果，所以关系操作的核心内容是查询。

关系模型仅给出了关系操作应达到的目标，不同的数据库管理系统可以用不同的方法实现这些目标，目前普遍采用的是 SQL 语言，用 SQL 语言可以实现所有的关系操作，但并不和关系操作一一对应，不同的数据库管理系统对 SQL 有各自不同的功能上的扩展。

下面主要介绍关系模型数据操作中的核心操作，即查询操作。

关系的查询操作也就是关系代数中的关系运算，这些运算包括：选择（Select）、投影（Project）、连接（Join）、并（Union）、交（Except）、差（Intersection）、除（Divide）和广义笛卡儿积。

所有运算的对象都是关系（表），而运算结果也是一个关系（表）。

（1）选择运算。通俗地讲，选择运算就是选行运算。从整个学校学生的名册中取出某个班级学生信息就是选择，即从整个表中选出符合条件的行。

当表中数据非常多时，选择的效率是必须解决的技术问题。设想在手工操作的情况下，要你从包含 1 万名学生信息的资料中寻找某个学号的学生信息，如果资料没有按学号排序，那是不可想象的；反之，如果学生信息是按学号排序了，查找就变得非常快捷！可以想象一下两者的差距是何等的巨大。对计算机也一样，通常建立索引可以极大地提高选择效率，所有的数据库管理系统都提供了各种为表建立索引的方法。

（2）投影。通俗地讲，投影运算就是选列运算。从一个包含数十项（列）内容的个人档案中选取本次查询所关心的内容（列）就是投影运算，即从整个表中选出若干个列。

从数学角度看，二维表是 n 维笛卡儿积的一个子集，即为 n 维空间的一个子空间，选列的实质是把 n 维空间投影到 m ($m \leq n$) 维空间上，投影的名称由此而来。

手工情况下，在表格中已有数据的情况下若要增加列，往往由于纸张宽度的限制，要重新做表，然后把原表格数据抄入已增加新列的新表中，导致这种重复的工作的原因是由于表格设计时考虑不周。同样，关系模型的设计也要避免这种情况的发生，在设计阶段，每一个表要尽可能地包含所有需要的信息，尽管这些信息并不是在所有场合都需要，但由于有投影运算，可以在不同的场合输出不同的信息。

（3）广义笛卡儿积。有时对表格数据的查阅可能要同时比对着查阅多个表格，比如在手工情况下，我们通常把学生基本信息和每个学生各门课的成绩分成两个表，如果要求查询男生中成绩最好的学生信息，就必须同时查询两张表。

广义笛卡儿积就是把多张表组合在一起查询。

多个表的行的所有组合构成这些表的广义笛卡儿积，“所有组合”的特性和笛卡儿积相同，不同的是笛卡儿积组合的对象是各个域的值，而广义笛卡儿积组合的对象是各个表的行。广义笛卡儿积的示例如表 1-1~表 1-3 所示。