

Access 2010 数据库 应用教程（第2版）

程凤娟 赵玉娟 主 编
卫权岗 李 浩 副主编

清华大学出版社

北 京

内 容 简 介

本书是以教育部高等学校计算机基础课程教学指导委员会制定的计算机基础课程教学的基本要求为指导,从数据库的基础理论开始,由浅入深、循序渐进地介绍了 Access 2010 数据库各种对象的功能及创建,最后通过一个完整的 Access 2010 数据库应用系统开发案例将全书的内容贯穿起来。全书共分 9 章,内容分别为数据库系统概述、Access 2010 入门、表、查询、窗体、报表、宏、模块与 VBA 编程、教学管理系统的开发。配套的《Access 2010 数据库应用教程学习指导(第 2 版)》(ISBN 978-7-302-52158-7)提供多种类型的实验案例,适合实践课堂教学。

本书既可作为高等院校数据库应用技术课程的教材,也可作为全国计算机等级考试二级 Access 的培训教材或参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Access 2010 数据库应用教程 / 程凤娟, 赵玉娟 主编. —2 版. —北京: 清华大学出版社, 2019
ISBN 978-7-302-52163-1

I. ①A… II. ①程… ②赵… III. ①关系数据库系统—高等学校—教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2019)第 003547 号

责任编辑: 王 定

封面设计: 孔祥峰

版式设计: 思创景点

责任校对: 牛艳敏

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市龙大印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 19 字 数: 439 千字

版 次: 2015 年 2 月第 1 版 2019 年 1 月第 2 版 印 次: 2019 年 1 月第 1 次印刷

定 价: 58.00 元

产品编号: 082070-01

前 言

目前,数据库技术与应用已成为高等院校非计算机专业必修的计算机基础教学的典型核心课程。Access 是一个关系型数据库管理系统,是 Microsoft Office 的组件之一,它可以有效地组织、管理和共享数据库中的数据,并将数据库与 Web 结合在一起。

本书在写作模式上吸取了国内外优秀教材重视案例教学的优点,以应用为目的、以案例贯穿始终,系统详细地介绍了数据库的基础理论知识、Access 2010 数据库中的各种对象和 VBA 编程等内容,最后还给出教学案例的开发过程,为读者模仿、修改、拓展、延伸和创新提供了原型。本书努力将知识传授、能力培养和素质教育融为一体,实现理论教学与实践教学的完美结合。配套的《Access 2010 数据库应用教程学习指导(第 2 版)》(ISBN 978-7-302-52158-7)提供多种类型的实验案例,适合实践课堂教学。

全书共 9 章,各章内容安排如下:

第 1 章主要介绍数据库系统的基础知识和理论。

第 2 章主要介绍 Access 2010 的基础操作,包括数据库的创建、打开与关闭、管理及数据库对象的导入、导出、复制、删除等操作。

第 3 章主要介绍 Access 2010 中表的操作,包括表的创建、表的基本操作及表间关系等。

第 4 章主要介绍 Access 2010 中的查询及其应用,包括各类查询的创建、SQL 语句等。

第 5 章主要介绍 Access 2010 中的窗体设计及应用,包括窗体的创建、窗体和控件的属性设置等。

第 6 章主要介绍 Access 2010 中的报表设计及编辑,包括报表的创建、在报表中进行计算和统计等。

第 7 章主要介绍 Access 2010 中的宏及其应用,包括宏的创建、调试和运行方法等。

第 8 章主要介绍 Access 2010 中的模块和 VBA 编程,包括模块的创建、VBA 程序的基本结构、过程的创建等。

第 9 章主要介绍教学案例“教学管理系统”的开发过程。

书中加星号*的章节属于能力提升部分,不要求读者掌握。

由于时间仓促,书中难免存在疏漏和不妥之处,敬请各位读者和专家批评指正。

本书课件、习题参考答案和教学案例“教学管理系统”下载:



课 件



习题参考答案



教学管理系统

作 者
2018 年 10 月

目 录

第 1 章 数据库系统概述	1	2.3 数据库的打开与关闭	32
1.1 数据库系统基本概念	1	2.3.1 打开数据库	32
1.1.1 数据和信息	1	2.3.2 关闭数据库	32
1.1.2 数据管理技术的产生与发展	2	2.4 管理数据库	32
1.1.3 数据库系统的组成	4	2.4.1 压缩和修复数据库	32
1.1.4 数据库系统的结构	6	2.4.2 备份与还原数据库	33
1.2 数据模型	8	2.4.3 加密数据库	34
1.2.1 基本概念	8	2.4.4 数据库版本的转换	35
1.2.2 组成要素	8	2.5 数据库对象的基本操作	36
1.2.3 数据模型的层次类型	9	2.5.1 Access 2010 的数据库对象	36
1.3 关系数据库	14	2.5.2 数据库对象的导入	37
1.3.1 基本概念	14	2.5.3 数据库对象的导出	37
1.3.2 常用术语	14	2.5.4 复制数据库对象	37
1.3.3 关系完整性	15	2.5.5 删除数据库对象	38
1.3.4 关系运算	16	2.5.6 重命名数据库对象	38
1.4 数据库设计基础	18	2.6 习题	38
1.4.1 数据库设计原则	18	2.6.1 简答题	38
1.4.2 数据库设计步骤	18	2.6.2 操作题	39
1.4.3 数据库设计范式	19	第 3 章 表	40
1.5 习题	21	3.1 表设计概述	40
1.5.1 简答题	21	3.1.1 表的设计原则	40
1.5.2 选择题	21	3.1.2 表的构成	41
1.5.3 填空题	22	3.1.3 数据类型	42
第 2 章 Access 2010 入门	23	3.1.4 字段属性	43
2.1 Access 2010 概述	23	3.1.5 “教学管理系统”数据库的表 结构设计实例	45
2.1.1 Access 2010 的特点	23	3.2 创建表	47
2.1.2 Access 2010 的启动与退出	25	3.2.1 使用数据表视图创建	47
2.1.3 Access 2010 的工作界面	25	3.2.2 通过导入创建	49
2.2 Access 2010 数据库的创建	30	3.2.3 使用设计视图创建	51
2.2.1 使用模板创建数据库	30	3.2.4 修改表的结构	52
2.2.2 创建空数据库	31		

3.2.5 设置和取消表的主键	54	4.5 交叉表查询	89
3.3 表的基本操作	55	4.5.1 使用向导创建	89
3.3.1 打开和关闭表	55	4.5.2 使用设计视图创建	91
3.3.2 选定和添加记录	56	4.6 操作查询	92
3.3.3 修改和删除记录	57	4.6.1 生成表查询	92
3.3.4 数据的排序与筛选	57	4.6.2 更新查询	93
3.3.5 数据的查找与替换	59	4.6.3 追加查询	94
3.3.6 表的索引	60	4.6.4 删除查询	95
3.3.7 设置表的外观	60	4.7 SQL 查询	96
3.4 表间关系	62	4.7.1 SQL 概述	96
3.4.1 建立表间关系	62	4.7.2 数据查询语句	96
3.4.2 查看和编辑表间关系	64	4.7.3 单数据源查询	98
3.4.3 参照完整性和级联规则	65	4.7.4 多数据源查询	103
3.5 习题	65	*4.7.5 嵌套查询	105
3.5.1 简答题	65	*4.7.6 联合查询	107
3.5.2 选择题	66	4.8 其他 SQL 语句	108
3.5.3 填空题	66	4.8.1 数据定义语句	108
3.5.4 操作题	66	4.8.2 数据更新语句	110
第 4 章 查询	68	4.9 习题	112
4.1 查询概述	68	4.9.1 简答题	112
4.1.1 查询的功能	68	4.9.2 选择题	112
4.1.2 查询的类型	69	4.9.3 填空题	114
4.1.3 查询的视图	70	4.9.4 操作题	115
4.1.4 创建查询的方法	70	第 5 章 窗体	116
4.2 查询条件	72	5.1 窗体概述	116
4.2.1 常量和变量	72	5.1.1 窗体的功能	116
4.2.2 函数	72	5.1.2 窗体的构成	116
4.2.3 运算符	76	5.1.3 窗体的类型	117
4.3 选择查询	77	5.1.4 窗体的视图	119
4.3.1 使用向导创建	77	5.2 创建窗体	120
4.3.2 使用设计视图创建	82	5.2.1 使用按钮创建	121
4.3.3 运行和修改查询	84	5.2.2 使用向导创建	123
4.3.4 在查询中使用计算	85	5.2.3 使用设计视图创建	124
4.4 参数查询	87	5.2.4 使用数据透视表创建窗体	128
4.4.1 单参数查询	87	5.2.5 使用数据透视图创建窗体	129
4.4.2 多参数查询	88	5.2.6 主/子窗体	130

5.3 窗体控件	134	6.4.1 页面设置	185
5.3.1 控件概述	134	6.4.2 打印预览和打印	186
5.3.2 “属性表”对话框	136	6.5 习题	187
5.3.3 常用控件的使用	139	6.5.1 简答题	187
5.3.4 事件与事件过程	146	6.5.2 选择题	187
5.4 美化窗体	148	6.5.3 填空题	188
5.4.1 使用主题	148	6.5.4 操作题	189
5.4.2 设置窗体的布局和格式	148	第 7 章 宏	190
5.4.3 页眉和页脚的美化	149	7.1 宏概述	190
5.5 创建用户界面窗体	151	7.1.1 宏的概念	190
5.5.1 创建导航窗体	151	7.1.2 宏的构成	191
5.5.2 设置启动窗体	153	7.1.3 宏的类型	191
5.6 习题	153	7.1.4 宏的设计视图	192
5.6.1 简答题	153	7.1.5 常用的宏操作	192
5.6.2 选择题	153	7.2 创建宏	194
5.6.3 填空题	154	7.2.1 基本宏	194
5.6.4 操作题	154	7.2.2 条件宏	195
第 6 章 报表	155	7.2.3 宏组	198
6.1 报表概述	155	7.2.4 自动运行宏 AutoExec	199
6.1.1 报表的功能	155	7.3 宏的编辑	200
6.1.2 报表的构成	156	7.3.1 添加宏操作	200
6.1.3 报表的类型	157	7.3.2 删除宏操作	200
6.1.4 报表的视图	158	7.3.3 移动宏操作	200
6.2 创建报表	159	7.3.4 复制宏操作	200
6.2.1 使用“报表”按钮创建	160	7.3.5 添加注释	200
6.2.2 使用报表向导创建	160	7.4 宏的调试与运行	201
6.2.3 使用报表设计视图创建	162	7.4.1 调试宏	201
6.2.4 创建标签报表	165	7.4.2 运行宏	202
6.2.5 创建空报表	167	7.5 习题	204
6.3 报表的高级设计	168	7.5.1 简答题	204
6.3.1 报表的编辑	168	7.5.2 选择题	204
6.3.2 排序、分组和汇总	170	7.5.3 填空题	205
6.3.3 利用计算控件实现计算	178	7.5.4 操作题	205
6.3.4 子报表	181	第 8 章 模块与 VBA 编程	206
6.3.5 导出报表	183	8.1 模块概述	206
6.4 打印报表	185	8.1.1 模块的概念	206

8.1.2	模块的类型	207	8.7	VBA 程序调试和错误处理	248
8.1.3	模块的构成	208	8.7.1	程序调试	248
8.1.4	VBA 编程与宏	208	8.7.2	错误类型	251
8.1.5	将宏转换为模块	209	8.7.3	错误处理	251
8.2	创建模块	209	8.8	习题	253
8.2.1	VBA 概述	209	8.8.1	简答题	253
8.2.2	VBA 编程环境	210	8.8.2	选择题	253
8.2.3	创建模块	212	8.8.3	填空题	254
8.3	面向对象程序设计语言	213	8.8.4	操作题	254
8.3.1	基本概念	213	第 9 章	教学管理系统的开发	256
8.3.2	面向对象的语法与 关键字 Me	215	9.1	管理信息系统的一般开发 过程	256
8.3.3	Access 对象模型	215	9.2	“教学管理系统”的系统 规划	258
8.4	VBA 编程基础	217	9.3	“教学管理系统”的系统 分析	259
8.4.1	数据类型	217	9.4	“教学管理系统”的系统 设计	260
8.4.2	常量和变量	217	9.4.1	功能模块设计	260
8.4.3	数组	219	9.4.2	数据库设计	261
8.4.4	运算符和表达式	221	9.5	“教学管理系统”的系统 实施	271
8.4.5	常用内部函数	223	9.5.1	查询的设计与实现	271
8.5	VBA 程序流程结构	227	9.5.2	窗体的设计与实现	273
8.5.1	VBA 语句的书写规则	228	9.5.3	报表的实现	283
8.5.2	VBA 常用语句	228	参考文献		293
8.5.3	流程控制语句	229			
8.5.4	GoTo 语句	240			
8.6	过程与参数传递	240			
8.6.1	过程声明	240			
8.6.2	过程调用	243			
8.6.3	参数传递	244			

第1章 数据库系统概述

数据库技术产生于 20 世纪 60 年代末期，它是现代信息科学与技术的重要组成部分，是计算机数据处理与信息管理的核心。数据库技术是通过研究数据库的结构、存储、设计、管理和应用的基本实现方法，并利用这些理论来实现对数据库中的数据进行处理、分析和理解的技术。简而言之，数据库技术是研究、管理和应用数据库的一门软件科学。

本章将介绍数据库、数据库系统、数据模型、关系数据库及其运算等知识。

【学习要点】

- 数据库系统及其组成
- 数据库模型
- 关系数据库

1.1 数据库系统基本概念

1.1.1 数据和信息

数据是描述现实世界中各种事物的物理符号记录，是最原始的、彼此分散孤立的、未被加工处理过的记录，其具体的表现形式有数字、字母、文字、图形、图像、动画、声音等。在计算机系统中，一切能被计算机接收和处理的物理符号都称为数据。

信息是对现实世界中事物运动状态和特征的描述，是一种已经被加工为特定形式的数据。信息是对数据的解释，是数据含义的体现。

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。只有经过加工处理，形成的具有使用价值的信息才能称为信息。

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以理解的原始数据中，提炼、抽取出人们所需要的有价值、有意义的信息(即信息)，作为科学决策的依据。

可用下式简单地表示数据、信息与数据处理的关系：

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

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、信息和数据处理的关系如图 1-1 所示。

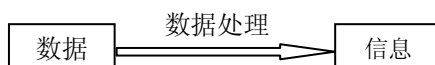


图 1-1 数据、信息和数据处理的关系

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 数据管理技术的产生与发展

数据管理技术就是数据库技术，是对数据进行收集、分类、组织、编码、存储、检索和维护等一系列活动的总和，是数据处理的核心问题。数据管理技术经历了人工管理、文件系统和数据库系统三个阶段，每个阶段的发展都以数据存储冗余不断减小、数据独立性不断增强、数据操作更加方便简单为标志。

1. 人工管理阶段

这一阶段(20 世纪 50 年代中期以前)的计算机主要用于科学计算。外部存储器只有卡片、纸带、磁带，没有磁盘等直接存取存储设备；软件只有汇编语言，没有操作系统，更无统一的数据管理方面的软件；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，即每个应用程序中还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。概括起来，这个阶段有如下特点：

- (1) 用户管理数据。
- (2) 数据不能共享，冗余度极大。
- (3) 数据不独立，完全依赖于程序。
- (4) 数据无结构。

在人工管理阶段，数据和程序之间的关系如图 1-2 所示。

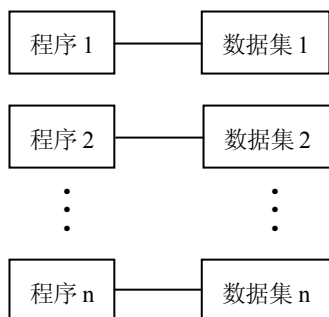


图 1-2 人工管理阶段数据和程序之间的关系

2. 文件系统阶段

这一阶段(20世纪50年代后期至20世纪60年代中期)的计算机不仅用于科学计算,还大量用于信息管理。随着数据量的增加,数据的存储、检索和维护等问题成为紧迫的需要,数据结构和数据管理技术迅速发展了起来。外部存储器有了磁盘、磁鼓等直接存取的存储设备;软件方面出现了高级语言和操作系统。操作系统中的文件系统专门管理外部存储设备中的数据,文件是操作系统管理的重要资源之一,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,也有联机实时处理。文件用户可随时对文件进行查询、修改和增删等处理。这一阶段的特点如下:

- (1) 数据以“文件”的形式可以长期保存。
- (2) 数据的逻辑结构和物理结构有了区别,但比较简单。
- (3) 数据共享性差,冗余度大。
- (4) 数据有了一定的独立性。
- (5) 对数据的操作以记录为单位。

在文件系统阶段,数据和程序之间的关系如图1-3所示。

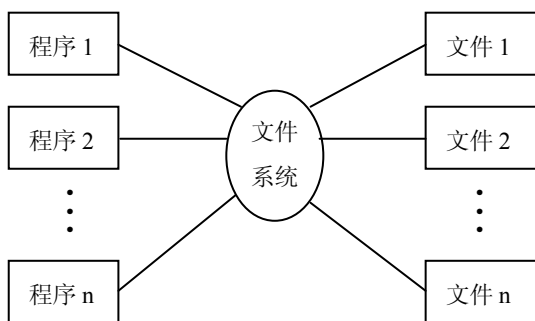


图 1-3 文件系统阶段数据和程序之间的关系

尽管文件系统有了很大的进步,但仍然存在一些缺点,主要体现在以下几个方面:

(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,如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。

3. 数据库系统阶段

这一阶段(20世纪60年代后期以来)的计算机硬件和软件技术得到了飞速发展,计算机

管理的对象规模越来越大,应用范围也越来越广泛,数据量急剧增长,同时多种应用、多种语言相互覆盖地共享数据集合的要求也越来越强烈,数据库技术便应运而生,出现了统一管理数据的专门软件系统,即数据库系统。

与人工管理和文件系统阶段相比较,数据库系统阶段具有以下特点:

- (1) 数据结构化,使用数据模型描述。
- (2) 数据共享性高、冗余度小、易扩充。
- (3) 数据独立性高。
- (4) 统一的数据管理和控制。

在数据库系统阶段,数据和程序之间的关系如图 1-4 所示。

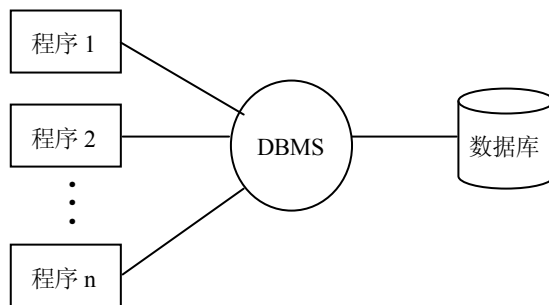


图 1-4 数据库系统阶段数据和程序之间的关系

以上三个阶段的特点对照参见表 1-1。

表 1-1 数据管理三阶段的特点对照

三个阶段 特点	人工管理阶段	文件系统阶段	数据库系统阶段
数据的管理者	用户	文件系统	数据库管理系统(DBMS)
数据面向的对象	某一应用程序	某一应用程序	现实世界(一个部门、企业等)
数据共享程度	无共享,冗余度极大	共享性差,冗余度大	共享性高,冗余度小
数据的独立性	不独立,完全依赖于程序	独立性差	具有高度的物理独立性和一定的逻辑独立性
数据的结构化	无结构	记录内有结构,整体无结构	整体结构化,用数据模型描述
数据控制能力	应用程序自己控制	应用程序自己控制	由 DBMS 提供数据安全性、完整性、并发控制和恢复能力

1.1.3 数据库系统的组成

数据库系统(DataBase System, DBS)是指引进数据库技术后的整个计算机系统。它能够实现有组织地、动态地存储大量相关数据,并能提供数据处理和信息资源共享。

数据库系统一般由数据库、硬件、软件和人员组成。

1. 数据库

数据库(DataBase, DB)是指长期存储在计算机内有组织、可共享的数据集合。数据库中的数据按一定的数学模型组织、描述和存储, 具有较小的冗余、较高的数据独立性和易扩展性, 并可为各种用户共享。

2. 硬件

硬件是指构成计算机系统的各种物理设备, 包括存储所需的外部设备。由于数据库系统承担着数据管理的任务, 它主要在计算机操作系统的支持下工作, 而且包含着数据库管理例程序、应用程序等, 因此要求有足够大的内存空间。同时, 由于用户的数据库管理软件都要保存在外存储器上, 所以对外存储器容量的要求也很高, 还应具有较好的通道性能。简而言之, 硬件的配置应满足整个数据库系统的需要。

3. 软件

软件包括操作系统、数据库管理系统和数据库应用系统。数据库管理系统(DataBase Management System, DBMS)是数据库系统的核心软件, 是在操作系统的支持下工作, 解决如何科学地组织和存储数据, 如何高效获取和维护数据的系统软件。其主要功能包括如下几点:

(1) 数据定义。DBMS 提供数据定义语言(Data Definition Language, DDL), 供用户定义数据库的三级模式结构、两级映象及完整性约束和保密限制等约束。DDL 主要用于建立、修改数据库的库结构。DDL 所描述的库结构仅仅给出了数据库的框架, 数据库的框架信息被存放在数据字典(Data Dictionary)中。

(2) 数据操作。DBMS 提供数据操作语言(Data Manipulation Language, DML), 供用户实现对数据的追加、删除、更新、查询等操作。

(3) 数据库的运行管理。数据库的运行管理功能包括多用户环境下的并发控制、安全性检查和存取限制控制、完整性检查和执行、运行日志的组织管理、事务的管理和自动恢复, 即保证事务的原子性。这些功能保证了数据库系统的正常运行。

(4) 数据组织、存储与管理。DBMS 要分类组织、存储和管理各种数据, 包括数据字典、用户数据、存取路径等, 需确定以何种文件结构和存取方式在存储级上组织这些数据, 如何实现数据之间的联系。数据组织和存储的基本目标是提高存储空间利用率, 选择合适的存取方法, 提高存取效率。

(5) 数据库的保护。数据库中的数据是信息社会的战略资源, 所以数据的保护至关重要。DBMS 对数据库的保护通过 4 个方面来实现: 数据库的恢复、数据库的并发控制、数据库的完整性控制、数据库的安全性控制。DBMS 的其他保护功能还有系统缓冲区的管理和数据存储的某些自适应调节机制等。

(6) 数据库的维护。这部分包括数据库的数据载入、转换、转储、数据库的重组及性能监控等功能, 分别由各个使用程序来完成。

(7) 通信。DBMS 具有与操作系统的联机处理、分时系统及远程作业输入的相关接口, 负责处理数据的传送。网络环境下的数据库系统, 还应该包括 DBMS 与网络中其他软件系

统的通信功能以及数据库之间的互操作功能。

4. 人员

人员主要有三类：数据库开发人员、数据库管理员和最终用户。数据库开发人员包括系统分析员、系统设计员和程序员。系统分析员负责应用系统的需求分析和规范说明，与用户和数据库管理员一起确定系统的硬件配置，并参与数据库系统的概要设计；系统设计员负责数据库中数据的确定、数据库各级模式的设计；程序员负责设计和编写应用系统的程序模块，并进行调试和安装。数据库管理员(DataBase Administrator, DBA)，负责数据库的总体信息控制。DBA的具体职责包括：确定数据库中的信息内容和结构，决定数据库的存储结构和存取策略，定义数据库的安全性要求和完整性约束条件，监控数据库的使用和运行，负责数据库的性能改进、数据库的重组和重构。最终用户通过应用系统的用户接口访问数据库。

数据库系统的组成如图 1-5 所示。

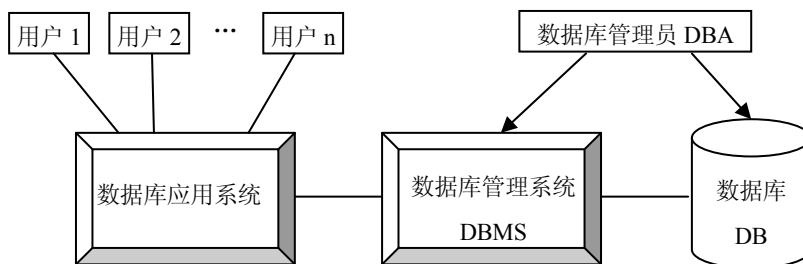


图 1-5 数据库系统的组成

1.1.4 数据库系统的结构

数据库系统的结构是数据库系统的一个总框架，可以从多种不同的角度考查数据库系统的结构。从数据库管理系统的角度看，数据库系统通常采用“外模式-模式-内模式”三级模式结构，如图 1-6 所示。

1. 外模式

外模式(External Schema)又称为子模式(Subschema)或用户模式，它是数据库用户(包括应用程序员和最终用户)能够看到并使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。

外模式通常是模式的子集，一个数据库可以有多个外模式。由于它是各个用户的数据视图，如果不同的用户在应用需求、看待数据的方式、对数据保密的要求等存在差异，则其外模式描述就是不同的。即使对模式中同一数据记录，在外模式中的结构、类型、长度、保密级别等都可以不同。此外，同一个外模式也可为某一用户的多个应用系统所用，但一个应用程序只能使用一个外模式。

外模式是保证数据库安全性的一个有力措施。每个用户只能看见和访问所对应的外模式中的数据，数据库中其余数据是不可见的。

DBMS 提供子模式描述语言(子模式 DDL)来严格定义子模式。

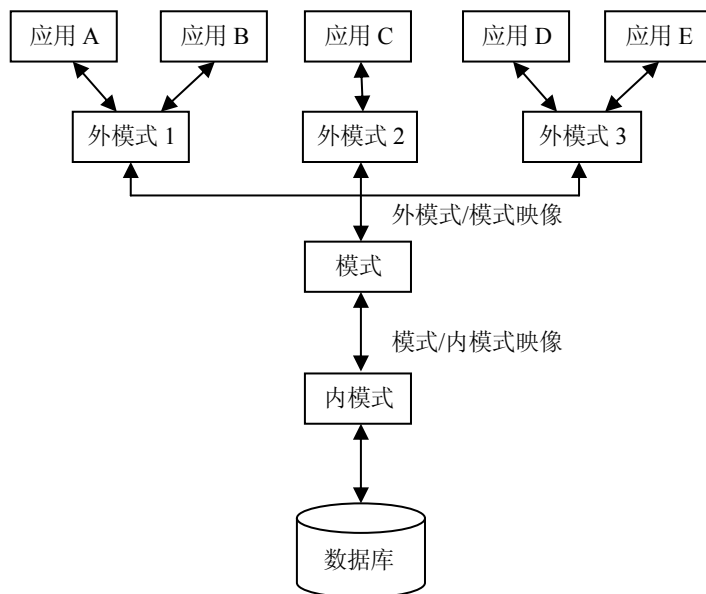


图 1-6 数据库系统的三级模式结构

2. 模式

模式(Schema)又称为概念模式或逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个模式。

模式处于三级结构的中间层，它以某一种数据模型为基础，表示了数据库的整体数据。模式是客观世界某一应用环境中所有数据的集合，也是所有个别用户视图综合起来的结果，又称用户公共数据视图。视图可理解为用户或程序员看到和使用的数据库的内容。

定义模式时不仅要定义数据的逻辑结构(例如数据记录由哪些数据项构成，数据项的名字、类型、长度和取值范围等)，而且要定义与数据有关的安全性、完整性要求，定义这些数据之间的联系。

DBMS 提供模式描述语言(模式 DDL)来描述逻辑模式，严格定义数据的名称、特征、相互关系和约束等。

3. 内模式

内模式(Internal Schema)又称为存储模式(Storage Schema)，是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。一个数据库只有一个内模式。

DBMS 提供内模式描述语言(内模式 DDL，或者存储模式 DDL)来严格定义内模式。

总而言之，内模式处于最底层，它反映数据在计算机物理结构中的实际存储形式；模式处于中间层，它反映设计者的数据全局逻辑要求；外模式处于最外层，它反映用户对数据的要求。

1.2 数据模型

1.2.1 基本概念

数据是对客观事物的符号表示,模型是对现实世界特征的模拟和抽象。数据模型(Data Model)是对数据特征的抽象。

数据库系统的核心是数据库,数据库是根据数据模型建立的,因而数据模型是数据库系统的基础。

计算机不能直接处理现实世界中的客观事物,人们必须把具体事物转换成计算机能够处理的数据。将客观事物转换为数据,是一个逐步转化的过程,经历了现实世界、信息世界和机器世界这三个不同的世界,经历了两级抽象和转化。首先将现实世界中的客观事物抽象为某一种信息结构,这种信息结构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;然后再将概念模型转换为计算机上某一个 DBMS 支持的数据模型,如图 1-7 所示。

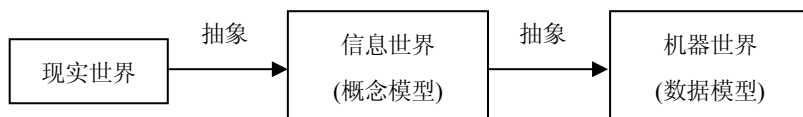


图 1-7 数据抽象过程

1.2.2 组成要素

一般来讲,任何一种数据模型都是严格定义的概念集合。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、数据操作和数据完整性约束三个要素组成。

1. 数据结构

数据结构研究数据之间的组织形式(数据的逻辑结构)、数据的存储形式(数据的物理结构)及数据对象的类型等。存储在数据库中的对象类型的集合是数据库的组成部分。例如,在教学管理中,要管理的数据对象有学生、教师、成绩等基本情况。学生对象集中,每个学生包括学号、姓名、性别、出生日期、政治面貌等信息,这些基本信息描述了每个学生的特性,构成在数据库中存储的框架,即对象类型。

数据结构用于描述系统的静态特性,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,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、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

数据操作用于描述系统的动态特性,是指对数据库中的各种对象(型)的实例(值)允许执

行的操作的集合,包括操作及有关的操作规则。数据库主要有查询和更新(包括插入、删除、修改)两大类操作。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)及实现操作的语言。

3. 数据完整性约束

数据完整性约束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制约和储存规则,用以符合数据模型的数据库状态及状态的变化,以保证数据的正确、有效和相容。

数据模型应该反映和规定本数据模型必须遵守的、基本的、通用的完整性约束。此外,数据模型还应该提供定义完整性约束的机制,以反映具体所涉及的数据必须遵守的特定语义约束。例如,在学生信息中,学生的“性别”只能为“男”或“女”。

数据模型是数据库技术的关键,它的三个要素完整地描述了一个数据模型。

1.2.3 数据模型的层次类型

根据数据抽象的不同级别,可以将数据模型分为三层,即概念数据模型、逻辑数据模型和物理数据模型。

从现实世界到概念模型的转换由数据库设计人员完成;从概念模型到逻辑模型的转换可由数据库设计人员完成,也可用数据库设计工具协助设计人员来完成;从逻辑模型到物理模型的转换一般由 DBMS 完成。

1. 概念数据模型

概念数据模型简称为概念模型或信息模型。概念模型用于信息世界的建模,与具体的 DBMS 无关。为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型,人们常常先将现实世界抽象为信息世界,然后再将信息世界转换为机器世界。

概念模型最常用的表示方法是实体-联系方法(Entity-Relationship approach, E-R 方法),该方法用 E-R 图(E-R diagram)描述概念模型。所以, E-R 方法也称 E-R 模型、E-R 图或实体-联系图等。E-R 方法是由美籍华裔计算机科学家陈品山(Peter Chen)于 1976 年首先提出的。它提供不受任何 DBMS 约束的面向用户的表达方法,是在数据库设计中被广泛用作数据建模的工具。下面对一些基本概念和基本 E-R 数据模型进行介绍。

(1) 基本概念。

① 实体。现实世界中的客观事物称为实体,它是现实世界中任何可区分、可识别的事物。实体可以指人,如教师、学生等,也可以指物,如书、仓库等。它不仅可以指客观存在的事物,也可以指抽象的事件,如演出、足球赛等,还可以指事物与事物之间的联系,如学生选课、客户订货等。

② 属性。每个实体必定具有一定的特征(性质),这样才能以此来区分一个个实体。如教师的编号、姓名等都是教师实体的特征。实体的特征称为属性,一个实体可用若干属性来刻画。属性有“型”和“值”之分,属性型就是属性名及其取值类型,属性值就是属性

在其值域中所取的具体值。如学生实体中的姓名属性中，姓名和取值字符类型是属性型，而“郭莹”是属性值。

③ 实体型。具有相同属性的实体必然具有共同的特征，所以若干个属性型所组成的集合可以表示一个实体的类型，简称实体型，一般用实体名和属性名的集合表示。

④ 实体集。性质相同的同类实体的集合称为实体集，如所有学生、所有课程等。

⑤ 实体之间的联系。实体之间的对应关系称为联系，它反映了现实世界事物之间的相互关联。如学生和课程是两个不同的实体，但学生进行选课时，两者之间就发生了关联，建立了联系。联系的种类分为以下三种：

- 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 中有且只有一个实体与之联系，反之亦然，则称实体集 A 与实体集 B 具有一对一联系。例如，一个班级只有一个班长，一个班长只在一个班级任职，班长与班级之间的联系是一对一的联系。
- 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 中有多个实体与之联系，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 有一对多的联系。例如，一个班级有许多学生，但一个学生只能在一个班级就读，所以班级和学生之间的联系是一对多的联系。
- 多对多联系(N : M)。如果对于实体集 A 中的每一个实体，实体集 B 中有多个实体与之联系，而对于实体集 B 中的每一个实体，实体集 A 中也有多个实体与之联系，则称实体集 A 与实体集 B 之间有多对多的联系。例如，一个学生可以选修多门课程，任何一门课程可以被多个学生选修，所以学生和课程之间的联系是多对多的联系。

(2) E-R 模型。

E-R 模型又称 E-R 图，是由实体集、属性和联系构成，其表示方法如下：

用矩形表示实体集，矩形框内写明实体名；用椭圆表示实体的属性，并用无向边将其与相应的实体型连接起来；用菱形表示实体型之间的联系，在菱形框内写明联系名，并用无向边分别与有关实体型连接起来，同时，在无向边旁标上联系的类型(1 : 1, 1 : N 或 N : M)。

学生和课程的联系对应的 E-R 模型如图 1-8 所示(图中只列出了部分属性)。

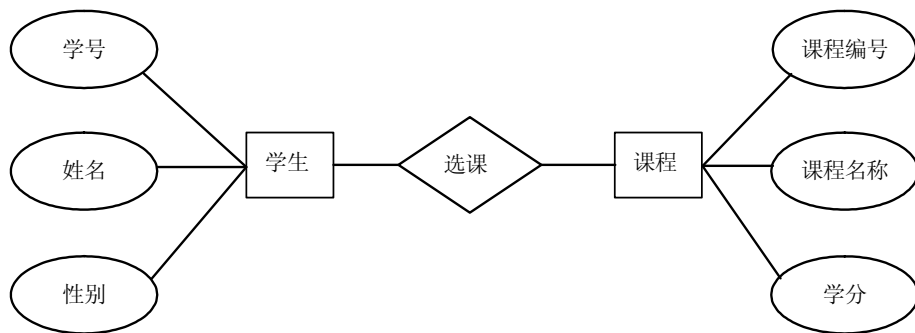


图 1-8 学生和课程的联系对应的 E-R 模型

2. 逻辑数据模型

逻辑数据模型简称为逻辑模型或数据模型。概念数据模型是概念上的抽象，它与具体的 DBMS 无关，而逻辑数据模型与具体的 DBMS 有关，描述数据库数据的整体逻辑结构。

逻辑模型是用户通过 DBMS 看到的现实世界，是按计算机系统的观点对数据建模，即数据的计算机实现形式，主要用于 DBMS 的实现。因此，逻辑模型既要考虑用户容易理解，又要考虑便于 DBMS 的实现。

下面简单介绍三种常用的逻辑数据模型：层次模型、网状模型和关系模型。

(1) 层次模型。

层次模型(Hierarchical Model)是最早出现的数据模型，它是采用层次数据结构来组织数据的数据模型。层次模型可以简单、直观地表示信息世界中实体、实体的属性及实体之间的一对多联系。它使用记录类型来描述实体；使用字段来描述属性；使用结点之间的连线表示实体之间的联系。

层次数据结构也称树型结构，树中的每个结点代表一种记录类型。满足以下两个条件的数据模型称为层次模型：

- 只有一个结点没有双亲结点(双亲结点也称父结点)，该结点称为根结点。
- 根结点以外的其他结点有且只有一个双亲结点。

层次模型可以很自然地表示家族结构、行政组织结构等。图 1-9 所示为使用层次模型表示的某高校的部分组织结构。

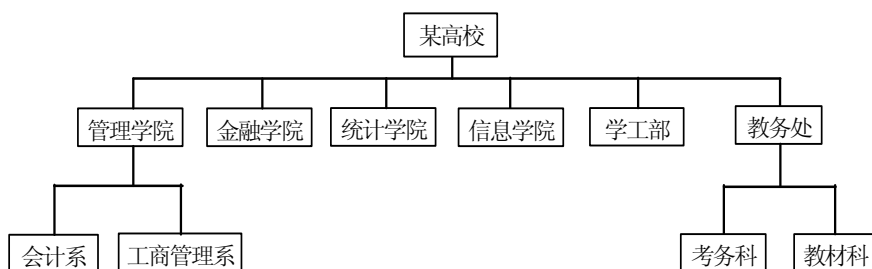


图 1-9 层次模型示例

① 层次模型的三要素。

- 数据结构：使用记录类型表示实体，使用结点之间的连线表示一对多的联系。
- 数据操作：包括结点的查询和结点的更新(如插入、删除和修改)操作。
- 数据完整性约束：一个模型只有一个根结点，其他结点只能有一个双亲结点，结点之间是一对多的联系。

② 层次模型的优点。

- 结构简单、清晰，容易理解。
- 结点之间联系简单，查询效率高。

③ 层次模型的缺点。

- 不能表示一个结点有多个双亲的情况。

- 不能直接表示多对多的联系，需要将多对多联系分解成多个一对多的联系。常用的分解方法是冗余结点法和虚拟结点法。
- 插入、删除限制多。例如，删除父结点则相应的子结点也被同时删除等。
- 必须要经过父结点，才能查询子结点。因为在层次模型中，没有一个子结点的记录值能够脱离父结点的记录值而独立存在。

(2) 网状模型。

网状模型(Network Model)采用网状结构，能够直接描述一个结点有多个父结点，以及结点之间的多对多联系。

网状模型是满足以下两个条件的基本层次联系的集合：

- 允许一个以上的结点无双亲。
- 一个结点可以有多个的双亲。

实际上，层次模型是网状模型的一个特例。网状模型去掉了层次模型中的限制，允许多个结点没有双亲结点，允许结点有多个双亲结点，还允许结点之间存在多对多的联系。使用网状模型可以表示多对多联系。

在教学过程中，学生、教师、课程和教学安排之间的关系可以用网状模型表示，如图 1-10 所示。

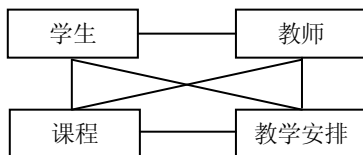


图 1-10 网状模型示例

① 网状模型的三要素。

- 数据结构：使用记录类型表示实体，使用字段描述实体的属性，每个记录类型可包含若干个字段，使用结点之间的连线表示一对多的联系。
- 数据操作：包括结点的查询和结点的更新操作。
- 数据完整性约束：支持码的概念，用于唯一标识记录的数据项的集合；保证一个联系中双亲结点与子结点之间是一对多联系；支持双亲记录和子记录之间的某些约束条件，如只删除双亲结点等。

② 网状模型的优点。

- 网状模型具有良好的性能，存取效率较高。
- 结点之间的联系具有灵活性，能表示事物之间的复杂联系，更适合描述客观世界。

③ 网状模型的缺点。

- 结构复杂，实现网状数据库管理系统比较困难。
- 不容易学习和掌握。
- 应用程序在访问数据时还需要指定存取路径。
- 不支持集合处理，即没有提供一次处理多个记录的功能。

(3) 关系模型。

关系模型(Relational Model)在1970年由IBM公司的E.F.Codd首次提出。关系模型可以描述一对一、一对多和多对多的联系，并向用户隐藏存取路径，大大提高了数据的独立性及程序员的工作效率。此外，关系模型建立在严格的数学概念和数学理论基础之上，支持集合运算。

① 关系模型的三要素。

- 数据结构：实体及实体间的各种联系都用关系来表示，关系用二维表表示。
- 数据操作：可实现数据的更新、查询等操作。
- 数据完整性约束：包括实体完整性、参照完整性和用户定义完整性。

② 关系模型的优点。

- 有严格的数学理论根据。
- 数据结构简单、清晰，用户易懂易用。
- 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建立开发的工作。

③ 网状模型的缺点。

- 查询效率不如非关系模型。
- 增加了开发数据库管理系统的负担。

在关系模型中，实体和实体之间的联系均由关系来表示。关系模型的本质是一张二维表，表1-2所示为“学生”关系示例。

表 1-2 “学生”关系

学号	姓名	性别	出生日期	政治面貌
201801010101	郭莹	女	2001年3月6日	普通居民
201801010102	刘莉莉	女	1999年1月27日	普通居民
201801010103	张婷	女	1999年10月9日	普通居民
201801010104	辛盼盼	女	1999年6月28日	普通居民
201801010105	许一润	男	1999年4月26日	普通居民
201801010106	李琪	女	1999年5月23日	普通居民
201801010107	王琰	女	1999年3月10日	普通居民
201801010108	马洁	女	1999年8月10日	普通居民
201801010109	牛丹霞	女	1999年2月14日	普通居民
201801010110	徐易楠	女	1999年10月28日	普通居民

3. 物理数据模型

物理层的数据抽象称为物理数据模型(Physical Data Model, PDM)，简称为物理模型，它不但由DBMS的设计决定，而且与操作系统、计算机硬件密切相关。物理模型的具体实现是DBMS的任务，数据库设计人员要了解和选择物理模型，一般用户不必考虑物理层的

细节。

物理层是数据抽象的最底层，用于描述数据的物理存储结构和存取方法。例如，数据的物理记录格式是变长还是定长；数据是压缩还是非压缩；一个数据库中的数据和索引(index)是存放在相同的还是不同的数据段上等。索引提供了对包含特定值的数据项的快速访问，建立索引是加快查询速度的有效手段。索引占用一定的存储空间，当基本表更新时，索引需要进行相应的维护，这就增加了数据库的负担，因此要根据实际应用的需求有选择地创建索引。

1.3 关系数据库

1.3.1 基本概念

用关系模型建立的数据库就是关系数据库。关系数据库建立在严格的数学理论基础之上，数据结构简单、易于操作和管理。在关系数据库中，数据被分散到不同的数据表中，每个表中的数据只记录一次，从而避免数据的重复输入，减少数据冗余。Microsoft Access 2010 数据库就是关系数据库。

1.3.2 常用术语

1. 关系

一个关系就是一个二维表，每个关系都有一个关系名。在 Access 中，一个关系可以存储在一个数据库表中，每个表有唯一的表名，即数据表名。

2. 元组

在二维表中，每一行称为一个元组，对应表中一条记录。

3. 属性

在二维表中，每一列称为一个属性，每个属性都有一个属性名。在 Access 数据库中，属性也称为字段。字段由字段名、字段类型组成，在定义和创建表时对其进行定义。

4. 域

属性的取值范围称为域，即不同的元组对于同一属性的取值所限定的范围。例如，性别属性的取值范围只能是“男”或“女”，百分制的成绩属性值应在 0~100 之间。

5. 关键字

关键字是二维表中的一个属性或若干个属性的组合，即属性组，它的值可以唯一地标识一个元组。例如，在学生表中，每个学生的学号是唯一的，它对应唯一的学生，因此，

学号可以作为学生表的关键字，而姓名不能作为关键字。

6. 主键

当一个表中存在多个关键字时，可以指定其中一个作为主关键字，而其他的关键字为候选关键字。主关键字简称为主键。一个关系只有一个主键。

7. 外部关键字

如果一个关系中的属性或属性组并非该关系的关键字，但它们是另外一个关系的关键字，则称其为该关系的外部关键字。

1.3.3 关系完整性

关系完整性是为保证数据库中数据的正确性和相容性，对关系模型提出的某种约束条件或规则。完整性通常包括实体完整性、参照完整性和用户定义完整性，其中实体完整性和参照完整性是关系模型必须满足的完整性约束条件。

1. 实体完整性

实体完整性(Entity Integrity)是指关系的主关键字不能重复也不能取“空值”。

一个关系对应现实世界中一个实体集。现实世界中的实体是可以相互区分、识别的，即它们应具有某种唯一性标识。在关系模式中，以主关键字作为唯一性标识，而主关键字中的属性(称为主属性)不能取空值。否则，表明关系模式中存在着不可标识的实体(因空值是“不确定”的)，这与现实世界的实际情况相矛盾，这样的实体就不是一个完整实体。按实体完整性规则要求，主属性不得取空值，如主关键字是多个属性的组合，则所有主属性均不得取空值。

2. 参照完整性

参照完整性(Referential Integrity)是定义建立关系之间联系的主关键字与外部关键字引用的约束条件。

关系数据库中通常都包含多个存在相互联系的关系，关系与关系之间的联系是通过公共属性来实现的。所谓公共属性，它是一个关系 R(称为被参照关系或目标关系)的主关键字，同时又是另一关系 K(称为参照关系)的外部关键字。那么，参照关系 K 中外部关键字的取值，要么与被参照关系 R 中某元组主关键字的值相同，要么取空值，在这两个关系间建立关联的主关键字和外部关键字引用，符合参照完整性规则要求。如果参照关系 K 的外部关键字也是其主关键字，根据实体完整性要求，主关键字不得取空值，因此，参照关系 K 外部关键字的取值实际上只能取相应被参照关系 R 中已经存在的主关键字值。

在教学管理数据库中，如果将成绩表作为参照关系，学生表作为被参照关系，以“学号”作为两个关系进行关联的属性，则成绩关系通过外部关键字“学号”参照学生关系。

3. 用户定义完整性

实体完整性和参照完整性约束机制,主要是针对关系的主键和外键取值必须有效而给出的约束规则。除了实体完整性和参照完整性约束之外,关系数据库管理系统允许用户定义其他的数据完整性约束条件。用户定义完整性(User Defined Integrity)约束是用户针对某一具体应用的要求和实际需要,以及按照实际的数据库运行环境要求,对关系中的数据所定义的约束条件,它反映的是某一具体应用所涉及的数据必须要满足的语义要求和条件。这一约束机制一般由关系模型提供定义并检验。

用户定义的完整性约束包括属性上的完整性约束和整个元组上的完整性约束。属性上的完整性约束也称为域完整性约束。域完整性约束是最简单、最基本的约束,是指对关系中属性取值的正确性限制,包括关系中属性的数据类型、精度、取值范围、是否允许空值等。例如,在课程(课程编号,课程名称,学分,课程类型)关系中,规定“课程编号”属性的数据类型是字符型,长度为 4 位;“学分”取值在 1~6 之间等。

关系数据库管理系统一般都提供了 NOT NULL 约束、UNIQUE 约束(唯一性)、值域约束等用户定义的完整性约束。例如,在使用 SQL 语言 CREATE TABLE 时,可以用 CHECK 短语定义元组上的约束条件,即元组级的限制,当插入元组或修改属性的值时,关系数据库管理系统将检查元组上的约束条件是否被满足。

1.3.4 关系运算

关系模型中常用的关系操作有查询、插入、删除和修改 4 种。查询操作是在一个关系或多个关系中查找满足条件的列或行,得到一个新的关系;插入操作是在指定的关系中插入一个或多个元组;删除操作是将指定关系中的一个或多个满足条件的元组删除;修改操作是针对指定关系中满足条件的一个或多个元组,修改其数据项的值。

关系代数是关系操作能力的一种表示方式。关系代数是一种抽象的查询语言,也是关系数据库理论的基础之一。

关系代数是通过对关系的运算来表达查询的,关系运算的三要素是运算对象、运算符和运算结果。关系运算的对象是关系,运算的结果也是关系。关系代数运算符通常包括 4 类:集合运算符、专门的关系运算符、算术比较符和逻辑运算符。

按照运算符的不同,将关系代数的操作分为传统的集合运算和专门的关系运算两大类。

1. 传统的集合运算

从集合论的观点来定义关系,将关系看成是若干个具有 K 个属性的元组集合。通过对关系进行集合操作来完成查询请求。传统的集合运算是从关系的水平方向进行的,包括并、交、差和广义笛卡儿积,属于二目运算。

要使并、差、交运算有意义,必须满足两个条件:一是参与运算的两个关系具有相同的属性数目;二是这两个关系对应的属性取自同一个域,即属性的域相同或相容。

(1) 并(Union)。设关系 R 和关系 S 具有相同的目 K, 即两个关系都有 K 个属性, 且相应的属性取自同一个域, 则关系 R 与 S 的并是由属于 R 或属于 S 的元组构成的集合, 并运算的结果仍是 K 目关系。其形式定义如下: $R \cup S = \{t | t \in R \vee t \in S\}$, 其中, t 为元组变量。

(2) 交(Intersection)。设关系 R 和关系 S 具有相同的目 K, 即两个关系都有 K 个属性, 且相应的属性取自同一个域, 则关系 R 与 S 的交是由既属于 R 又属于 S 的元组构成的集合, 交运算的结果仍是 K 目关系。其形式定义如下: $R \cap S = \{t | t \in R \wedge t \in S\}$ 。

交运算可以使用差运算来表示: $R \cap S = R - (R - S)$ 或者 $R \cap S = S - (S - R)$ 。

(3) 差(Difference)。设关系 R 和关系 S 具有相同的目 K, 即两个关系都有 K 个属性, 且相应的属性取自同一个域, 则关系 R 与 S 的差是由属于 R 但不属于 S 的元组构成的集合, 差运算的结果仍是 K 目关系。其形式定义如下: $R - S = \{t | t \in R \wedge t \notin S\}$ 。

进行并、交、差运算的两个关系必须具有相同的结构。对于 Access 数据库来说, 是指两个表的结构要相同。

(4) 广义笛卡儿积(Extended Cartesian Product)。设关系 R 的属性数目是 K1, 元组数目为 M; 关系 S 的属性数目是 K2, 元组数目为 N; 则 R 和 S 的广义笛卡儿积是一个 $(K1+K2)$ 列的 $(M \times N)$ 个元组的集合, 记作 $R \times S$ 。

广义笛卡儿积是一个有序对的集合。有序对的第一个元素是关系 R 中的任何一个元组, 有序对的第二个元素是关系 S 中的任何一个元组。

如果 R 和 S 中有相同的属性名, 可在属性名前加上所属的关系名作为限定。

2. 专门的关系运算

专门的关系运算既可以从关系的水平方向进行运算, 也可以从关系的垂直方向运算, 主要包括选择、投影和连接运算。

(1) 选择(Selection)。选择运算是从关系的水平方向进行运算, 是从关系 R 中选取符合给定条件的所有元组, 生成新的关系, 记作: Σ 条件表达式(R)。

其中, 条件表达式的基本形式为 $X \theta Y$, θ 表示运算符, 包括比较运算符($<$, $<=$, $>$, $>=$, $=$, \neq)和逻辑运算符(\wedge , \vee , \neg)。X 和 Y 可以是属性、常量或简单函数。属性名可以用它的序号或者它在关系中列的位置来代替。若条件表达式中存在常量, 则必须用英文引号将常量括起来。

选择运算是从行的角度对关系进行运算, 选出条件表达式为真的元组。

(2) 投影(Projection)。投影运算是从关系的垂直方向进行运算, 在关系 R 中选取指定的若干属性列, 组成新的关系, 记作: Π 属性列(R)。

投影操作是从列的角度对关系进行垂直分割, 取消某些列并重新安排列的顺序。在取消某些列后, 元组或许有重复, 该操作会自动取消重复的元组, 仅保留一个。因此, 投影操作的结果使得关系的属性数目减少, 元组数目可能也会减少。

(3) 连接(Join)。连接运算从 R 和 S 的笛卡尔积 $R \times S$ 中选取(R 关系)在 A 属性组上的值与(S 关系)在 B 属性组上值满足比较关系 θ 的元组。

在连接运算中有两种最为重要的连接: 等值连接和自然连接。

① 等值连接(Equal Join)。当 θ 为“=”时的连接操作称为等值连接。也就是说，等值连接运算是从 $R \times S$ 中选取 A 属性组与 B 属性组的值相等的元组。

② 自然连接(Natural Join)。自然连接是一种特殊的等值连接。关系 R 和关系 S 的自然连接，首先要进行 $R \times S$ ，然后进行 R 和 S 中所有相同属性的等值比较的选择运算，最后通过投影运算去掉重复的属性。自然连接与等值连接的主要区别是，自然连接的结果是两个关系中的相同属性只出现一次。

1.4 数据库设计基础

数据库设计是数据库技术的主要内容之一。数据库设计是指对于给定的应用环境(包括硬件环境和操作系统、DBMS 等软件环境)，构建一个性能良好的、能满足用户要求的、能够被选定的 DBMS 所接受的数据库模式，建立数据库及应用系统，使之能够有效地、合理地采集、存储、操作和管理数据，满足企业或组织中各类用户的应用需求。

数据库设计的主要内容有数据库的结构特性设计和数据库的行为特性设计。数据库的结构特性设计起着关键作用。

数据库的结构特性是静态的，一般情况下不会轻易变动。因此，数据库的结构特性设计又称为静态结构设计。其设计过程是：先将现实世界中的事物、事物之间的联系用 E-R 图表示，再将各个分 E-R 图汇总，得出数据库的概念结构模型，最后将概念结构模型转换为数据库的逻辑结构模型表示。

数据库的行为结构设计是指确定数据库用户的行为和动作。数据库用户的行为和动作是指数据查询和统计、事物处理及报表处理等，这些都需要通过应用程序来表达和执行，因而设计数据库的行为特征要与应用系统的设计结合进行。由于用户的行为是动态的，所以，数据库的行为特性设计也称为数据库的动态设计。其设计过程是：首先将现实世界中的数据及应用情况用数据流图和数据字典表示，并详细描述其中的数据操作要求，进而得出系统的功能结构和数据库的子模式。

1.4.1 数据库设计原则

为了合理组织数据，应遵循以下的基本设计原则：

- (1) 确保每个表描述的是一个单一的事物。
- (2) 确保每个表都有一个主键。
- (3) 确保表中的字段不可分割。
- (4) 确保在同一个数据库中，一个字段只在一个表中出现(外键除外)。
- (5) 用外键保证有关联的表之间的联系。

1.4.2 数据库设计步骤

考虑数据库及其应用系统开发的全过程，可以将数据库设计过程分为以下 6 个阶段。

(1) 需求分析阶段。进行数据库应用软件的开发，首先必须准确了解与分析用户需求(包括数据处理)。需求分析是整个开发过程的基础，是最困难、最耗费时间的一步。作为地基的需求分析是否做得充分与准确，决定了在其上建造数据库大厦的速度与质量。需求分析做得不好，会导致整个数据库应用系统开发返工重做的严重后果。

(2) 概念结构设计阶段。概念结构设计是整个数据库设计的关键，它通过对用户需求进行综合、归纳与抽象，形成一个独立于具体 DBMS 的概念模型，一般用 E-R 图表示概念模型。

(3) 逻辑结构设计阶段。逻辑结构设计是将概念结构转化为选定的 DBMS 所支持的数据模型，并使其在功能、性能、完整性约束、一致性和可扩充性等方面均满足用户的需求。

(4) 数据库物理设计阶段。数据库的物理设计是为逻辑数据模型选取一个最适合应用环境的物理结构(包括存储结构和存取方法)。即利用选定的 DBMS 提供的方法和技术，以合理的存储结构设计一个高效的、可行的数据库的物理结构。

(5) 数据库实施阶段。数据库实施阶段的任务是根据逻辑设计和物理设计的结果，在计算机上建立数据库，编制与调试应用程序，组织数据入库，并进行系统测试和试运行。

(6) 数据库运行和维护阶段。数据库应用系统经过试运行后即可投入正式运行。在数据库系统运行过程中必须不断地对其进行评价、调整与修改。

开发一个完善的数据库应用系统不可能一蹴而就，它往往是上述 6 个阶段的不断反复。而这 6 个阶段不仅包含了数据库的(静态)设计过程，而且包含了数据库应用系统(动态)的设计过程。在设计过程中，应该把数据库的结构特性设计(数据库的静态设计)和数据库的行为特性设计(数据库的动态设计)紧密结合起来，将这两个方面的需求分析、数据抽象、系统设计及实现等各个阶段同时进行，相互参照，相互补充，以完善整体设计。

1.4.3 数据库设计范式

设计关系数据库时，遵从不同的规范要求，设计出合理的关系型数据库，这些不同的规范要求被称为不同的范式，各种范式呈递次规范，越高的范式数据库冗余越小。

目前关系数据库有 6 种范式：第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、巴德斯科范式(BCNF)、第四范式(4NF)和第五范式(5NF，又称完美范式)。满足最低要求的范式是第一范式(1NF)。在第一范式的基础上进一步满足更多规范要求的称为第二范式(2NF)，其余范式以此类推。一般说来，数据库只需满足第三范式(3NF)就行了。

1. 第一范式(1NF)

在任何一个关系数据库中，1NF 是对关系模式的基本要求，不满足 1NF 的数据库就不是关系数据库。所谓 1NF 是指关系中每个属性都是不可再分的数据项。

例如，表 1-3 所示是不符合第一范式的，而表 1-4 所示是符合第一范式的。

表 1-3 非规范化关系

教师编号	姓名	联系电话	
		固定电话	移动电话
14250	张三	8686888	13901234567

表 1-4 满足 1NF 的关系

教师编号	姓名	联系电话
14250	张三	86868888

2. 第二范式(2NF)

在一个满足 1NF 的关系中,不存在非关键字段对任一候选关键字的部分函数依赖(部分函数依赖是指存在组合关键字中的某些字段决定非关键字段的情况),即所有非关键字段都完全依赖于任意一组候选关键字,则称这个关系满足 2NF。

假定成绩关系为(学号,姓名,年龄,课程名称,成绩,学分),关键字为组合关键字(学号,课程名称),因为存在如下决定关系:

(学号,课程名称)→(姓名,年龄,成绩,学分); (课程名称)→(学分); (学号)→(姓名,年龄)

所以这个数据库表不满足第二范式,原因是存在组合关键字中的字段决定非关键字的情况。

由于不符合 2NF,这个选课关系表会存在如下问题:

(1) 数据冗余。同一门课程由 n 个学生选修,“学分”就重复 $n-1$ 次;同一个学生选修了 m 门课程,姓名和年龄就重复 $m-1$ 次。

(2) 更新异常。若调整了某门课程的学分,数据表中所有行的“学分”值都要更新,否则会出现同一门课程学分不同的情况。

(3) 插入异常。假设要开设一门新的课程,暂时还没有人选修。由于没有“学号”关键字,课程名称和学分无法输入数据库。

(4) 删除异常。假设一批学生已经完成课程的选修,这些选修记录就应该从数据库表中删除。但是,与此同时,课程名称和学分信息也被删除。很显然,这会导致插入异常。

把选课关系表改为如下 3 个表:

学生(学号,姓名,年龄); 课程(课程名称,学分); 成绩(学号,课程名称,成绩)

这样的数据库表是符合第二范式的,消除了数据冗余、更新异常、插入异常和删除异常。

另外,所有单关键字的数据库表都符合第二范式,因为不可能存在组合关键字。

3. 第三范式(3NF)

在一个满足 2NF 的关系中,如果不存在非关键字段对任一候选关键字段的传递函数依赖,则符合第三范式。所谓传递函数依赖,指的是如果存在 $A \rightarrow B \rightarrow C$ 的决定关系,则 C 传递函数依赖于 A 。因此,满足第三范式的关系应该不存在如下依赖关系:

关键字段 \rightarrow 非关键字段 $x\rightarrow$ 非关键字段 y

假定学生关系为(学号,姓名,年龄,所在学院,学院地点,学院电话), 关键字为单一关键字“学号”, 因为存在如下决定关系:

(学号) \rightarrow (姓名,年龄,所在学院,学院地点,学院电话)

所以这个数据库是符合 2NF 的, 但是不符合 3NF, 原因是存在如下决定关系:

(学号) \rightarrow (所在学院) \rightarrow (学院地点,学院电话)

即存在非关键字段“学院地点”“学院电话”对关键字段“学号”的传递函数依赖。它会存在数据冗余、更新异常、插入异常和删除异常的情况。

把学生关系表分为如下两个表:

学生(学号,姓名,年龄,所在学院); 学院(学院,地点,电话)

这样的数据库表是符合第三范式的, 消除了数据冗余、更新异常、插入异常和删除异常。

以上三范式的通俗理解如下。

第一范式: 是对属性的原子性约束, 要求属性具有原子性, 不可再分解。

第二范式: 是对记录的唯一性约束, 要求记录有唯一标志, 即实体的唯一性。

第三范式: 是对字段冗余性的约束, 即任何字段不能由其他字段派生而来, 它要求字段没有冗余。

1.5 习 题

1.5.1 简答题

1. 简述数据库系统的组成。
2. 简述层次模型、网状模型和关系模型的优缺点。
3. 什么是 E-R 图? 构成 E-R 图的基本要素是什么?
4. 简述关系模型的基本术语。
5. 什么是关系的完整性? 试举例说明关系的完整性约束条件。

1.5.2 选择题

1. 下列有关数据库的描述, 正确的是()。
 - A. 数据库设计是指设计数据库管理系统
 - B. 数据库技术的根本目标是要解决数据共享的问题
 - C. 数据库是一个独立的系统, 不需要操作系统的支持
 - D. 数据库系统中, 数据的物理结构必须与逻辑结构一致
2. 在数据库管理技术的发展中, 数据独立性最高的是()。

第2章 Access 2010入门

Access 2010 是 Office 2010 办公软件的一个重要组件，是小型的桌面型关系数据库管理系统，主要用于数据管理。它可以高效地完成各类中小型数据库管理工作，可以广泛应用于财务、金融、统计和审计等多种管理领域，可以大大提高数据处理的效率。它还是特别适合非计算机专业的普通用户开发所需的各种数据库应用系统。

【学习要点】

- Access 2010 的工作界面
- Access 2010 的数据库对象
- 数据库的创建
- 数据库的基本操作

2.1 Access 2010 概述

2.1.1 Access 2010 的特点

Access 2010 不仅继承和发扬了以前版本功能强大、界面友好、易学易用的优点，又有了新的变化。其发生的变化主要有智能特性、用户界面、创建 Web 网络数据功能、新的数据类型、宏的改进和增强、主题的改进、布局视图的改进及生成器功能的增强等方面。这些增加的功能，使得原来十分复杂的数据库管理、应用和开发工作变得更简单、更轻松、更方便；同时更加突出了数据共享、网络交流、安全可靠的特性。

Access 2010 的主要功能和特点如下。

1. 入门更快速方便

利用 Access 2010 的社区功能，不但可以使用以前版本开发的成果，还能以他人创建的数据库模板为基础展开工作。用户可以从社区提交的数据库模板或 Office.com 上提供的数据库模板中选择并进行修改，快速地完成开发数据的具体需要。

2. 面向对象的开发环境

Access 是一个面向对象、采用事件驱动的关系型数据库管理系统，允许用户使用 VBA(Visual Basic for Applications)语言作为应用程序开发工具，通过数据库对象、控件、属性、事件、方法及类、封装、继承、消息、传递等面向对象程序设计机制，实现对数据库应用系统的开发。