21世纪高等学校计算机规划教材

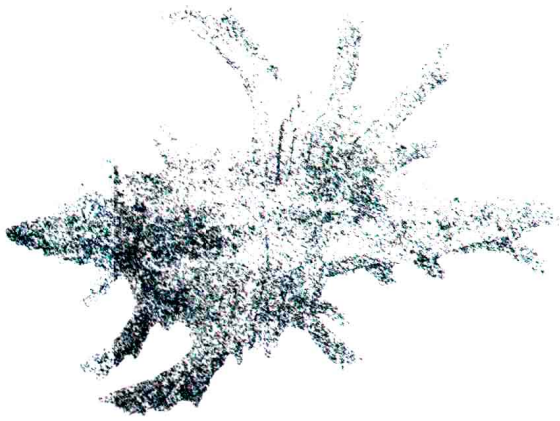
21st Century University Planned Textbooks of Computer Science

数据库应用初级教程

Database Application Elementary Course

朱怀宏 主编

- 本着简单和实用的原则选择教材内容
- 讲解基础知识，强调实践操作
- 避免编程开发，注重实际应用



高校系列

人民邮电出版社
POSTS & TELECOM PRESS

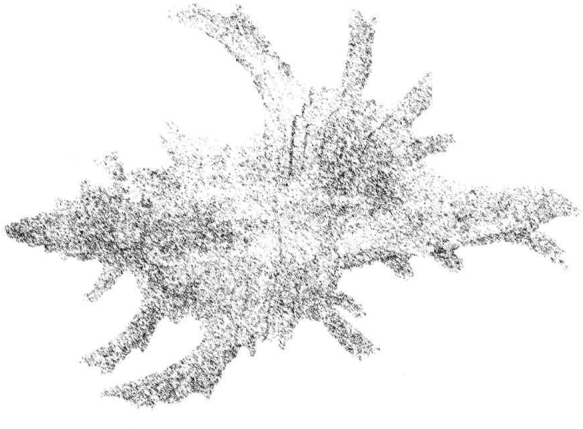
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

数据库应用初级教程

Database Application Elementary Course

朱怀宏 主编



高校系列

人 民 邮 电 出 版 社

北 京

图书在版编目（CIP）数据

数据库应用初级教程 / 朱怀宏主编. -- 北京 : 人民邮电出版社, 2013.2
21世纪高等学校计算机规划教材. 高校系列
ISBN 978-7-115-28901-8

Ⅰ. ①数… Ⅱ. ①朱… Ⅲ. ①数据库系统－高等学校－教材 Ⅳ. ①TP311.13

中国版本图书馆CIP数据核字(2012)第317906号

内 容 提 要

本书为数据库应用初级教程，强调初学者对数据库的初级应用，避免涉及编程、开发等初学者和非计算机专业人员头痛的问题。尤其是对成人教育学生来说，本书能够使他们用较短的时间学会在实际工作中使用数据库。

本书以 Microsoft Access 2003 关系型数据库为背景，所有操作均以 Access 2003 软件为基础。全书由数据库基础知识、Access 2003 数据库简介、建立数据表、设计查询、窗体、报表和数据访问页 7 章组成。

本书适用于非计算机专业和成人教育相关专业开设的数据库初级应用课程，同时也适合需要掌握数据库应用而又不要求编程的自学者。

21 世纪高等学校计算机规划教材-高校系列

数据库应用初级教程

◆ 主　　编　朱怀宏
责任编辑　武恩玉
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷
◆ 开本：787×1092　　1/16
印张：12　　2013 年 2 月第 1 版
字数：312 千字　　2013 年 2 月北京第 1 次印刷

ISBN 978-7-115-28901-8

定价：29.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

目　录

第 1 章　数据库基础知识 ……1

1.1　数据管理的需求及相关技术的发展……1
1.1.1　人工管理阶段……1
1.1.2　文件管理阶段……2
1.1.3　数据库管理阶段……2
1.2　数据库系统……3
1.2.1　数据库的基本概念……3
1.2.2　数据库系统的优点……4
1.3　数据模型……5
1.3.1　客观对象到模型的转换……5
1.3.2　概念模型及其建立……6
1.4　关系数据库……9
1.4.1　数据结构……9
1.4.2　实际应用中的关系模型……11
1.4.3　关系数据模型的完整性……13
1.4.4　关系运算……13
1.5　设计数据库……20
1.5.1　数据库设计的任务和方法……20
1.5.2　数据库的设计原则……20
1.5.3　数据库的设计步骤……21
本章小结……21
习题……22

第 2 章　Access 2003 数据库简介 ……27

2.1　Access 2003 简述……27
2.1.1　Access 2003 的安装……27
2.1.2　Access 2003 的启动与退出……27
2.2　创建数据库……28
2.2.1　利用模板创建数据库……28
2.2.2　直接创建空数据库……30
2.3　打开数据库……30
2.4　关闭数据库……33
2.5　版本的文件格式……33
2.6　设置默认文件夹……34
2.7　数据库窗口中的视图模式……34
2.8　数据库的组对象……34
2.9　Access 2003 数据库对象……36
2.9.1　表……37
2.9.2　查询……38
2.9.3　窗体……38
2.9.4　报表……39
2.9.5　宏……40
2.9.6　数据访问页……40
2.9.7　模块……40
本章小结……41
习题……42

第 3 章　建立数据表……44

3.1　表的组成与创建……44
3.1.1　表的组成……45
3.1.2　Access 2003 中的数据类型……46
3.1.3　创建表……47
3.1.4　复制、重命名及删除表……52
3.2　数据的输入……53
3.2.1　数据输入与编辑……53
3.2.2　导入、导出数据表和链接外部数据表……55
3.3　字段的设置……60
3.3.1　字段名称及类型……61
3.3.2　字段的插入、删除和移动……61
3.3.3　重新设置主键……62
3.3.4　字段的属性设置……62
3.4　建立表之间的关联关系……74
3.4.1　建立表间关系的优越性……74
3.4.2　表间关联关系的建立方法……75
3.4.3　子数据表……76
3.4.4　实施参照完整性……77
3.4.5　编辑表间关系……79
3.4.6　删除表间关系……79

3.4.7　查阅向导……79
3.5　调整表的外观……82
3.5.1　调整列宽……82
3.5.2　调整行高……83
3.5.3　改变字段顺序……83
3.5.4　隐藏字段……83
3.5.5　取消隐藏字段……84
3.5.6　冻结列……85
3.5.7　解冻列……85
3.5.8　设置字体……85
3.5.9　设置数据表格式……86
3.6　记录操作……86
3.6.1　记录排序……86
3.6.2　筛选记录……87
本章小结……90
习题……91

第 4 章　设计查询……97

4.1　概述……97
4.1.1　查询的类型……97
4.1.2　查询条件……98
4.1.3　创建查询的方法……99
4.1.4　查询涉及的视图种类……100
4.1.5　查询与数据表的比较……101
4.2　选择查询……101
4.2.1　使用“查询向导”创建……101
4.2.2　使用“设计视图”创建……105
4.2.3　查询中的计算功能……108
4.2.4　建立多表间的关系……110
4.3　参数查询……111
4.3.1　单参数查询……111
4.3.2　多参数查询……112
4.4　交叉表查询……113
4.4.1　使用交叉表查询向导创建交叉表……113
4.4.2　使用设计视图创建交叉表……115
4.5　操作查询……116
4.5.1　生成表查询……116
4.5.2　追加查询……117
4.5.3　更新查询……118
4.5.4　删除查询……118
本章小结……120
习题……120

第 5 章　窗体……125

5.1　了解窗体……125
5.1.1　窗体的概念……125
5.1.2　窗体的类型……126
5.2　使用向导创建窗体……127
5.2.1　自动创建窗体……127
5.2.2　使用文件另存创建窗体……128
5.2.3　使用窗体向导创建窗体……129
5.2.4　创建图表窗体……131
5.3　使用设计视图创建窗体……133
5.3.1　窗体设计视图的组成……133
5.3.2　窗体设计工具……134
5.4　窗体控件及其使用……136
5.4.1　控件的功能……136
5.4.2　设计视图中针对控件的基本操作……136
5.4.3　控件的使用……137
5.5　窗体的格式化……140
5.5.1　常见的格式属性……140
5.5.2　自动套用格式……140
5.5.3　条件格式……141
本章小结……142
习题……143

第 6 章　报表……147

6.1　报表的基本概念……147
6.1.1　报表的构成……147
6.1.2　报表的类型……148
6.1.3　报表的视图分类……149
6.2　根据系统引导创建报表……149
6.2.1　自动创建报表……150
6.2.2　使用“报表向导”建立报表……150
6.2.3　使用“图表向导”建立图表报表……154
6.2.4　使用“标签向导”建立报表……155
6.3　利用设计视图建立报表……158

6.4 报表的排序与分组 ……159
6.4.1 报表的排序 ……159
6.4.2 报表的分组 ……161
6.5 报表的计算 ……162
6.6 报表的外观处理 ……164
6.6.1 自动套用格式 ……164
6.6.2 添加背景图片 ……165
6.6.3 添加分页符强制分页 ……166
6.6.4 添加日期和时间 ……167
6.7 报表的预览和打印 ……167
6.7.1 页面的设置 ……167
6.7.2 报表的打印预览 ……168
6.7.3 报表的打印 ……168
本章小结 ……169
习题 ……169
第 7 章 数据访问页 ……173
7.1 基本概念 ……173
7.1.1 数据访问页的类型 ……173
7.1.2 数据访问页的视图种类 ……173
7.1.3 数据访问页的打开方式 ……174
7.2 数据访问页的创建 ……174
7.2.1 自动创建数据访问页 ……174
7.2.2 使用向导创建数据访问页 ……175
7.2.3 使用设计视图创建数据访问页 ……177
7.2.4 使用其他对象直接转换到数据访问页 ……179
7.3 数据访问页的编辑 ……180
7.3.1 添加命令按钮 ……180
7.3.2 添加标签 ……181
7.3.3 添加滚动文字 ……181
7.3.4 添加 Office 组件 ……182
本章小结 ……182
习题 ……183
参考文献 ……185

第 1 章 数据库基础知识

随着人类社会的进步和信息技术的发展，人们对复杂数据管理的需求越来越强烈，必须解决在现有计算机系统中如何准确地表示数据，如何有效地采集与组织数据，以及如何高效地存储和处理数据。面对数据量的爆炸式增长，在 20 世纪 60 年代末期，人类发明了有效处理数据的数据库系统，作为信息系统核心和基础的数据库技术在随后的日子里得到了越来越广泛的应用与发展，数据库技术已经成为计算机科学与技术的一个重要研究方向。

本章主要介绍数据库系统的基本概念、特征、组成、数据模型以及关系数据库的一些基本理论知识。

1.1 数据管理的需求及相关技术的发展

实际情况是，一方面数据量越来越大，另一方面相关的处理技术不断提高，两者互相促进、不断发展并有所突破。通常根据相关有代表性的技术，将数据处理或管理人为划分成若干阶段。

1.1.1 人工管理阶段

自 20 世纪 40 年代电子数字计算机问世至 50 年代中期，计算机主要用于科学计算。硬件方面的性能处于初级阶段。没有专门针对数据管理的软件，数据是与其相关程序绑定在一起的，即由专业程序员编制的应用程序与数据不可分离，或者说数据是编制在程序中的。当数据有变动时，必须由程序员去修改程序，数据没有独立性。另一个问题是各程序中自带的数据不能互相传递，没有共享性，各应用程序之间存在大量的重复数据，即数据存在大量冗余。对于存储结构，存取方式、输入输出等都要由相应程序的设计人员自己编制。此阶段的数据不能单独长期保存，它们在程序运行时起作用，当携带数据的程序运行结束退出计算机系统后，它的数据也不起作用了，应用程序和数据之间是一一对应关系，如图 1.1 所示。

图 1.1 人工管理阶段应用程序与数据集合的对应关系

1.1.2 文件管理阶段

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，硬件方面出现大容量的磁带、磁鼓、磁盘外存储器，软件方面出现了高级语言和操作系统，而操作系统的主要功能之一就是有专门进行数据管理的文件系统部分。

在文件管理阶段，程序和数据分开存放于程序文件和数据文件中，而数据文件可以脱离应用程序而独立存放在存储器中，并且可以被多次存取。此时程序若要使用数据，只需用相关文件名去调用其数据，大量数据不必放在程序中了，程序编制者的精力也可集中于算法及程序的高效率上，但是程序和数据之间只能说具有一定的独立性，而不是说数据已经完全独立出来了。此时的数据文件是针对特定应用领域而专门设计的，其相关的应用程序也是与这种特定的文件结构对应的。程序和数据文件相互依赖，如果数据的结构有所改变，必须修改相关的程序，反之程序结构一旦有所变化，也必须修改相应数据的结构，另一个问题是同一个数据可能重复出现在它要被使用的多个文件中，导致数据冗余量大。更为严重的是当一个文件中的某数据被修改时，位于其他文件中的同一个数据不能统一修改，造成数据的不一致性，导致出现错误。应用程序与数据之间的对应关系如图 1.2 所示。

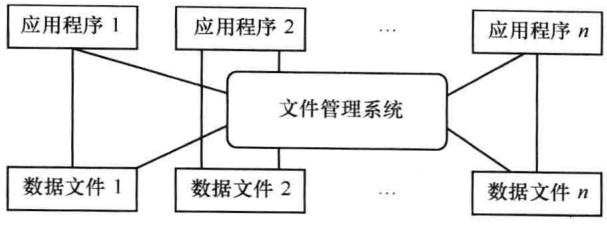


图 1.2 文件管理阶段应用程序与数据之间的关系

1.1.3 数据库管理阶段

到了 20 世纪 60 年代，计算机的应用范围越来越广泛，数据量急剧增加，数据管理变得越来越复杂，人们希望实现共享数据的要求越来越迫切，同时硬件方面已出现大容量的磁盘。在这样的背景下，解决计算机系统中如何准确地表示数据，如何有效地获取与组织数据，以及如何高效地存储和处理数据成为可能。同时，以文件系统作为数据管理的方式已经不能满足时代的需求。在 60 年代后期，人们逐步开发成功了以统一管理和共享数据为主要特征的数据库系统（DataBase System，DBS），进入了数据库管理阶段。在数据库系统中，数据不再仅仅服务于单个程序或用户，而是按一定的结构存储于数据库，成为一种可以被多个程序或用户共享的资源，由称为数据库管理系统（DataBase Management System，DBMS）的特定软件进行管理。在这样的管理方式中，应用程序不再是只能与一个针对它的数据文件相对应，而是由 DBMS 实现多个程序与多个数据文件的对应使用，实现了应用程序灵活方便地对数据的访问和管理，数据与程序之间是一种完全独立的关系，所编程序的质量大大提高；另外各不同的数据文件之间可以建立关联，克服了冗余量大的问题，提高了数据的共享性。应用程序与数据之间的对应关系如图 1.3 所示。

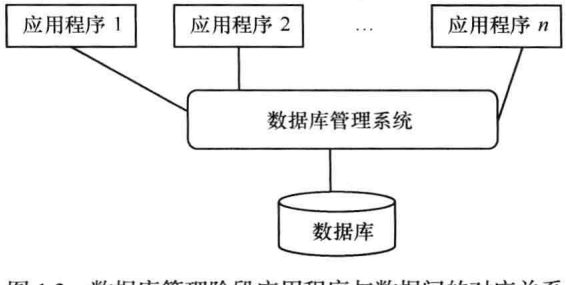


图 1.3 数据库管理阶段应用程序与数据间的对应关系

1.2　数据库系统

1.2.1　数据库的基本概念

数据库的基本概念一般包含数据、信息、数据结构、数据处理、数据库、数据库管理系统、数据库应用系统、数据库管理员、数据库系统等。

1. 数据

数据（Data）是指对事物进行描述的一种符号，是信息的载体。最简单、常见的数据是数字。实际上数据有多种表现形式，比如文字、图形、图像、声音、身份证等都是数据，它们都可以经过数字化转换后存入目前常见的电子数字计算机中进行处理，而对数据库来说数据则是其中存储的基本对象。

2. 信息

到目前为止对于信息（Information）还没有一个唯一、精确的定义，一般把信息描述为经过加工处理的有用数据，信息具有真实性、确定性、共享性、有用性和扩散性。信息通常以数据的形式表示。

3. 数据结构

数据结构指由某一数据元素的集合和该集合中数据元素之间的关系组成。也就是说把某些形式上、含义上类同的、相关的数据集中在一起，规定这些元素之间的关系，并且用特定的形式进行描述。

4. 数据处理

数据处理（Data Processing）通常指数据的采集、存储、检索、变换、加工等方面，在计算机的数据库系统中一般指对信息相关联的数据进行处理，其核心是数据管理。

5. 数据库

数据库（Data Base，DB）类似于工厂存放零配件的仓库，其中各种物品分门别类，按照一定的次序、规则存放，而数据库中存放的是各种数据，是放数据的仓库，它们按照一定的格式存放于计算机系统的存储设备上，同时保存在其中的还有相关数据之间的关系。

比较专业化的解释是，数据库是指可以长时间地存储在计算机系统内的，有组织、可共享的数据集合。通常是为了实现一定的目标，而构建一个数据库，其中的数据按一定的数据模型组织、描述和存放，具有较低的冗余度、较高的数据独立性和易扩展性，它不仅仅是针对某一项指定的应用，而是面向多种应用，可以被多个用户、多个应用程序共享，比如说教学管理数据库中的课表数据可给老师、学生、教室管理部门等多方使用。

6. 数据库管理系统

数据库管理系统（DataBase Management System，DBMS）是指位于用户与操作系统之间的，为数据的建立、使用、管理和维护而编写的数据管理软件。DBMS 通常有以下几个功能。

（1）数据定义，包括定义数据库的结构、有关约束条件等，会提供数据定义语言（Data Definition Language，DDL）对数据库中的数据对象方便地进行定义。

（2）数据操纵，对数据库中的数据进行查询、插入、删除、修改等，会提供数据操纵语言（Data manipulation Language，DML）对数据实现相关的基本操作。

（3）数据库的运行管理。在建立、运行、管理和维护数据库时，对其进行并发控制，安全性检查、约束条件检查、多用户对数据的开发使用、发生故障后的系统恢复、数据库的内部维护等。

（4）为了提高存储空间的利用率和操作效率，对数据进行组织、存储和管理。

（5）数据库的建立和维护。数据的输入和转换，数据库的转储、恢复，数据库的重组与重构、性能的监管与分析等功能。

（6）数据通信的接口，具有与其他软件进行通信的功能。

常见的数据库管理系统有 Access、FoxPro、SQL Server、Oracle 等。

7. 数据库应用系统

针对各种实际应用，计算机软件开发人员利用数据库系统资源可以开发出相应的数据库应用系统（Database Application System，DBAS），比如人事管理系统、教学管理系统、图书管理系统等。这些面向某一方面进行数据管理的应用系统可统称为管理信息系统（Management Information Syestem，MIS）。

8. 数据库管理员

数据库管理员（DataBase Administrator，DBA）是指对某个数据库系统进行全面管理和维护的人员，他们的工作主要包括规定数据库中的数据及结构；决定数据库的存储结构和存储策略；监督、控制数据库的运行和使用；保证数据库的完整性和安全性；作为主要成员，参与数据库的改造、升级和重组等事宜。

9. 数据库系统

数据库系统（DataBase System，DBS）主要由计算机软硬件系统、数据库、数据库管理系统及相关软件、数据库应用系统、数据库管理员和用户组成。数据库系统层次如图 1.4 所示。

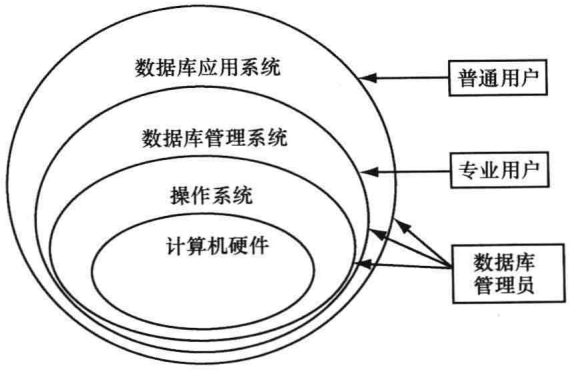


图 1.4　数据库系统层次图

1.2.2　数据库系统的优点

数据库系统具有以下优点。

（1）数据结构化。数据库系统中的数据是面向全局应用的。数据以一定的逻辑结构存放，并采用一定的数据模型来进行描述和定义。数据具有整体结构化的特征，不仅数据内部是结构化的，而且整体也是结构化的，数据之间是关联的，故数据库系统不仅可以表示事物内部各数据项的关系，而且还可以表示事物和事物之间的关系。按照目前的计算机体系结构，只有按一定结构组织和存放的数据，才能实现有效的管理。因此，在说明数据结构时，不但要描述数据本身的特征，同时要描述数据之间的关系。

（2）数据共享性好，冗余度低。数据库系统是从全局分析和描述数据，使得数据可以适应多个用户、多种应用共享数据的需求。也就是说，数据不仅面向某个应用，而且面向整个系统中的用户。数据共享可以明显地减少数据冗余，节省存储空间，即克服了当多个用户使用相同数据时，以前要多次重复存储该数据，现在只需存储一次。数据共享带来的另一大好处是能够避免数据的不相容与不一致。因为以前同一数据被存放在不同的地方，供不同的用户使用，当一个用户修改该数据时，只修改了位于面向此用户存储的数据，而存放在其他地方的该数据还是原来的值，造成不一致。

（3）系统灵活，易于扩充。由于是面向整个系统的结构化数据，数据库系统既有利于系统中多用户共享使用，又便于增加新的应用，可以从整个系统的数据集合中按照用户的需求选取数据子集。当某用户的需求改变或增减时，只需要重新选择新的子集或增减部分数据即可。

（4）具有较高的数据独立性。数据独立性涉及逻辑独立性和物理独立性。数据的逻辑独立性指用户使用的应用程序与数据库的逻辑结构相互独立，系统中数据的逻辑结构改变不应该影响用户的应用程序；而数据的物理独立性指用户使用的应用程序与存储在数据库中的数据相互独立，数据在磁盘等设备上的物理存储位置发生改变也不影响用户的应用程序。实际上数据库中的数据是由 DBMS 统一管理的，编制的应用程序只用较为简单的逻辑结构使用数据，不用考虑数据在存储设备上的物理结构与位置，实现了应用程序与数据的总体逻辑结构、物理存储结构之间的独立。特别有意义的是，应用程序的编写可以简化，应用程序的修改和维护成本很大程度上降低了。

（5）统一管理和控制数据。由于数据库要被多个应用程序所共享，多个用户可以同时使用一个数据库，其中的数据有可能被不同的用户修改、存取，专业说法是并发使用，会造成不一致，因此 DBMS 必须强化对数据的统一管理和控制，通常会提供数据的安全性保护、数据的完整性控制、并发操作控制及数据库恢复等功能。

（6）具有良好的用户接口。使得用户应用程序可以方便地使用数据库。

1.3　数据模型

数据模型是指可以反映世界上各种事物与事物之间关系的数据形式和相关的组织结构，是用于抽象地表示和处理现实世界中的数据和信息的工具。数据库通常是某个组织或行业所涉及的数据集合，它不但要反映这些数据本身的含义，而且要反映数据之间的关系。但是这些现实世界的数据不能直接放到目前这种计算机系统的数据库中去，而数据模型就起到了从现实世界表示到计算机表示的一个中间层作用，一种好的数据模型应该具有能够比较真实地模拟现实世界、容易被人理解和便于在计算机中实现的功能。有多种数据模型，而任何一种数据库管理系统都是基于某种数据模型的。

1.3.1　客观对象到模型的转换

把现实世界中存在的客观对象（事物）转换为模型表示，一般做法是先把客观对象通过概念系统逐步抽象，然后再组合为 DBMS 可以支持的数据模型。通俗地说是首先把现实世界的对象抽象为某一种不依赖于具体计算机系统的数据结构（称为概念模型），然后再把概念模型转换为计算机中某种 DBMS 所支持的数据模型，如图 1.5 所示。

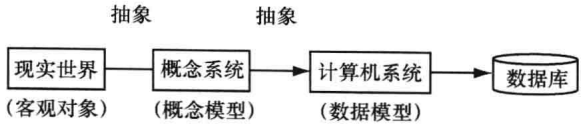


图 1.5　客观对象抽象到数据模型

概念模型可以按照用户的想法准确地模拟某组织或行业单位对数据的描述及业务要求，即对应用的数据建模，最常见的是用“实体——关系”（Entity-Relationship），简称 E-R 方法建立概念模型，再过渡为计算机系统所能支持的数据模型来组织数据，此时数据模型要包括数据的静、动态两方面的内容。

（1）数据的静态特性。应包括数据的基本逻辑结构，数据间的关系和数据完整性约束。

（2）数据的动态特性。指对数据定义的操作，包括操作的规则及实现操作的语言等。

1.3.2　概念模型及其建立

1. 概念模型

概念模型是客观世界事物向抽象世界转换的第一次抽象，也是用户和数据库设计人员之间对于用户单位数据的理解与表示进行交流和沟通的工具，它可以解决现实世界问题如何转换为概念世界问题，最终转换为计算机可以处理的数据世界问题。概念模型的特点是能够方便直观地表达应用中数据的各种语义，比如所描述数据对象的意义和相互关系等，即具有语义表达能力。

在使用概念模型时常用到以下部件或概念。

（1）实体（Entity）。凡是可以被识别而又可以互相区别的客观事物统称为实体。实体可以是实际的事物，也可以是抽象的事物，比如教学楼、教师、学生是具体物理上存在的，而信仰、爱好是抽象的。在某一环境中具有共性的一类实体通常组合为一个实体集。比如王春、李刚等人是某学校的学生，他们每一个人都是一个实体，可以把这些学生定义为“学生”实体集，那么此时每个学生都是此实体集中的成员。

（2）实体的属性（Attribute）。一般实体具有若干特征，可以表现其性质，这种特征称为实体的属性。比如教师实体具有“教师编号”、“姓名”、“专业”等属性。

（3）实体集和实体型（Entity Set and Entity Type）。属性的集合表示一种实体的类型，称为实体型，而同类型的实体的集合称为实体集。比如，教师（教师编号、姓名、性别、出生年月、教龄、职称）是一个实体型，全体教师就形成了一个实体集，（9409001、张立、男、1959.1、30、教授）就是教师实体集中的一个实体，即代表教师名单中的一个具体的教师张立。

（4）实体主键（Entity Primary Key）。实体集中的实体键指能够唯一标识实体属性或属性组的数据项。如果一个实体集中存在多个实体键，可以视情况从中选择一个作为实体主键来进行唯一性标识。比如教师实体集有属性教师、编号、姓名、性别、出生年月等，则可选取教师编号这个数据项作为实体主键，可用它来唯一地标识某个特定的教师。

（5）关系（Relationship）。概念模型中的实体集之间可以形成各种关系，这种关系是从客观的现实世界抽象过来的，以反映客观事物之间的关联，这种关系通常分为两种。

① 实体集内部的关系，主要表示实体集内部不同属性之间的关系，比如在课程表实体集中具有的属性（课程编号、课程名、学分、教室、上课时间）中，当确定课程编号时，则与其对应的课程名、学分、教室等属性的值也被唯一地确定了。

② 不同实体集之间的关系。针对通常使用的二元关系（两个实体集之间的关系）而言，具有以下 3 种不同语义的关系。

a. 一对一关系。一对一关系常标识为 1：1。如果对于实体集 X 中的每一个实体，另一个实体集 Y 中最多有一个实体与之相关系，反之 Y 对 X 也是如此，则称实体集 X 与实体集 Y 具有 1：1 关系，比如中学里的班级具有固定的教室上课，那么班级实体集与教室实体集就存在 1：1 关系，因为按照规定，一个教室只有一个班级使用，而一个班级也只能使用一个教室。根据语义，如果有多余的教室未被使用，也没有破坏班级与教室两个实体集之间的 1：1 关系，如图 1.6 所示。

b. 一对多关系。一对多关系常标识为 1：n。如果对于实体集 X 中的每一个实体，实体集 Y 中有 n 个实体（$n \geqslant 0$）与之相关系，而对于实体集 Y 中的每一个实体，实体集 X 中最多只有一个实体与之相关系，则称实体集 X 与实体集 Y 存在 1：n 的关系。比如某学校的班级实体集与学生实体集就是 1：n 的关系，即一个班级可以包括多名学生，而一个学生只属于一个班级，如图 1.7 所示。

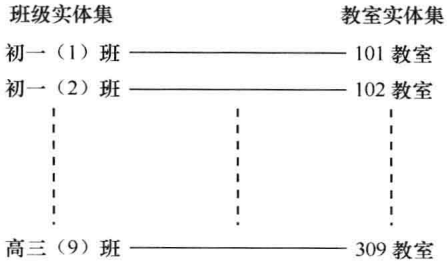


图 1.6　一对一关系示例

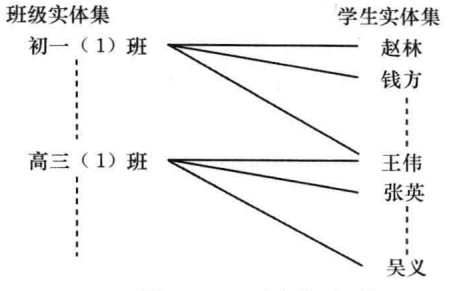


图 1.7　一对多关系示例

c. 多对多关系。多对多关系常标识为 m：n。如果对于实体集 X 中的每一个实体，实体集 Y 中有 n 个实体（$n \geqslant 0$）与之相关系，而对于实体集 Y 中的每一个实体，实体集 X 中有 m 个实体（$m \geqslant 0$）与之相联系，则称 X 实体集与 Y 实体集之间存在 m：n 关系。比如一个学生可以选修多门课程，而一门课程也可以被多名学生选修，则学生实体集与课程实体集之间就存在 m：n 关系，如图 1.8 所示。

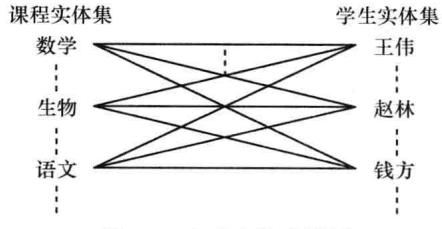


图 1.8　多对多关系示例

上述关系中一对多是最普遍常用的关系，一对一关系 1：1 可以看成是一对多关系 1：n 的特殊情况。

2. 实体关系模型

实体关系模型又叫 E-R 图或 E-R 模型，它是一种目前最常用的概念模型，是一种描述概念世界、建立概念模型的实用工具。当一个单位（比如企事业单位、学校）要建立数据库时，常用 E-R 模型对此单位的信息结构进行模拟，得到一个单位的 E-R 概念模式，这种概念模式可以用直观的 E-R 图体现。

E-R 图包括以下要素或构成部件，如图 1.9 所示。

（1）实体。用矩形表示一个实体，矩形内标有实体名。

（2）属性。用椭圆形表示某个属性，并用直线与相关实体连接，通常一个实体矩形有若干与之连接的椭圆形属性。

（3）实体之间的关系。用菱形表示，即在相关实体之间用菱形表示中介，将不同的实体关联起来。

（4）关系类型。将 1∶1、1∶n、及 m∶n 3 种类型的关系标注在菱形与矩形连接的直线上。

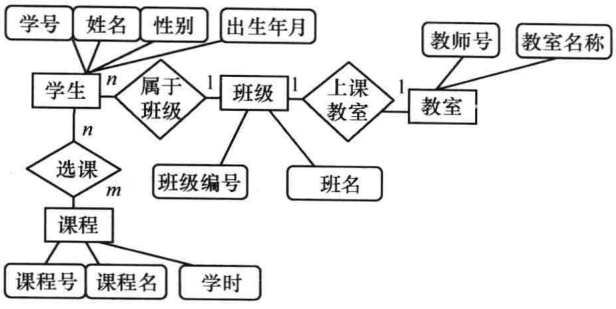


图 1.9　E-R 图例

3. 数据模型

由于现代计算机系统无法直接表示客观世界的事物，为了表示客观事物本身及事物之间的各种关系，数据库中的数据必须具有计算机系统能够识别并处理的结构。数据模型就是直接面向计算机系统（针对数据库）中数据的逻辑结构。数据库不仅要管理数据本身，而且还要利用数据模型表示出数据之间的关系，数据模型是支持 DBMS 用于表示实体及实体间关系的方法，它应该正确地表达所涉及数据间存在的整体逻辑关系。

某个特定的 DBMS 都是基于某种数据模型的。根据实体集之间的不同结构安排，常见的数据模型有层次模型、网状模型、关系模型和面向对象模型 4 种。由于关系数据模型从理论到实践更适用于现代计算机系统和人们的习惯思维，所以它成为目前最为流行的数据库所采用的数据模型。

（1）层次数据模型（Hierarchical Data Model）。这是设计数据库时最早出现的数据模型。它用树形的层次结构表示各类实体以及实体之间的关系，类似一棵倒立的树，从最上面的根结点层次开始定义，逐步往其下层称为子孙的结点进行定义，如图 1.10 所示。

层次模型将一对多的层次关系描述得非常直观，易于理解，这是其突出优点。但是它也存在较大缺点。在这种结构中查找数据时，每次需要从最上层的根结点开始，一个个地沿路径上的结点逐层查找，使用起来也不够方便。当需要应用于不同情形时，容易造成数据重复，给数据维护造成很大的不便。

（2）网状数据模型（Network Data Model）。网状数据模型用网状结构表示实体与实体之间的关系。在现实世界中事物之间的关系多数是非层次结构的，如果用前述的层次模型很难描述，而用网状模型表示起来比较方便。从理论上说，网状模型中的任意结点间都可以用连线建立关系，可以支持多对多的关系。它的明显缺点是，应用程序在访问数据库时必须选择适当的效率高的存取路径，即访问路径必须事先设定。这就要求用户编程序之前了解系统内实体集之间结构的细节，加重了编写应用程序的负担，特别是需要重新建立关系或建立新连接时很麻烦，如图 1.11 所示。

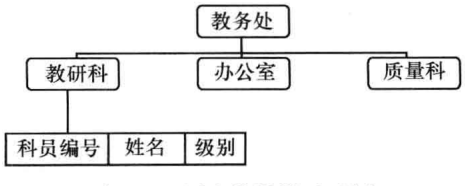


图 1.10　层次数据模型示例

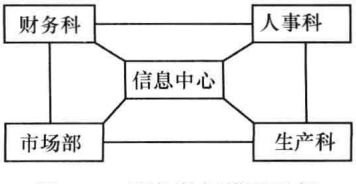


图 1.11　网状数据模型示例

（3）关系数据模型（Relational Data Model）。关系数据模型是用二维表结构表示实体集和实体集之间关系的一种当今使用最为普遍的数据模型，也是本书重点介绍的 Access 2003 数据库所采用的数据模型。

用关系数据模型建立的数据库是以关系数学理论为基础的，用 m 行 n 列的二维表格来描述数据集合及其相互关系。在关系模型中操作的对象和结果都是统一格式的二维表，这种二维表称为关系，见表 1.1。

表 1.1　　关系数据模型示例

教师编号	姓　　名	系　　别	所上课程
19401001	赵刚	计算机	电子商务
19401202	王方	英语	英语
19401098	周小丽	国际贸易	国际结算
19401103	李万文	国际贸易	物流管理
19401120	吴建	国际贸易	计量经济学

在关系模型的二维表中，每一行数据称为一个记录，通常表示一条具体的数据信息。比如上述表中第一条记录表示了教师编号为 19401001 的赵刚老师是计算机系的，给相关学生上电子商务课程；表中的每一列数据称为一个字段，可以表示一类属性，比如上述表中教师编号列，表示本列数据是所有教师的编号。

关系数据模型具有一系列优点。

① 它建立在严密的数学基础之上。

② 无论是实体还是实体集之间的联系都用关系来表示，概念单一，易于理解，数据结构简单，方便用户使用。

③ 所有数据运算（各种操作）的结果也用关系表示。

④ 它的存取路径对用户透明，数据独立性强，安全性好，可以简化数据库开发和应用程序员的工作。

（4）面向对象模型（Object-Orinted Model）。面向对象的数据模型作为一种可扩充的数据模型，于 20 世纪 80 年代被提出并开始进行研究。在面向对象模型中，现实世界的实体被看成一种对象，类同的对象归为一类（class）。面向对象模型具有语义表达能力强、可支持复杂的数据模型、可封装性、继承性、可支持长事务处理等优点。面向对象的方法和技术在计算机各个领域包括软件工程、信息系统设计、程序设计语言等方面均有很大影响，并有广泛的研究。

1.4　关系数据库

本节主要讲述人们利用集合论中的数学理论、方法与计算机系统中的数据库应用相结合，开发出了基于关系数据模型的关系数据库系统，介绍了关系数据库的基本概念、结构、方法及操作等。

1.4.1　数据结构

1. 逻辑结构

关系数据模型中的结构是基于关系的，而这种关系所表示的逻辑结构具有二维表的结构形

式。这与我们日常生活中使用的一种二维表是一样的，由表名、行和列组成。这种简单的表格可以直观地描述数据及其关系，这里表格称为关系，或者说关系数据库中的关系是用表格来表示的见表 1.2。

表 1.2　　“学生”表

学　号	姓　名	性　别	出生年月	籍　贯	专　业
2012012001	刘伟	男	1993-3	江苏	中文
2012012002	王大刚	男	1992-2	河北	中文
2012023012	周金玉	女	1993-5	江苏	计算机
2012032011	马雯红	女	1993-7	北京	金融

为了表示和处理，表格规定了以下一些术语。

（1）关系（Relation）。一个关系对应一张表。它要有一个关系名，抽象表示为：关系名（属性名 1，属性名 2…，属性名 n）。比如上述“学生”表具体描述为：学生（学号，姓名，性别，出生年月，籍贯，专业）。规定了一个表中每一列应该填入的数据内容，又称为关系数据模式。

（2）元组（Tuple）。在一个填入内容的具体的二维表中，水平方向的每一行称为一个元组，元组对应表中的一条具体的数据记录。比如表 1.2 中有学生刘伟等 4 条表示各人情况的具体数据。

（3）属性（Attribute）。在一个填入内容的二维表中，垂直方向的每一列称为一个属性，并且具有各自的属性名。在具体的表中属性名又叫字段名，每个字段的数据类型宽度等在一开始创建这张表时要做出规定。比如表 1.2 中的表具有“学生”、“姓名”、“性别”、“出生年月”、“籍贯”、“专业”6 个字段名。

（4）域（Domain）。属性的取值范围称为域，是指不同的元组（表中不同的行）对同一个属性（表中某一个列）的取值所限定的范围，比如表 1.2 中的“学号”列取值范围是 10 位长的数字，“性别”列的取值范围是“男”、“女”两个汉字之一。

（5）主键（Primary Key）。主键（又称为关键字）的值能够唯一地标识一个元组的属性或属性的组合，即指字段或字段的组合。比如表 1.2“学生”表中的“学号”字段可以唯一确定一个元组（一条记录），所以可以作为“学生”表的主键，而“姓名”字段可能会有同名同姓的人重复出现在不同的元组（记录）中，所以“姓名”字段不能单独作为“学生”的主键。一个表只能规定一个主键，而主键通常是一个字段，但也可以由多个字段组合构成。

（6）外键（Foreign Key）。外键又称为外关键字，它是指表中的某一个字段不是本表的主键，而是另外一个表的主键，则这个字段（属性）就称为外键，外键在表中相同的值可能会出现在本表中不同的元组里。

2. 存储结构

这里存储结构是指关系数据库最终是以何种方式放在现行体系结构的计算机中的。按逻辑结构设计好的二维表是以文件形式存储在计算机系统的物理组织中的。

前文所述关系、模型、模式、元组、属性等术语，是属于数学理论体系的描述，而关系数据库从设计到最终应用还要涉及用户和程序设计人员，他们要把抽象的理论表示理解、转换到自己的实际使用上去。把各个角色直接面对的术语进行对照的结果见表 1.3。

表 1.3　　基本术语的对照

关系模型	程序员	用　户
关系模式	文件结构	二维表结构
关系（二维表）	文件	表
元组	记录	行
属性	数据项（字段）	列

3. 关系的特点

在实际使用关系数据模型方法去实现关系数据库时要有所限制，当用表来对应表示关系时要符合一些规定。

（1）关系的规范化。关系中的每一个属性都应该是原子数据（atomic data）项，是不能再分的数据项。比如整数、字符串、汉语、文字等，不包括组合数据，比如集合、数组、记录等，在我们日常生活中出现的复合表，见表 1.4，不属于在此定义的二维表，不能直接作为一个关系对应的表来存储，即表中某列下属不能再有子列。

表 1.4　　复合表示例

车间编号	车间名	生产量		
		第一班	第二班	第三班

（2）每个属性都要有其取值范围，即有值域。

（3）同一个关系中不能出现相同的属性名，也就是说同一张表中不能有相同的字段名。

（4）关系中不能有相同的元组，即同一张表中不同行的内容不能相同。

（5）关系中元组的次序可以任意交换，即同一张表中的行可以放在不同的位置上。

（6）从理论上讲关系中属性的顺序无关紧要，可以任意交换。但在实际应用时，由于关系数据库要求在建立表（关系）时对其属性要顺序逐个定义，所以实际使用时最好不要随意交换一个表的字段次序。

1.4.2　实际应用中的关系模型

在实际构造关系数据库时，具体的关系模型往往有多个关系模式组成。比如在 Access 2003 关系数据库中，一个数据库会包含相互之间存在关系的多个表。这些表的关系模式一般不同，即它们有各自的二维表结构。这样存储在计算机系统中的数据库文件就对应一个实际的关系模型。在实际构造表时，要考虑到不同表中的实体之间如何建立关系的问题，一般采用公共字段名作为中介，并且利用主键、外键的概念。而这种中介关系应当从语义出发来考虑如何建立。

【例 1.1】 在学生管理数据库中，建立学生及选课成绩关系模型，并利用公共字段名建立关系，以得到学生成绩表。

解：在学校学生管理数据库中已建有学生表和学生选课成绩表，见表 1.2 和表 1.5。

表 1.2 和表 1.5 的关系数据模型（表的结构）分别为：

学生（学号，姓名，性别，出生年月，籍贯，专业）；

学生选课成绩（学号，课程表，成绩）。

可以建立如表 1.6 和表 1.7 所示的学生及选课成绩关系模型。