



河南省“十四五”普通高等教育规划教材

高等院校计算机应用系列教材

# Access 2021数据库应用基础教程 (微课版)

卫琳 主编

赵哲 高军亮 石育澄 副主编

清华大学出版社  
北京

## 内 容 简 介

本书全面讲述了Access数据库管理系统的基本原理和技术。本书共分为10章，内容涵盖Access数据库的基本概念、数据库对象及视图、Access界面环境的使用与定制、数据库的整体设计流程、创建数据库和表、设计表结构、设置表的主键和索引、创建表之间的关系、在数据表视图中操作数据、使用查询操作数据、使用窗体显示和编辑数据、使用报表呈现与打印数据、使用表达式和SQL语句、使用宏让操作自动化、数据库的安全管理与维护等。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实。本书主要面向数据库初学者，适合作为高等院校数据库教材，还可作为Access应用开发人员的参考书。

本书配套的电子课件和习题答案可以到<http://www.tupwk.com.cn/downpage>网站下载，也可以扫描前言中的二维码获取。扫描前言中的视频二维码可以直接观看教学视频。

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。举报：010-62782989，[beiqinquan@tup.tsinghua.edu.cn](mailto:beiqinquan@tup.tsinghua.edu.cn)。

### 图书在版编目(CIP)数据

Access 2021数据库应用基础教程：微课版 / 卫琳主编.

北京：清华大学出版社，2026. 1. -- (高等院校计算机应用系列教材).

ISBN 978-7-302-70418-8

I. TP311.132.3

中国国家版本馆CIP数据核字第2025RP7241号

责任编辑：胡辰浩

封面设计：高娟妮

版式设计：妙思品位

责任校对：成凤进

责任印制：曹婉颖

出版发行：清华大学出版社

网 址：<https://www.tup.com.cn>，<https://www.wqxuetang.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-83470000 邮 购：010-62786544

投稿与读者服务：010-62776969，[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者：河北龙大印务有限公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：22.5 字 数：534千字

版 次：2026年1月第1版 印 次：2026年1月第1次印刷

定 价：79.80元

---

产品编号：102705-01

随着人类步入信息社会，数据库作为信息系统的核心与基础设施，已经渗透到社会生活的方方面面，人们时时刻刻都在从数据库中汲取所需，利用数据库中的数据沟通世界。本书面向具有一定计算机基础知识和操作技能的各个层次的读者，通过本书10章内容的介绍，让读者全面了解数据库基本知识，学会运用Access设计开发简单的数据库应用实例，帮助读者养成计算思维能力，运用抽象与自动化思想解决实际问题，并能够为解决日常数据管理问题提供参考。

本书作为数据库技术的入门教材，期望带领读者进入数据库之门，掌握数据库管理数据的思想、方法和技术，培养以抽象与自动化为核心的计算思维能力。学会数据建模，熟练使用简单好用的Access数据库管理系统，创建和管理关系数据库，设计开发数据库应用系统，解决数据管理的现实问题。

本书希望帮助读者达到以下学习目标。①认识数据库系统，认识数据管理；认识经过两级抽象建立的数据模型；认识关系数据模型，掌握建模方法。这些知识适用于所有数据库系统，读者在学习其他数据库技术时，可以将这些知识迁移到新的学习场景中，有助于快速进入新场景下的核心内容的学习。②学会使用数据库管理系统创建数据库、数据表和表间关系，实现数据模型的数据结构和完整性约束；掌握使用Access中的查询，实现一般意义的查询和批量数据维护，认识作为关系数据库标准语言的SQL，学会SQL基本查询语句；设计开发数据库应用系统，使用Access创建图形化用户界面，实现数据的方便输出，实现操作自动化。在宏的使用中体现了很多的程序设计思想和方法。读者可以方便地将以上技能和经验运用到实际应用场景下。本书使用Access数据库管理系统创建数据库及其应用系统，为读者今后设计、创建、管理、开发大中型数据库系统打下坚实的基础。

与市面上大多数的Access教程不同，本书没有呆板地介绍Access功能的基本使用方法和操作步骤，而是在讲解过程中包含更多的概念性和原理性的介绍，并提供了很多经验技巧，列举了操作中需要注意的问题。只有这样才能让读者真正掌握Access，而不只是按部就班、机械性的傻瓜式操作。书中的每一章都是一个独立的主题。本书以数据库系统的开发流程来组织各章内容和排列顺序，有助于梳理Access知识体系和数据库开发流程，建议读者按照书中的章节顺序进行阅读。本书在很多章的内容讲解上都是先从整体上介绍功能特性、相关概念和基本步骤，然后从细节上对特定的主题进行详细讲解，这种总分式的结构便于读者从整体到局部快速理解和掌握Access知识和技术。本书对设计过程中的细节操作介绍得非常详细，例如在设计窗体和报表时，对单态控件的位置、大小、对齐方式的调整，以及多态控件的排版和布局等都进行了详细讲解，这些内容是决定数据库设计质量和操作效率的关键因素。本书以Access 2021作为主

要操作环境，内容适用于Access 2019以及Access 2016之前的Access版本。本书共10章，列举了72个案例，其中包括71个小案例和1个大型综合案例。

本书作为省级十四五规划教材，主要面向数据库初学者，适合作为高等院校的数据库教材及各种数据库应用程序开发人员的参考书。

本书的作者团队是从事数据库技术教学工作多年的资深教师，有丰富的教学和项目开发经验，本书的编写工作主要由卫琳、赵哲、高军亮和石育澄完成，其中赵哲负责第1~3章，石育澄负责第4~6章，卫琳负责第7~10章，高军亮负责视频的组织架构及录制，除封面署名的作者外，参与本书编写的人员还有石磊、刘振昊、李昊帆、孙俊豪等。

由于作者水平有限，书中难免有不足之处，恳请专家和广大读者批评指正。在编写本书的过程中参考了相关文献，在此向这些文献的作者深表感谢。我们的电话是010-62796045，邮箱是992116@qq.com。

本书配套的电子课件和习题答案可以到<http://www.tupwk.com.cn/downpage>网站下载，也可以扫描下方二维码获取。扫描下方二维码可以直接观看教学视频。



配套资源



看视频

作者  
2025年8月

<b>第 1 章</b>	<b>数据库基础及Access概述</b> ··· 1	
1.1	数据库的基本概念····· 2	
1.1.1	数据与数据处理····· 2	
1.1.2	数据库····· 3	
1.1.3	数据库技术的发展历程····· 4	
1.1.4	数据库系统····· 5	
1.1.5	数据库管理系统(DBMS)····· 8	
1.1.6	数据库应用系统(DBAS)····· 9	
1.2	数据库系统的体系结构····· 9	
1.2.1	集中式体系结构····· 10	
1.2.2	客户—服务器结构····· 11	
1.2.3	并行系统结构····· 11	
1.2.4	分布式系统结构····· 12	
1.3	数据模型····· 12	
1.3.1	概念模型····· 13	
1.3.2	用E-R方法表示概念模型····· 14	
1.3.3	逻辑数据模型····· 15	
1.4	关系数据库····· 17	
1.4.1	关系模型中的基本术语····· 17	
1.4.2	关系数据库中表之间的关系····· 19	
1.4.3	关系模型的完整性约束····· 19	
1.5	关系代数····· 21	
1.5.1	传统的集合运算····· 21	
1.5.2	专门的关系运算····· 23	
1.6	规范化理论····· 25	
1.6.1	非规范化的关系····· 25	
1.6.2	第一范式(1NF)····· 25	
1.6.3	第二范式(2NF)····· 26	
1.6.4	第三范式(3NF)····· 27	
1.6.5	其他范式····· 27	
1.7	数据库语言····· 29	
1.7.1	数据定义语言(DDL)····· 30	
1.7.2	数据操纵语言(DML)····· 30	
1.8	数据库设计····· 30	
1.8.1	数据库设计的目标····· 31	
1.8.2	数据库设计的特点····· 31	
1.8.3	数据库设计的方法····· 31	
1.8.4	数据库设计的步骤····· 32	
1.9	Access概述····· 35	
1.9.1	Access简介····· 35	
1.9.2	Access数据库····· 36	
1.9.3	Access数据库对象····· 37	
1.10	本章小结····· 39	
1.11	思考和练习····· 39	
<b>第 2 章</b>	<b>Access表的建立和操作</b> ··· 41	
2.1	Access 2021环境简介····· 41	
2.1.1	启动Access 2021····· 41	
2.1.2	打开和新建数据库····· 42	
2.1.3	标题栏····· 44	
2.1.4	功能区····· 45	
2.1.5	导航窗格····· 45	
2.1.6	状态栏····· 47	
2.1.7	关闭Access 2021····· 47	
2.2	Access 2021的功能区····· 48	
2.2.1	显示或隐藏功能区····· 48	
2.2.2	常规命令选项卡····· 48	
2.2.3	上下文命令选项卡····· 52	
2.2.4	自定义功能区····· 52	
2.3	操作数据库和数据库对象····· 54	
2.3.1	使用模板创建数据库····· 54	
2.3.2	转换数据库····· 56	

2.3.3	打开数据库	57	3.2.1	查找	103
2.3.4	操作数据库对象	58	3.2.2	替换	105
2.3.5	保存数据库	60	<b>3.3</b>	<b>复制与删除</b>	105
2.3.6	关闭数据库	60	3.3.1	复制记录	105
<b>2.4</b>	<b>表的相关知识</b>	61	3.3.2	删除记录	106
2.4.1	数据表相关概念	61	<b>3.4</b>	<b>改变记录的显示方式</b>	106
2.4.2	表之间的关系	61	3.4.1	改变字段的显示顺序	107
2.4.3	表的结构	62	3.4.2	隐藏/取消隐藏字段	107
2.4.4	数据类型	63	3.4.3	调整行高与列宽	108
2.4.5	字段属性	66	3.4.4	冻结/解冻列	109
<b>2.5</b>	<b>创建Access 2021数据表</b>	72	3.4.5	设置数据表格式	110
2.5.1	使用数据表视图创建表	73	<b>3.5</b>	<b>排序和筛选记录</b>	111
2.5.2	使用设计视图创建表	75	3.5.1	排序规则	112
2.5.3	使用模板创建表	77	3.5.2	数据排序	112
2.5.4	通过导入或链接创建表	78	3.5.3	数据筛选	115
2.5.5	创建查阅字段列	82	<b>3.6</b>	<b>对数据表中的行汇总统计</b>	120
<b>2.6</b>	<b>表的相关操作</b>	86	3.6.1	添加汇总行	120
2.6.1	打开表	86	3.6.2	隐藏汇总行	121
2.6.2	复制表	86	<b>3.7</b>	<b>导出数据表</b>	121
2.6.3	重命名表	87	3.7.1	导出到文本文件	121
2.6.4	删除表	87	3.7.2	导出到Excel工作表	123
2.6.5	修改表结构	87	3.7.3	导出到PDF或XPS	124
2.6.6	设置表的主键	91	<b>3.8</b>	<b>本章小结</b>	125
2.6.7	创建索引	92	<b>3.9</b>	<b>思考和练习</b>	125
2.6.8	关闭表	94	<b>第4章</b>	<b>数据查询</b>	126
<b>2.7</b>	<b>表之间的关系</b>	94	4.1	查询概述	126
2.7.1	建立表间的关系	95	4.1.1	查询与表的区别	127
2.7.2	关系选项	97	4.1.2	查询的功能	127
2.7.3	编辑关系	97	4.1.3	查询的类型	128
2.7.4	删除关系	98	<b>4.2</b>	<b>SQL语言简介</b>	129
<b>2.8</b>	<b>本章小结</b>	98	4.2.1	SQL概述	129
<b>2.9</b>	<b>思考和练习</b>	98	4.2.2	使用SELECT语句	129
<b>第3章</b>	<b>数据的编辑</b>	100	4.2.3	高级查询语句	133
<b>3.1</b>	<b>表记录的输入和操作</b>	100	<b>4.3</b>	<b>使用向导创建查询</b>	138
3.1.1	表记录的添加	100	4.3.1	简单查询	138
3.1.2	输入数据	101	4.3.2	交叉表查询	140
3.1.3	查看记录	102	4.3.3	查找重复项查询	142
3.1.4	修改记录	103	4.3.4	查找不匹配项查询	143
<b>3.2</b>	<b>查找和替换</b>	103	<b>4.4</b>	<b>使用查询设计视图创建查询</b>	144

4.4.1	查询设计视图	144
4.4.2	创建多表查询	146
4.4.3	编辑查询	147
4.4.4	设置查询条件	151
4.4.5	参数查询	154
4.4.6	聚合查询	156
4.4.7	嵌套查询	158
4.5	其他查询	159
4.5.1	操作查询	159
4.5.2	SQL查询	163
4.6	本章小结	166
4.7	思考和练习	166
<b>第 5 章 模块与VBA</b> 167		
5.1	模块的基本操作	167
5.1.1	什么是VBA	167
5.1.2	模块的分类	169
5.1.3	创建和运行模块	170
5.2	VBA程序设计基础	171
5.2.1	VBA编程环境	172
5.2.2	数据类型	176
5.2.3	常量、变量和数组	177
5.2.4	运算符和表达式	182
5.2.5	VBA常用语句	185
5.2.6	面向对象程序设计概述	186
5.3	VBA的流程控制结构	188
5.3.1	顺序语句	189
5.3.2	分支结构	189
5.3.3	循环结构	193
5.3.4	跳转语句	195
5.4	VBA高级程序设计	196
5.4.1	过程与函数	196
5.4.2	过程调用	198
5.4.3	常用函数	200
5.4.4	程序调试	201
5.4.5	VBA程序中的错误处理	204
5.4.6	VBA代码的保护	205
5.5	本章小结	206
5.6	思考和练习	206
<b>第 6 章 窗体</b> 208		
6.1	窗体概述	208
6.1.1	窗体的功能	208
6.1.2	窗体的类型	209
6.2	创建窗体	210
6.2.1	快速创建窗体	211
6.2.2	窗体的视图	213
6.2.3	使用窗体向导创建窗体	213
6.2.4	创建“空白窗体”	215
6.3	设计窗体	216
6.3.1	窗体的设计视图	216
6.3.2	窗体设计工具的功能区选项卡	217
6.3.3	“属性表”窗格	219
6.3.4	使用控件	223
6.3.5	编辑控件	227
6.3.6	创建弹出式窗体	230
6.4	创建主/子窗体	235
6.4.1	利用向导创建主/子窗体	235
6.4.2	快速创建主/子窗体	239
6.5	本章小结	239
6.6	思考和练习	239
<b>第 7 章 报表</b> 240		
7.1	报表概述	240
7.1.1	报表与窗体的区别	240
7.1.2	报表的视图	241
7.1.3	报表的结构	242
7.1.4	报表的分类	242
7.2	创建报表	243
7.2.1	一键生成报表	243
7.2.2	使用向导创建报表	244
7.2.3	使用空报表创建报表	245
7.2.4	使用设计视图创建报表	248
7.2.5	创建标签报表	250
7.2.6	创建子报表	251
7.2.7	将窗体另存为报表	253
7.3	编辑与打印报表	254
7.3.1	设置报表的外观	254
7.3.2	编辑报表的页眉/页脚	256

7.3.3	应用主题	257	9.2.1	压缩和修复数据库	297
7.3.4	避免出现空白报表	258	9.2.2	设置权限	299
7.3.5	报表的预览与打印	258	9.2.3	设置密码	300
7.4	报表数据中的分组和汇总	261	9.2.4	使用密码	301
7.4.1	报表数据中的分组	261	9.2.5	撤销密码	302
7.4.2	在组页眉中创建表达式	263	9.3	备份数据库	302
7.4.3	对报表中的分组数据进行排序	264	9.4	数据的导入与导出	306
7.4.4	汇总报表数据	264	9.4.1	数据的导入	307
7.5	本章小结	266	9.4.2	数据的导出	313
7.6	思考和练习	266	9.5	Access中的安全机制	316
<b>第8章</b>	<b>宏</b>	<b>267</b>	9.5.1	用户级安全机制	316
8.1	宏简介	267	9.5.2	工作组信息文件	316
8.1.1	宏与事件	267	9.5.3	用户与组账户	316
8.1.2	宏的类型	270	9.6	本章小结	317
8.1.3	宏的设计视图	271	9.7	思考和练习	317
8.2	创建和使用宏	271	<b>第10章</b>	<b>数据库应用系统开发实例</b>	<b>318</b>
8.2.1	创建简单宏	272	10.1	需求分析与系统设计	318
8.2.2	创建子宏	275	10.1.1	项目背景	318
8.2.3	创建条件宏	278	10.1.2	需求分析	319
8.2.4	编辑宏	281	10.1.3	功能设计	319
8.2.5	运行宏	282	10.1.4	数据库设计	320
8.2.6	错误处理与宏调试	283	10.2	数据库实施	323
8.3	数据宏	286	10.2.1	创建空白数据库	323
8.3.1	表事件	287	10.2.2	创建数据表	323
8.3.2	数据宏的操作目录	287	10.2.3	创建查询	325
8.3.3	创建数据宏	290	10.2.4	创建窗体	328
8.4	宏的安全设置	293	10.2.5	创建报表	335
8.4.1	解除阻止的内容	293	10.2.6	添加VBA代码	339
8.4.2	设置信任中心	294	10.2.7	创建AutoExec宏	346
8.5	本章小结	294	10.3	系统功能实现	346
8.6	思考和练习	295	10.4	本章小结	349
<b>第9章</b>	<b>数据库的安全管理</b>	<b>296</b>	10.5	思考和练习	349
9.1	数据库安全定义	296	<b>参考文献</b>	<b>350</b>	
9.2	数据库的压缩与修复	297			

## 第 1 章

# 数据库基础及 Access 概述

计算机技术和互联网技术的飞速发展开启了一个全新的数字生活时代，以大学生的校园生活为例：使用校园卡就餐可以即时看到消费的金额及卡内余额，如需充值，使用手机银行App即可轻松转账；去图书馆查阅资料，校园卡又变成了借书卡，刷卡进入、借书、还书一卡通。学生在教学管理平台进行选/退课、查询课表、查询考试信息和考试成绩等，大大提高了学生的办事效率，也丰富了学生的学习资源。如此庞大的在线数据是如何实现数据准确、更新及时、安全存储以及各个系统之间数据一致的呢？这就是数据管理技术的重要作用。

数据库作为数据管理技术，是计算机科学的重要分支。在信息技术飞速发展的今天，信息与数据已经成为各行各业的重要财富和资源，数据库应用无处不在。因此，掌握数据库的基本知识和使用方法不仅是计算机相关专业的需要，而且是目前很多工科专业学生应该具备的技能。本章主要介绍数据库系统的基本概念、数据库系统的体系结构、数据模型、关系数据库、关系代数、规范化理论、数据库语言、数据库设计的方法与步骤、Access数据库简介等。

本章的学习目标：

- 掌握与数据库相关的基本概念
- 理解数据库系统的体系结构
- 掌握常见的数据模型
- 理解关系数据库的基本理论
- 了解关系代数的基本运算
- 掌握关系数据库的规范化理论
- 了解数据库语言
- 掌握数据库设计的方法与步骤
- 了解Access数据库的简单概况

## 1.1 数据库的基本概念

数据库是信息系统的核心与基础，它提供了最基本、最准确、最全面的信息资源，对这些资源的管理和应用，已经成为人们科学决策的依据。数据库应用已遍及人们生活中的各个角落，如铁路及航空公司的售票系统、图书馆的图书借阅系统、学校的教学管理系统、超市售货系统和银行的业务系统等。数据库与人们的生活密不可分，几乎每个人的生活都离不开数据库。对于一个国家而言，数据库的建设规模、数据库信息量的大小和使用频度已成为衡量这个国家信息化发达程度的重要标志之一，而信息化对于加快国家产业结构调整、促进经济增长和提高人们生活质量具有明显的倍增效应和带动作用。

### 1.1.1 数据与数据处理

人们在现实中进行的各种活动，都会产生相应的信息，例如，生产服装的工厂，其用于生产的原材料的名称、库存量、单价、产地；所生产的产品的名称、数量、单价；该工厂中职工的职称、编号、薪水、奖金等，所有这些都是信息。这些信息代表了所属实体的特定属性或状态，当把这些信息以文字记录下来便是数据，因此可以说，数据是信息的载体。本节主要介绍信息、数据和数据处理的概念。

#### 1. 信息与数据

信息与数据是两个密切相关的概念。信息是各种数据所包含的意义，数据则是负载信息的物理符号。例如，某个人的身高，某个学生的考试成绩，某年度的国民收入等，这些都是信息。如果将这些信息用文字或其他符号记录下来，则这些文字或符号就是数据。

同一数据在不同的场合具有完全不同的意义，例如，31这个数，既可以表示一个人的年龄，也可以表示长度，或者表示某个学生某科目的考试成绩等。在许多场合下，对信息和数据的概念并不做严格的区分，可互换使用，例如，通常所说的“信息处理”和“数据处理”，这两个概念的意义是相同的。

信息是对现实世界事物存在方式或运动状态的反映。它已成为人类社会活动的一种重要资源，与能源、物质并称为人类社会活动的三大要素。一般来说，信息是一种被加工成特定形式的数据，这种数据形式对接收者来说是有意义的。它具有如下特征。

- 信息可以被感知，不同的信息源有不同的感知方式。
- 信息的获取和传递不仅需要载体，而且还消耗能量。
- 信息可以通过载体进行存储、压缩、加工、传递、共享、扩散、再生和增值等。

在计算机内部，所有的数据均采用0和1进行编码。在数据库技术中，数据的含义很广泛，除数字外，文字、图形、图像、声音、视频等也视为数据，它们分别表示不同类型的信息。

另外，同一种信息可以用多种不同的数据形式进行表达，而信息的意义不随数据的表现形式的改变而改变。例如，要表示某只股票每天的收盘价格，既可以通过绘制曲线图表示，也可以通过绘制柱状图表示，还可以通过表格数据表示，而无论使用何种方式表示，丝毫不会改变信息的含义。

因此，对数据可以做如此定义，描述事物的符号记录称为数据。在学校的学生档案中，可以记录学生的姓名、性别、出生日期、所在系、电话号码和入学时间等。按这个次序排列组合成如下所示的一条记录：

(赵秀明，女，2003-03-25，医学院，13831706516，2021)

这条记录中的信息就是数据。当然数据可能存在因为记录介质被破坏而丢失的风险，例如，记录在纸上的数据，可能因为纸介质丢失、火灾而造成数据丢失；记录在计算机磁盘上的数据，可能因为病毒、误操作等造成数据丢失。

## 2. 数据与信息的关系

数据与信息有着不可分割的联系。信息是被加工处理过的数据，数据和信息的关系是一种原料和成品之间的关系，如图1-1所示。

数据与信息的关系主要表现在以下4个方面。

- (1) 数据是信息的符号表示，或称载体。
- (2) 信息是数据的内涵，是数据的语义解释。
- (3) 数据是符号化的信息。
- (4) 信息是语义化的数据。

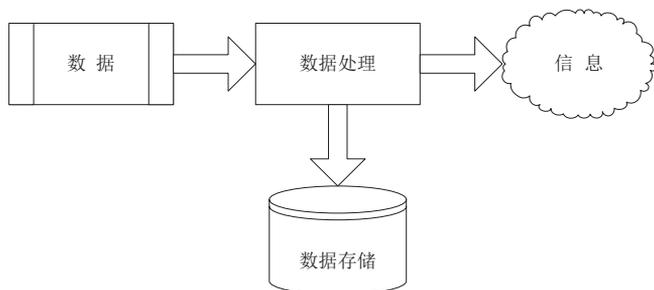


图1-1 数据与信息的关系

## 3. 数据处理

数据处理(data processing)是指对各种形式的数据进行收集、存储、检索、加工、变换和传播的一系列活动的总和。

进行数据处理的目的是有两个：一是从大量的、可能是杂乱无章的、难以理解的数据中抽取、推导出对人们有价值的信息，以作为行动和决策的依据；二是借助计算机科学地保存和管理复杂的、大量的数据，以便人们能够方便而充分地利用这些宝贵的资源。

数据处理是系统工程和自动控制的基本环节。数据处理贯穿于社会生产和社会生活的各个领域。数据处理技术的发展及其应用的广度和深度，极大地影响着人类社会发展的进程。

### 1.1.2 数据库

数据库(database，简称DB)是按照数据结构来组织、存储和管理数据的仓库，例如，日常生活中，我们用笔记本记录亲朋好友的联系方式，将他们的姓名、地址、电话等信息都记录下来。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”。我们可以在这个“数据库”中添加新朋友的个人信息，某个朋友的电话号码变动时也可以修改他的电话号码这个“数据”。使用笔记本这个“数据库”可以方便地查到某位亲朋好友的地址、邮编或电话号码等“数据”。

在信息化社会，充分有效地管理和利用各类信息资源，是进行科学研究和决策管理的

前提条件。数据库技术是管理信息系统、办公自动化系统、决策支持系统等各类信息系统的核心部分，是进行科学研究和决策管理的重要技术手段。

显然，数据库就是存放数据的仓库。它是为了实现一定的目的按某种规则组织起来的“数据”的“集合”。在信息社会中，数据库的应用非常广泛，如银行业用数据库存储客户的信息、账户、贷款以及银行的交易记录；学校用数据库存储学生的个人信息、选课信息、课程成绩等。

在计算机领域，数据库是指长期存储在计算机内的、有组织的、可共享的、统一管理的相关数据的集合。

### 1.1.3 数据库技术的发展历程

从最早的商用计算机起，数据处理就一直推动着计算机的发展。事实上，数据处理自动化早于计算机出现。Hollerith发明的穿孔卡片，早在20世纪初就用来记录美国的人口普查数据，用机械系统来处理这些卡片并列结果。穿孔卡片后来被广泛作为将数据输入计算机的一种手段。

按年代来划分，数据库系统的发展可划分为以下几个阶段。

#### 1. 20世纪50年代至60年代早期

20世纪50年代至60年代早期，磁带被用于数据存储。如工资单这样的数据处理已经自动化了，即把数据存储在上磁带上。数据处理包括从一个或多个磁带上读取数据，并将数据写回到新的磁带上。数据也可以由一叠穿孔卡片输入，然后输出到打印机上。

磁带(和卡片)只能顺序读取，数据量可以比内存大得多，因此，数据处理程序被迫用一种特定的顺序对来自磁带和卡片的数据进行读取和处理。

#### 2. 20世纪60年代末至20世纪70年代

20世纪60年代末，磁盘(硬盘)的广泛使用极大地改变了数据处理的情况，因为通过磁盘可以直接对数据进行访问。磁盘上数据的位置是无意义的，因为磁盘上的任何位置都可在几十毫秒内访问到，数据由此摆脱了顺序的限制。有了磁盘，就可以创建网状数据库和层次数据库，它们具有保存在磁盘上的如链表和树等数据结构。程序员也可以轻松创建和操作这些数据结构。

由Codd写的一篇具有里程碑意义的论文，定义了关系模型和在关系模型中用非过程化的方法查询数据，关系数据库由此诞生。关系模型的简单性和能够对程序员隐藏所有细节的能力具有真正的诱惑力。

#### 3. 20世纪80年代

尽管关系模型在学术上很受重视，但是最初并没有实际的应用，因为它在性能上不足，关系数据库在性能上还不能和当时已有的网状和层次数据库相提并论。这种情况直到System R的出现才得以改变。它是IBM研究院的一个突破性项目开发的一种能够构造高效的关系数据库系统的技术。Astrahan和Chamberlin等人提供了关于System R的很好的综述。功能完善后的System R原型诞生了IBM的第一个关系数据库产品SQL/DS。最初的商用关系

数据库系统，如IBM的DB2、Oracle、Ingres和DEC的Rdb，在推动有效地处理陈述式查询技术上起到了主要作用。到了20世纪80年代早期，关系数据库已经可以在性能上和网状、层次数据库进行竞争了。关系数据库是如此简单易用，最后完全取代了网状和层次数据库。因为程序员在使用后者时，必须处理许多底层的实现问题，并且不得不将要做的查询任务编码成过程化的形式。更重要的是，在设计应用程序时还要时刻考虑效率问题，而这需要付出很大的努力。相反，在关系数据库中，几乎所有的底层工作由数据库自动完成，这使得程序员可以只考虑逻辑层的工作。因为关系模型在20世纪80年代已经取得了优势，所以它在数据模型中具有最高的统治地位。

另外，在20世纪80年代，人们还对并行和分布式数据库进行了很多研究，同时在面向对象数据库方面也有初步的工作。

#### 4. 20世纪90年代初

SQL语言主要是为决策支持应用设计的，重在查询；而20世纪80年代，数据库应用的主流是联机事务处理系统，其工作负载以频繁的更新为主。决策支持和查询再度成为数据库的一个主要的应用领域。分析大量数据的工具有了很大的发展。

在这个时期许多数据库厂商推出了并行数据库产品。数据库厂商还开始在其数据库中加入对象—关系的支持。

#### 5. 20世纪90年代末至今

随着互联网的兴起和发展，数据库比以前有了更加广泛的应用。现在的数据库系统必须支持很高的事务处理速度，而且还要有很高的可靠性和24×7的可用性(一天24小时，一周7天都可用，也就是没有进行维护的停机时间)。数据库系统还必须支持网络接口。

## 1.1.4 数据库系统

数据库系统是计算机化的记录保持系统，它的目的是存储和产生所需要的有用信息。

### 1. 数据库系统的组成

通常，一个数据库系统包括以下4个主要部分：数据、用户、硬件和软件。

#### 1) 数据

数据是数据库系统的工作对象。为了区别输入、输出或中间数据，常把数据库数据称为存储数据、工作数据或操作数据。它们是某特定应用环境中进行管理和决策所必需的信息。特定的应用环境，可以指一个公司、一个银行、一所医院和一个学校等。在这些应用环境中，各种不同的应用可通过访问其数据库获得必要的信息，以辅助进行决策，决策完成后，再将决策结果存储在数据库中。

数据库中的存储数据是“集成的”和“共享的”。“集成”是指把某特定应用环境中的各种应用关联的数据及其数据间的联系全部集中地按照一定的结构形式进行存储，也就是把数据库看作若干个性质的数据文件的联合和统一的数据整体，并且在文件之间局部或全部消除了冗余，这使得数据库系统具有整体数据结构化和数据冗余小的特点；“共享”是指数据库中的一块块数据可为多个不同的用户所共享，即多个不同的用户，使用

多种不同的语言，为了不同的应用目的，而同时存取数据库的信息，甚至同时存取同一数据块。

## 2) 用户

用户是指存储、维护和检索数据库中数据的人员。数据库系统中主要有3类用户：终端用户、应用程序员和数据库管理员(DBA)。

- 终端用户：也称为最终用户，是指从计算机联机终端存取数据库中数据的人员，也可以称为联机用户。这类用户使用数据库系统提供的终端命令语言、表格语言或菜单驱动等交互式对话方式来存取数据库中的数据。终端用户一般是不精通计算机和程序设计的各级管理人员、工程技术人员和各类科研人员。
- 应用程序员：也称为系统开发员，是指负责设计和编制应用程序的人员。这类用户通常使用Access、SQL Server或Oracle等数据库语言来设计和编写应用程序，以对数据库进行存取和维护操作。
- 数据库管理员(DBA)：是指全面负责数据库系统的“管理、维护和正常使用”的人员，可以是一个人或一组人。特别对于大型数据库系统，DBA极为重要，通常设置有DBA办公室，应用程序员是DBA手下的工作人员。DBA不仅要具有较高的技术专长，而且要具备较深的资历，并具有了解和阐明管理要求的能力。DBA的主要职责包括参与数据库设计的全过程；与用户、应用程序员、系统分析员紧密结合，设计数据库的结构和内容；决定数据库的存储和存取策略，使数据的存储空间利用率和存取效率均较优；定义数据的安全性和完整性；监督控制数据库的使用和运行，及时处理运行程序中出现的問題；改进和重新构建数据库系统等。

## 3) 硬件

硬件是指存储数据库和运行数据库管理系统(Database Management System, DBMS)的硬件资源，包括物理存储数据库的磁盘、磁鼓、磁带或其他外存储器及其附属设备、控制器、I/O通道、内存、CPU以及外部设备等。数据库服务器的处理能力、存储能力、可靠性直接关系到整个系统的性能优劣，因此对服务器端硬件资源也有着较高的要求，应选用具有高可靠性、高可用性、高性价比的服务器。通常要求考虑以下问题。

- 具有足够大的内存，用于存放操作系统、DBMS的核心模块、应用程序，以及数据缓冲区中的内容。
- 具有高速大容量的直接存取设备。一般数据库系统的数据量和数据的访问量都很大，因此需要容量大、速度快的存储系统存放数据，如采用高速大缓存硬盘，或者应用光纤通道外接到外置的专用磁盘系统。
- 具有高速CPU，以拥有较短的系统响应时间。数据库服务器必须应对大量的查询并做出适当的应答，因此要求处理能力强的CPU以满足较高的服务器处理速度和对客户的响应速率的要求。
- 有较高的数据传输能力，以提高数据传输率，保证足够的系统吞吐能力，否则，系统性能将形成瓶颈。
- 有足够的外存来进行数据备份。常配备磁盘阵列、磁带机等存储设备。
- 高稳定性的系统。即数据库系统能够持续稳定运行，能提供长时间可靠稳定的服务。

#### 4) 软件

软件是指负责数据库存取、维护和管理软件系统，通常叫作数据库管理系统(DBMS)。数据库系统的各类用户对数据库的各种操作请求，都是由DBMS来完成的，它是数据库系统的核心软件。DBMS提供一种超出硬件层之上的对数据库管理的功能，使数据库用户不受硬件层细节的影响。DBMS是在操作系统支持下工作的。

## 2. 数据库系统的特点

数据库系统具有如下特点。

### 1) 数据低冗余、共享性高

数据不再是面向某个应用程序而是面向整个系统。当前所有用户可同时存取数据库中的数据，从而减少了数据冗余，节约存储空间，同时也避免了数据之间的不相容性和不一致性。

### 2) 数据独立性提高

数据的独立性包括逻辑独立性和物理独立性。

- 数据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，因此应用程序可不必修改，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时，均可确保数据的逻辑独立性。
- 数据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，均可确保数据的物理独立性。

### 3) 有统一的数据控制功能

数据库可以被多个用户所共享，当多个用户同时存取数据库中的数据时，为保证数据库中数据的正确性和有效性，数据库系统提供了以下4个方面的数据控制功能。

- 数据的安全性(security)控制：可防止不合法使用数据造成数据的泄漏和破坏，保证数据的安全和机密。例如，系统提供口令检查或其他手段来验证用户身份，以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。
- 数据完整性(integrity)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。正确性是指数据的合法性，如代表年龄的整型数据，只能包含0~9，不能包含字母或特殊符号；有效性是指数据是否在其定义的有效范围内，如月份只能用1~12的数字来表示；相容性是指表示同一事实的两个数据应相同，否则就不相容，例如，一个人的性别不能既是男又是女。
- 并发(concurrency)控制：多个用户同时存取或修改数据库时，防止因相互干扰而提供给用户不正确的数据，并使数据库受到破坏的情况发生。
- 数据恢复(recovery)：当数据库被破坏或数据不可靠时，系统有能力将数据库从错误状态恢复到最近某一时刻的正确状态。

## 1.1.5 数据库管理系统(DBMS)

数据库管理系统是位于用户和数据库之间的一个数据管理软件，它的主要任务是对数据库的建立、运行和维护进行统一管理、统一控制，即用户不能直接接触数据库，而只能通过DBMS来操纵数据库。

### 1. DBMS概述

数据库管理系统负责数据库的统一操作、管理与维护，因此，DBMS是一种非常复杂的、综合性的，在数据库系统中对数据进行管理的大型系统软件，它是数据库系统的核心组成部分，在操作系统(OS)支持下工作。用户在数据库系统中的一切操作，包括数据定义、查询、更新及各种操作，都是通过DBMS完成的。

DBMS是数据库系统的核心部分，它把所有应用程序中使用的数据汇集在一起，并以记录为单位存储起来，便于应用程序查询和使用，如图1-2所示。

常见的DBMS有Access、Oracle、SQL Server、DB2、Sybase和MySQL等。不同的数据库管理系统有不同的特点。Access相对于其他的一些数据库管理软件，如SQL

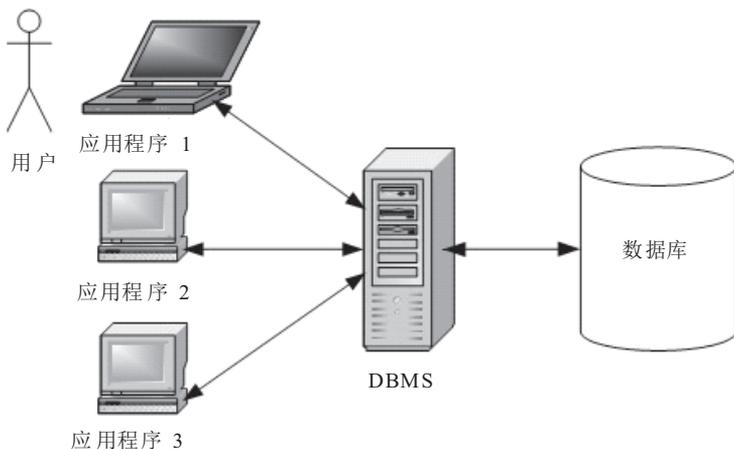


图1-2 DBMS、数据库以及与用户之间的关系

Server、Oracle等来说，操作相对简单，不需要用户具有高深的数据库知识，就能完成数据库所有的构造、检索、维护等操作，并且Access拥有简捷、美观的操作界面。

Access属于小型桌面数据库管理系统，通常用于办公管理。它允许用户通过构建应用程序来收集数据，并可以通过多种方式对数据进行分类、筛选，将符合要求的数据供用户查看，用户可以通过显示在屏幕上的窗体来查看数据库中的数据，也可以通过报表将相关数据打印出来，以便更详细地进行研究。

### 2. DBMS的功能

由于DBMS缺乏统一的标准，其性能、功能等许多方面随系统而异，通常情况下，DBMS提供了以下几个方面的功能。

- 数据库定义功能：DBMS提供相应数据定义语言定义数据库结构，刻画数据库的框架，并将其保存在数据字典中。数据字典是DBMS存取和管理数据的基本依据。
- 数据存取功能：DBMS提供数据操纵语言实现对数据库数据的检索、插入、修改和删除等基本存取操作。
- 数据库运行管理功能：DBMS提供数据控制功能，即数据的安全性、完整性和并发

控制等，从而对数据库运行进行有效的控制和管理，以确保数据库数据正确有效和数据库系统的有效运行。

- 数据库的建立和维护功能：包括数据库初始数据的装入，数据库的转储、恢复、重组织、系统性能监视、分析等功能。这些功能大都由DBMS的实用程序来完成。
- 数据通信功能：DBMS提供传输数据的功能，实现用户程序与DBMS之间的通信，这通常与操作系统协调完成。

### 3. DBMS的组成

DBMS大多是由许多系统程序组成的一个集合。每个程序都有各自的功能，一个或几个程序一起协调完成DBMS的一件或几件工作任务。各种DBMS的组成因系统而异，一般来说，它由以下几部分组成。

- 语言编译处理程序：语言编译处理程序主要包括数据描述语言翻译程序、数据操作语言处理程序、终端命令解释程序、数据库控制命令解释程序等。
- 系统运行控制程序：主要包括系统总控程序、存取控制程序、并发控制程序、完整性控制程序、保密性控制程序、数据存取与更新程序、通信控制程序等。
- 系统建立、维护程序：主要包括数据装入程序、数据库重组织程序、数据库系统恢复程序和性能监督程序等。
- 事务运行管理：提供事务运行管理及运行日志，事务运行的安全性监控和数据完整性检查，事务的并发控制及系统恢复等功能。
- 数据字典：数据字典通常是一系列表，它存储着数据库中有关信息的当前描述。它能帮助用户、数据库管理员和数据库管理系统本身使用和管理数据库。

## 1.1.6 数据库应用系统(DBAS)

数据库应用系统(Database Application System, DBAS)，是指在DBMS的支持下建立的计算机应用系统。数据库应用系统是由数据库系统、应用程序系统和用户组成的，具体包括：数据库、数据库管理系统、数据库管理员、硬件平台、软件平台、应用软件和应用界面。数据库应用系统的7个部分以一定的逻辑层次结构方式组成一个有机的整体。如以数据库为基础的电子银行系统就是一个数据库应用系统，无论是面向内部业务和管理的管理信息系统，还是面向外部，提供信息服务的开放式网上银行系统，从实现技术角度而言，都是以数据库为基础和核心的计算机应用系统。

## 1.2 数据库系统的体系结构

数据库系统的体系结构受数据库系统运行所在的计算机系统的影响很大，尤其是受计算机体系结构中的联网、并行和分布方面的影响。

- 计算机的联网可以实现某些任务在服务器系统上执行，而另一些任务在客户系统上执行。这种工作任务的划分导致了客户-服务器数据库系统的产生。

- 一个计算机系统中的并行处理能够加速数据库系统的活动，对事务做出更快速的响应，并且在单位时间中处理更多的事务。查询能够以一种充分利用计算机系统所提供的并行性的方式来处理。并行查询处理的需求引发了并行数据库系统的产生。
- 在一个组织机构的多个站点或部门间对数据进行分布，可以使得数据能够存放在产生它们或最需要它们的地方，而同时仍能被其他站点或其他部门访问。在不同站点上保存数据库的多个副本，使得大型组织机构甚至一个站点在受水灾、火灾、地震等自然灾害影响而遭到破坏的情况下，仍能继续进行数据库操作。分布式数据库系统用来处理地理上或管理上分布在多个数据库系统中的数据。

本节将从传统的集中式系统结构开始介绍数据库系统的体系结构，包括客户-服务器结构、并行系统结构和分布式系统结构。

## 1.2.1 集中式体系结构

集中式数据库系统是运行在一台计算机上、不与其他计算机系统交互的数据库系统。这样的数据库系统范围很广，既包括运行在个人计算机上的单用户数据库系统，也包括运行在高端服务器系统上的高性能数据库系统。

现代通用的计算机系统包括一到多个CPU，以及若干个设备控制器，它们通过公共总线连接在一起，但却提供对共享内存的访问，如图1-3所示。CPU具有本地的高速缓冲存储器，用于存放存储器中部分数据的本地副本，从而加快对数据库的访问。每个设备控制器负责一种类型的设备(如磁盘驱动器、音频设备或视频播放设备)。CPU和设备控制器可以并行地工作，它们同时竞争对存储器的访问。

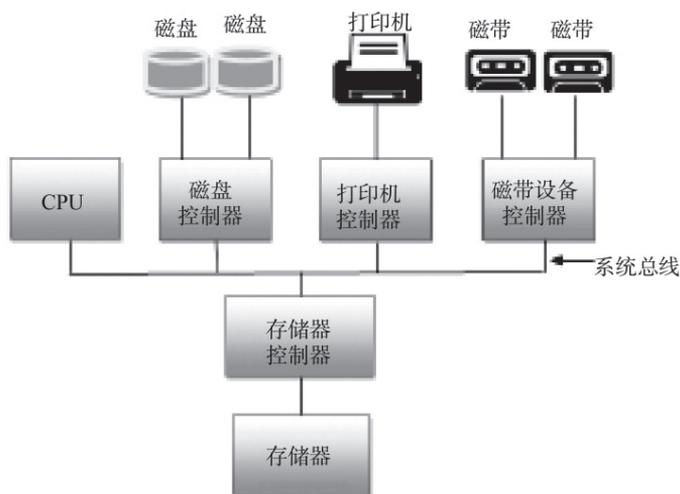


图1-3 集中式计算机系统

虽然当今通用的计算机系统有多个处理器，但它们具有粗粒度并行性，一般只有几个处理器(2~4个)，它们共享一个主存。在这样的机器上运行的数据库一般不将一个查询划分到多个处理器上，而是在每个处理器上运行一个查询，从而使多个查询能并行运行。因此，这样的系统能够提供较高的吞吐量，也就是说，尽管单个事务并没有运行得更快，但在单位时间内能运行更多的事务了。

## 1.2.2 客户—服务器结构

在客户—服务器结构中，数据库存放在服务器中，应用程序可以根据需要安排在服务器或客户工作站上，实现了客户端程序和服务器端程序的协同工作。这种结构解决了集中式结构和文件服务器结构的费用和性能问题。SQL Server和Oracle都支持客户—服务器结构。

## 1.2.3 并行系统结构

并行系统通过并行地使用多个CPU和磁盘来提高处理速度和I/O速度。并行计算机变得越来越普及，相应地使得并行数据库系统的研究变得更加重要。有些应用需要查询非常大的数据库，有些应用需要在每秒钟处理很大数量的事务，这些应用的需求推动了并行数据库系统的发展。

并行机器有若干种体系结构模式。图1-4所示是其中比较重要的几种，图中的M表示存储器，P表示处理器，圆柱体表示磁盘。

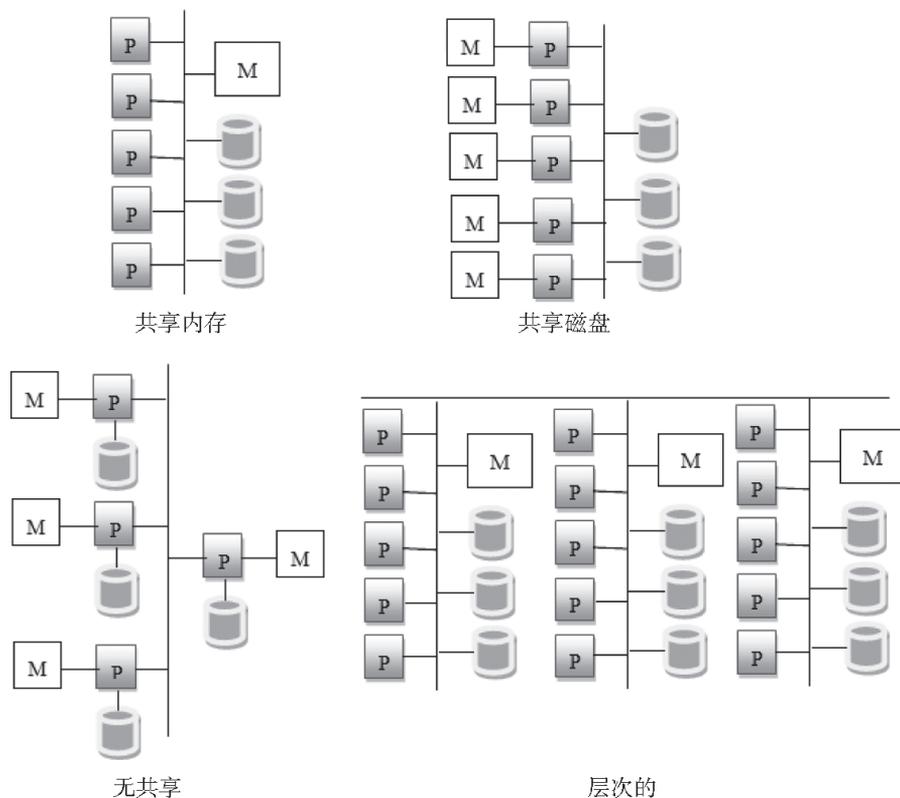


图1-4 并行数据库体系结构

- 共享内存：所有的处理器共享一个公共的存储器。
- 共享磁盘：所有的处理器共享一组公共的磁盘，共享磁盘系统有时又称作群集。
- 无共享：各个处理器既不共享公共的存储器，又不共享公共的磁盘。
- 层次的：这种模式是前几种体系结构的混合。

## 1.2.4 分布式系统结构

在分布式数据库系统中，数据库存储在几台计算机中。分布式系统中的计算机之间通过诸如高速网络或电话线等各种通信媒介互相通信。这些计算机不共享主存储器或磁盘。分布式系统中的计算机的规模和功能可大可小，小到工作站，大到大型机系统。

无共享并行数据库与分布式数据库之间的一个主要区别在于：分布式数据库一般是地理上分开的、分别管理的，并且是以较低的速度互相连接的。另一个主要的区别是：在分布式系统中，将事务区分为局部事务和全局事务。局部事务是仅访问在发起事务的站点上的数据的事务，而全局事务是需要访问发起事务的站点之外的某个站点或几个不同站点上的数据的事务。

建立分布式数据库系统有几个原因，主要包括共享数据、自治性和可用性。

- 共享数据：建立分布式数据库系统的主要优点就是它能提供一个环境，使一个站点上的用户可以访问存放在其他站点上的数据。例如，在一个分布式银行系统中，每一个支行存储与自己相关的数据，一个支行的用户可以访问另一个支行的数据。而如果没有这种功能，那么想要将资金从一个支行转到另一个支行的用户就必须求助于某种将已存在的系统相互关联起来的外部机制。
- 自治性：提供数据分布的方法来共享数据，其主要优点在于，每个站点可以对局部存储的数据保持一定程度的控制。在集中式系统中，中心站点的数据库管理员对数据进行控制。在分布式系统中，每个站点的局部数据库管理员可以有不同程度的局部自治，程度的大小依赖于分布式数据库系统的设计。
- 可用性：在分布式系统中，如果一个站点发生故障，其他站点可能还能继续运行。特别是，如果数据项在几个站点上进行了备份，需要某一个特定数据项的事务可以在这几个站点中的任何一个站点上找到该数据项。

## 1.3 数据模型

计算机不能直接处理现实世界中的具体事物，因此，必须通过将信息规范化整理和归类，然后才能将规范信息数据化并传入计算机的数据库中保存起来。这一过程经历了3个领域——现实世界、信息世界和数据(机器)世界。

- 现实世界：存在于人脑之外的客观世界，包括事物及事物之间的联系。
- 信息世界：是现实世界在人们头脑中的反映。
- 数据(机器)世界：将信息世界中的实体进行数据化，事物及事物之间的联系用数据模型来描述。

在现实世界中，常常用模型来对某个对象进行抽象或描述，如飞机模型，它反映了该飞机的大小、外貌特征及其型号等；并可用文字语言来对该对象进行抽象或描述。

为了用计算机来处理现实世界的事物，首先需要将它们反映到人的大脑中，即首先要把这些事务抽象为一种既不依赖于某一具体的计算机，又不受某一具体DBMS所左右的

信息世界的概念模型，然后再把该概念模型转换为某一具体DBMS所支持的计算机世界的数据库模型。信息的3个世界及其关系如图1-5所示。

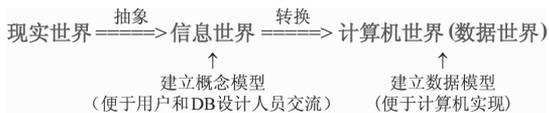


图1-5 信息的3个世界及其关系

这个过程是通过研究“过程 and 对象”，然后建立相应的数据模型来实现的。在这两个转换过程中，需要建立两个模型：概念模型和逻辑数据模型。

### 1.3.1 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。这类模型简单、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。这种信息结构并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。

概念模型主要用来描述世界的概念化结构，它使数据库的设计人员在设计的初始阶段，摆脱计算机系统及DBMS的具体技术问题，集中精力分析数据以及数据之间的联系等，与具体的数据管理系统无关。概念数据模型必须换成逻辑数据模型，才能在DBMS中实现。

在概念模型中主要有以下几个基本术语。

#### 1. 实体与实体集

实体是现实世界中可区别于其他对象的“事件”或物体。实体可以是人，也可以是物；可以指实际的对象，也可以指某些概念；还可以指事物与事物间的联系。例如，学生就是一个实体。

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。实体集不必互不相交，例如，可以定义学校所有学生的实体集students和所有教师的实体集teachers，而一个person实体可以是student实体，也可以是teacher实体，也可以都不是。

#### 2. 属性

实体通过一组属性来描述。属性是实体集中每个成员所具有的描述性性质。将一个属性赋予某实体集表明数据库为实体集中每个实体存储相似信息，但每个实体在每个属性上都有各自的值。一个实体可以由若干属性来刻画，如学生实体有学号、姓名、年龄、性别和班级等属性。

每个实体的每个属性都有一个值，例如，某个特定的student实体，其学号是201206304F2，姓名是邱舒娅，年龄是13，性别是女。

#### 3. 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，也称码。如学号是学生实体集的关键字，由于姓名有相同的可能，故不应作为关键字。

每个属性都有一个可取值的集合，称为该属性的域，或者该属性的值集。如姓名的域为字符串集合，性别的域为“男”和“女”。

#### 4. 联系

现实世界的事物之间总是存在某种联系，这种联系可以在信息世界中加以反映。一般存在两种类型的联系：一是实体内部的联系，如组成实体的属性之间的联系；二是实体与实体之间的联系。

两个实体之间的联系又可以分为如下3类。

- 一对一联系(1:1)：例如，一个班级有一个班主任，而每个班主任只能在一个班任职。这样班级和班主任之间就具有一对一的联系。
- 一对多联系(1:N)：例如，一个班有多个学生，而每个学生只可以属于一个班，因此，在班级和学生之间就形成了一对多的联系。
- 多对多的联系(M:N)：例如，学校中的学生与课程之间就存在着多对多的联系。每个学生可以选修多门课程，而每门课程也可以供多个学生选修。这种关系可以有多种处理方法。

### 1.3.2 用E-R方法表示概念模型

概念模型的表示方法很多，其中最著名的是E-R(Entity-Relationship，实体—联系)方法，它用E-R图来描述现实世界的概念模型。E-R图的主要成分是实体、联系和属性。E-R图通用的表现规则如下。

- 矩形：表示实体集。
- 椭圆：表示属性。
- 菱形：用菱形表示实体间的联系，菱形框内写上联系名。用无向边分别把菱形与有关实体相连接，在无向边旁标上联系的类型。如果实体之间的联系也具有属性，则把属性和菱形也用无向边连上。
- 线段：将属性连接到实体集或将实体集连接到联系集。
- 双椭圆：表示多值属性。
- 虚椭圆：表示派生属性。
- 双线：表示一个实体全部参与到联系集中。
- 双矩形：表示弱实体集。

E-R方法是抽象和描述现实世界的有力工具。用E-R图表示的概念模型与具体的DBMS所支持的数据模型独立，是各种数据模型的共同基础，因而比数据模型更一般、更抽象，更接近现实世界。

例如，画出某个学校学生选课系统的E-R图。学校每学期开设若干课程供学生选择，每门课程可接受多个学生选修，每个学生可以选修多门课程，每门课程有一个教师讲授，每个教师可以讲授多门课程。

首先，确定实体集和联系。在本例中，可以将课程、学生和教师定义为实体，学生和课程之间是“选修”关系，教师和课程之间是“讲授”关系。

接着，确定每个实体集的属性：“学生”实体的属性有学号、姓名、班级和性别；“课程”实体的属性有课程号、课程名和教科书；“教师”实体的属性有职工号、姓名和

性别。在联系中反映出教师讲授的课程信息、每门课程上课的学生数以及学生选修的所有课程。最终得到的E-R图如图1-6所示。

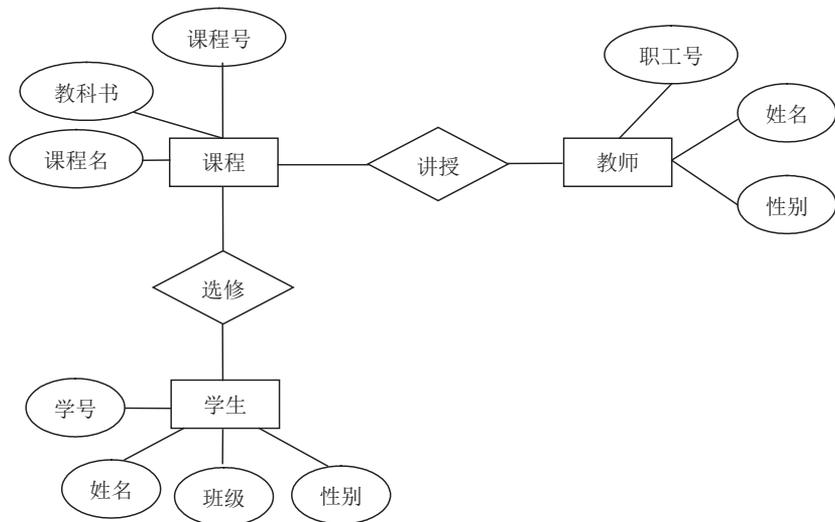


图1-6 学生选课系统的E-R图

### 1.3.3 逻辑数据模型

数据库中的数据是结构化的，这是按某种数据模型来组织的。当前流行的逻辑数据模型有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的联系的表示方式不同。层次模型是用树结构来表示数据之间的联系；网状模型是用图结构来表示数据之间的联系；关系模型是用二维表来表示数据之间的联系。

层次模型和网状模型是早期的数据模型。通常把它们统称为格式化数据模型，因为它们属于以“图论”为基础的表达方法。

按照这3类数据模型设计和实现的DBMS分别称为层次DBMS、网状DBMS和关系DBMS，相应地存在层次(数据库)系统、网状(数据库)系统和关系(数据库)系统等简称。下面分别对这3种数据模型做一个简单的介绍。

#### 1. 层次模型

层次模型是数据库系统最早使用的一种模型，它的数据结构是一棵有向树。层次结构模型具有如下特征。

- 有且仅有一个节点没有双亲，该节点是根节点。
- 其他节点有且仅有一个双亲。

在层次模型中，每个节点描述一个实体的型，称为记录类型。一个记录类型可有许多记录值，简称记录。节点间的有向边表示记录之间的联系。如果要存取某一记录类型的记录，可以从根节点起，按照有向树层次逐层向下查找。查找路径就是存取路径。

层次模型结构清晰，各节点之间联系简单，只要知道每个节点的(除根节点以外)双亲节

点，就可以得到整个模型结构，因此，画层次模型时可用无向边代替有向边。用层次模型模拟现实世界具有层次结构的事物及其之间的联系是很自然的选择方式，如表示“行政层次结构”“家族关系”等是很方便的。

层次模型的缺点是不能表示两个以上实体型之间的复杂联系和实体型之间的多对多联系。美国IBM公司1968年研制成功的IMS数据库管理系统就是这种模型的典型代表。

## 2. 网状模型

如果取消层次模型的两个限制，即两个或两个以上的节点都可以有多个双亲，则“有向树”就变成了“有向图”。“有向图”结构描述了网状模型。网状模型具有如下特征。

- 可有一个以上的节点没有双亲。
- 至少有一个节点可以有多于一个双亲。

网状模型由层次模型发展而来。从逻辑上看，它们都是基本层次联系的集合，用节点表示实体，用有向边(箭头)表示实体间的联系；从物理上看，它们的每一个节点都是一个存储记录，用链接指针来实现记录间的联系。当存储数据时这些指针就固定下来了，数据检索时必须考虑存取路径问题；数据更新时，涉及链接指针的调整，缺乏灵活性，系统扩充相当麻烦。网状模型中的指针更多，纵横交错，从而使数据结构更加复杂。

## 3. 关系模型

关系模型(Relational Model)是用二维表格结构来表示实体及实体之间联系的数据模型。关系模型的数据结构是一个“二维表框架”组成的集合，每个二维表又可称为关系，因此可以说，关系模型是“关系框架”组成的集合。

关系模型是使用最广泛的数据模型，目前大多数数据库管理系统都是关系型的，本书要介绍的Access就是一种关系数据库管理系统。

例如：对于某校学生、课程和成绩的管理，要用到如表1-1至表1-3所示的几个表格。如果要找到学生“邱舒娅”的“高等数学”成绩，首先需在学生信息表中找到“姓名”为“邱舒娅”的记录，记下她的学号201020202，如表1-1所示。

表1-1 学生信息表

学号	姓名	性别	年龄	院系ID	联系电话
982111056	葛冰	女	29	9001	13831705804
201400021	赵智暄	女	13	9002	15910806516
201021112	栾鹏	男	35	7482	13681187162
201020202	邱舒娅	女	22	1801	—
201231008	王兴宇	男	30	7012	13582107162

再到课程表中找到“课程名称”为“高等数学”的“课程号”：1003，如表1-2所示。

表1-2 课程表

课程号	课程名称	学 分	教师 ID
1001	经济学原理	3	91001
1002	社会心理学	4	61001
1003	高等数学	6	81002

接着到成绩表中查找“课程号”为1003，“学号”为201020202的对应成绩值，如表1-3所示。

表1-3 学生成绩表

课程号	学 号	成 绩
1001	982111056	91
1003	201020202	94
1003	944114044	52
1001	981000021	82

通过上面的例子可以看出，关系模型中数据的逻辑结构就是一张二维表，它由行和列组成。一张二维表对应了一个关系，表中的一行即为一条记录，表中的一列即为记录的一个属性。

关系模型的优点是：结构特别灵活，满足所有布尔逻辑运算和数学运算规则形成的查询要求；能搜索、组合和比较不同类型的数据；增加和删除数据非常方便。

其缺点是：数据库大时，查找满足特定关系的数据较费时；对空间关系无法满足。

## 1.4 关系数据库

关系数据库是当今世界的主流数据库。本节主要介绍关系模型中的一些基本术语，关系数据库中表之间的关系，关系模型的完整性约束。

### 1.4.1 关系模型中的基本术语

关系模型中经常用到的术语如下。

#### 1) 关系

一个关系就是一张二维表。

#### 2) 元组

二维表中的每一条记录就是一个元组，它是构成关系的一个个实体，可以说，“关系”是“元组”的集合，“元组”是属性值的集合，一个关系模型中的数据就是这样逐行逐列组织起来的。

### 3) 属性

二维表中的一列就是一个属性，又称为字段，第一行列出的是属性名(字段名)。属性的个数称为关系的元数或度。列的值称为属性值。

### 4) 域

属性的取值范围。例如，“性别”属性只能取值为“男”或“女”。

### 5) 分量

元组中的一个属性值。关系模型要求关系必须是规范化的，最基本的条件就是关系的每一个分量必须是一个不可分的数据项，即不允许表中还有表。

### 6) 关系模式

对关系的描述，一般表示如下：

关系名(属性1, 属性2, ……, 属性n)

例如，可以将学生关系描述为：

学生(学号, 姓名, 性别, 出生年月, 籍贯, 院系编号)

### 7) 候选关键字

候选关键字是数据库关系模式中的基本概念，指能唯一标识关系中元组且不含多余属性的属性集。

### 8) 关系数据库

对应于一个关系模型的所有关系的集合称为关系数据库。

### 9) 主关键字

一个关系中有多个候选关键字，可以选择其中一个作为主关键字，也称为主码或主键。

### 10) 外关键字

如果一个属性组不是所在关系的关键字，但它是其他关系的关键字，则该属性组称为外关键字，也称为外码或外键。

例如，描述院系的关系模式如下：

院系(院系编号, 院系名称)

其主键为“院系编号”，所以“学生”关系中的“院系编号”字段就是外键。

### 11) 主属性

包含在任一候选关键字中的属性称为主属性，不包含在任何候选关键字中的属性称为非关键字属性。

在关系模型中基本数据结构是二维表。记录之间的联系是通过不同关系中的同名属性来体现的。例如，要查找某个教师讲授的课程，首先要在“教师”关系中根据“姓名”查找到对应的教师“编号”，然后根据教师“编号”的值在“课程”关系中找到对应的“课程名”即可。在查询过程中，同名属性教师“编号”起到了连接两个关系的纽带作用。由此可见，关系模型中的各个关系模式不应当孤立起来，不是随意拼凑的一堆二维表，它必须满足相应的要求。

关系模型具有如下特征。

- 描述的一致性，不仅用关系描述实体本身，而且也用关系描述实体之间的联系。
- 可直接表示多对多的联系。
- 关系必须是规范化的关系，即每个属性是不可分的数据项，不允许表中有表。
- 关系模型是建立在数学概念基础上的，有较强的理论根据。

关系是一个二维表，但并不是所有的二维表都是关系。关系应具有以下性质。

- 每一列中的分量是同一类型的数据。
- 不同的列要给予不同的属性名。
- 列的次序可以任意交换。
- 一个关系中的任意两个元组不能完全相同。
- 行的次序可以任意交换。

## 1.4.2 关系数据库中表之间的关系

在关系数据库中，可以通过外部关键字来实现表与表之间的联系，公共字段是一个表的主键和另一个表的外键。图1-7所示的“学生”表和“院系”表都包含“院系ID”属性。

学号	姓名	性别	年龄	院系ID	联系电话
982111056	葛冰	女	29	9001	13831705804
201400021	赵智暄	女	13	9002	15910806516
201021112	栾鹏	男	35	7482	13681187162
201020202	邱舒娅	女	22	1801	—
201231008	王兴宇	男	30	7012	13582107162

院系ID	院系名称
9001	电子系
9002	软件学院
7482	法律系
1801	心理系
7012	医学院

图1-7 “学生”表和“院系”表之间的联系

通过这个字段就可以在“院系”和“学生”表之间建立联系，这个联系是一对多的联系，即一个院系中有多个学生。

## 1.4.3 关系模型的完整性约束

关系模型的完整性规则是对关系的某种约束条件，也就是说，关系的值随着时间变化应该满足一些约束条件。这些约束条件实际体现现实世界的要求。任何关系任何时刻都要满足这些语义约束。

关系模型中有3类完整性约束：实体完整性、参照完整性和用户定义的完整性。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，被称作关系的两个不变性，应该由关系系统自动支持。用户定义的完整性是应用领域需要遵循的约束条件，体现了具体领域中的语义约束。

### 1. 实体完整性(Entity Integrity)

实体完整性规则为：如果属性(指一个或一组属性)A是基本关系R的主属性，则A不能取空值。所谓空值，就是“不知道”或“不存在”的值。例如，在“学生”关系中，“学号”这个属性为主键，则该字段不能取空值。

按照实体完整性规则的规定，基本关系的主键都不能取空值。如果主键由若干属性组成，则所有这些主属性都不能取空值。

对于实体完整性规则说明如下。

(1) 实体完整性规则是针对基本关系而言的。一个基本表通常对应现实世界的一个实体集。例如，“学生”关系对应于学生的集合。

(2) 现实世界中的实体是可区分的，即它们具有某种唯一性标识。例如，每个学生都是独立的个体，是不一样的。

(3) 关系模型中以主键作为唯一性标识。

(4) 主键中的属性即主属性不能取空值。如果主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与(2)相矛盾，因此这个规则称为实体完整性。

### 2. 参照完整性(Referential Integrity)

参照完整性规则为：如果属性(或属性组)F是基本关系R的外键，它与基本关系S的主键Ks相对应(基本关系R和S不一定是不同的关系)，则对于R中每个元组在F上的值必须为空或者等于S中某个元组的主键值。

现实世界中的实体之间往往存在某种联系，在关系模型中，实体和实体之间的联系都是用关系来描述的，这样就自然存在着关系和关系间的引用。例如，图1-7中的“学生”表和“院系”表，“学生”表中每条学生记录的“院系ID”在“院系”表中必须存在，即学生所属的院系必须是该学校中已存在的院系。

#### 说明：

除不同关系之间存在参照完整性外，同一个关系的内部也可能存在参照完整性。

### 3. 用户定义的完整性(User-defined Integrity)

任何关系数据库系统都应该支持实体完整性和参照完整性。这是关系模型所要求的。除此之外，不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义的完整性就是针对某一具体关系数据库的约束条件。它反映某一具体应用所涉及的数据必须满足的语义要求。例如，某个属性必须取唯一值、某个非主属性也不能取空值、某个属性的取值范围为0~100(如学生的成绩)等。

关系模型应提供定义和检验这类完整性的机制，以使用统一的、系统的方法处理它们，而不应由应用程序承担这一功能。

## 1.5 关系代数

关系代数是一种抽象的查询语言，它用关系的运算来表达查询。

任何一种运算都是将一定的运算符作用于一定的运算对象之上，从而得到预期的结果，所以运算对象、运算符和运算结果是运算的三大要素。

关系代数的运算对象是关系，运算结果也是关系。关系代数用到的运算符包括4类：集合运算符、专门的关系运算符、比较运算符和逻辑运算符，如表1-4所示。

表1-4 关系代数用到的运算符

运算符		含义	运算符		含义
集合运算符	U	并	比较运算符	>	大于
	-	差		<	小于
	∩	交		≠	不等于
	×	笛卡儿积		≥	大于或等于
		≤		小于或等于	
专门的关系运算符	σ	选择	逻辑运算符	¬	非
	Π	投影		∧	与
	÷	除		∨	或
	⋈	连接			

按照运算符的不同，将关系代数的运算分为传统的集合运算和专门的关系运算两大类。其中，传统的集合运算将关系看成是元组的集合，其运算是从关系的“水平”方向即行的角度来进行的；而专门的关系运算同时涉及行和列的运算。比较运算符和逻辑运算符则是用来辅助专门的关系运算符进行操作的。

关于关系代数的理论，在这仅做简单介绍，详细信息请参考专门的数据库理论书籍。

### 1.5.1 传统的集合运算

传统的集合运算都是二目运算，包括并、差、交和笛卡儿积4种运算。

设关系R和关系S都具有n个属性，且相应的属性取自同一个域，t是元组变量， $t \in R$ 表示t是R的一个元组，如图1-8所示。

并、差、交、笛卡儿积运算介绍如下。

#### 1. 并

关系R和关系S的并运算记作：

$$R \cup S = \{t | t \in R \vee t \in S\}$$

其结果仍具有n个属性，由属于R或属于S的元组组成，结果如图1-9所示。

R			S		
A	B	C	A	B	C
a1	b1	c1	a1	b2	c2
a1	b2	c2	a1	b3	c2
a2	b2	c1	a2	b2	c1

图1-8 关系R和关系S

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1
a1	b3	c2

图1-9  $R \cup S$

## 2. 差

关系R和关系S的差记作：

$$R - S = \{t \mid t \in R \wedge t \notin S\}$$

其结果关系仍具有n个属性，由属于R而不属于S的所有元组组成，结果如图1-10所示。

A	B	C
a1	b1	c1

图1-10  $R - S$

## 3. 交

关系R和关系S的交记作：

$$R \cap S = \{t \mid t \in R \wedge t \in S\}$$

其结果关系仍具有n个属性，由既属于R又属于S的元组组成。关系的交也可用差来表示，结果如图1-11所示。

A	B	C
a1	b2	c2
a2	b2	c1

图1-11  $R \cap S$

## 4. 笛卡儿积

严格地讲，在这里的笛卡儿积应该是广义的笛卡儿积，因为这里的笛卡儿积的元素是元组。

两个分别具有n和m个属性的关系R和S的笛卡儿积是一个(n+m)列的元组的集合。元组的前n列是关系R的一个元组，后m列是关系S的一个元组。若R有k1个元组，S有k2个元组，则关系R和关系S的笛卡儿积有k1×k2元组。记作：

$$R \times S = \{t_r t_s \mid t_r \in R \wedge t_s \in S\}$$

结果如图1-12所示。

R.A	R.B	R.C	S.A	S.B	S.C
a1	b1	c1	a1	b2	c2
a1	b1	c1	a1	b3	c2
a1	b1	c1	a2	b2	c1
a1	b2	c2	a1	b2	c2
a1	b2	c2	a1	b3	c2
a1	b2	c2	a2	b2	c1
a2	b2	c1	a1	b2	c2
a2	b2	c1	a1	b3	c2
a2	b2	c1	a2	b2	c1

图1-12  $R \times S$

## 1.5.2 专门的关系运算

专门的关系运算包括选择、投影、连接、除运算等。下面简单介绍专门的关系运算。

### 1. 选择

从一个关系中选出满足给定条件的记录的操作称为选择或筛选。选择运算是从行的角度进行的运算，选出满足条件的那些记录构成原关系的一个子集，其中，条件表达式中可以使用=、<>、>=、>、<和<=等比较运算符，多个条件之间可以使用逻辑运算符AND(∧)、OR(∨)和NOT(¬)进行连接。选择操作记作：

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t) = \text{'真'}\}$$

其中，F表示选择条件。

假设对于表1-1所示的学生信息表Students，如果要查询年龄小于25岁的学生可以表示为：

$$\sigma_{\text{年龄}<25}(\text{Students}) \text{ 或 } \sigma_{4<25}(\text{Students})$$

运算的结果为学生信息表Students中所有年龄小于25的记录。这里的4表示Students表的第4列。

### 2. 投影

从一个关系中选出若干指定字段(列)的值的操作称为投影。投影是从列的角度进行的运算，所得到的字段个数通常比原关系少，或者字段的排列顺序不同。

投影操作记作：

$$\Pi_A(R) = \{t[A] \mid t \in R\}$$

其中，A为R中的属性列。

例如，查询学生的姓名和联系电话的操作如下：

$$\Pi_{\text{姓名, 联系电话}}(\text{students}) \text{ 或 } \pi_{2,6}(\text{students})$$

运算的结果为姓名和联系电话两列，以及这两列对应的所有数据组成的关系。

投影之后得到的关系不仅取消了原关系中的某些列，而且可能取消原关系中的某些元组，因为取消了某些列之后，就可能出现重复行，应取消这些完全相同的行。

### 3. 连接

连接是把两个关系中的记录按一定条件横向结合，生成一个新的关系。最常用的连接运算是自然连接，它是利用两个关系中公用的字段，将该字段值相等的记录连接起来。

需要明确的是，选择和投影都属于单目运算，它们的操作对象只是一个关系，而连接则是双目运算，其操作对象是两个关系。

连接操作记作：

$$R \bowtie_{A\theta B} S = \{t_r, t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta t_s[B]\}$$

其中，A和B分别是R和S上度数相等且可比的属性组；θ是比较运算符。常见的连接是自然连接，自然连接是一种特殊的等值连接，所以在介绍自然连接之前，先介绍等值连接

的定义：等值连接是  $\theta$  为 = 的连接。

例如，图1-13所示的关系R和S，因为两个关系中都有B属性，等值连接  $R \bowtie_{R.B=S.B} S$  的结果如图1-14所示。

R		
A	B	C
$a_1$	$b_1$	5
$a_1$	$b_2$	6
$a_2$	$b_3$	8
$a_2$	$b_4$	12

S	
B	E
$b_1$	3
$b_2$	7
$b_3$	10
$b_3$	2
$b_5$	2

图1-13 参与连接运算的关系R和关系S

$R \bowtie_{R.B=S.B} S$				
A	R.B	C	S.B	E
$a_1$	$b_1$	5	$b_1$	3
$a_1$	$b_2$	6	$b_2$	7
$a_2$	$b_3$	8	$b_3$	10
$a_2$	$b_3$	8	$b_3$	2

图1-14 等值连接

很显然，在图1-14中R.B和S.B是重复的属性列，没有必要都出现在结果中，这种在结果中把重复属性列去掉的等值连接就是自然连接，记作： $R \bowtie S$ 。

自然连接运算首先计算参与运算的两个关系的笛卡儿积，然后基于两个关系中都出现的属性上的相等性进行选择，最后，还要从结果中去除重复属性列。所以上面的R和S自然连接的结果如图1-15所示。

对于一般的连接， $\theta$  可以是其他比较运算，如  $R \bowtie_{C < E} S$  的结果如图1-16所示。

$R \bowtie S$			
A	B	C	E
$a_1$	$b_1$	5	3
$a_1$	$b_2$	6	7
$a_2$	$b_3$	8	10
$a_2$	$b_3$	8	2

图1-15 自然连接

$R \bowtie_{C < E} S$				
A	R.B	C	S.B	E
$a_1$	$b_1$	5	$b_2$	7
$a_1$	$b_1$	5	$b_3$	10
$a_1$	$b_2$	6	$b_2$	7
$a_1$	$b_2$	6	$b_3$	10
$a_2$	$b_3$	8	$b_3$	10

图1-16 一般连接

例如，“学生信息表”和“学生成绩表”可以通过“Students.学号”和“Scores.学号”作为连接的“纽带”。查询“赵智暄”的成绩可以表示为：

$$\sigma_{\text{姓名}=\text{赵智暄}}(\text{Students} \bowtie \text{Score})$$

#### 4. 除运算

给定关系R(X,Y)和S(Y,Z)，其中X、Y、Z为属性组。R中的Y和S中的Y可以有不同的属性名，但必须出自相同的域集。

那么R和S的除运算得到一个新的关系P(X)，P是R中满足下列条件的元组在X属性列上的投影：元组在X上分量值x的象集 $Y_x$ 包含S在Y上投影的集合。

除运算记作：

$$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_y(S) \subseteq Y_x\}$$

其中， $Y_x$ 为x在R中的象集， $x=t_r[X]$ 。

## 1.6 规范化理论

为了使数据库设计的方法趋于完善，人们研究了规范化理论。目前规范化理论的研究已经有了很大的发展。本节将主要介绍模式规范化在数据库设计过程中的必要性及其规范化原理。

### 1.6.1 非规范化的关系

一般而言，关系数据库设计的目标是生成一组关系模式，使用户既无须存储不必要的重复信息，又可以方便地获取信息。方法之一是设计满足适当范式的模式。在学习范式前，首先来了解非规范化的表格。

- 当一个关系中的所有字段都是不可分割的数据项时，称该关系是规范化的。但是，当表格中有一个字段含有组合数据项时，即为不规范化的表格，如图1-17所示。
- 当表格中含有多值数据项时，该表格同样为不规范化的表格，如图1-18所示。

编号	店名	季度营业额			
		第一季度	第二季度	第三季度	第四季度

图1-17 字段含有组合数据项的不规范化表格

教师编号	教师姓名	年龄	教授科目
101	郑颖	35	数据挖掘
102	王海泉	36	计算机网络 网络协议与安全

图1-18 多值数据项的不规范化表格

满足一定条件的关系模式称为范式(Normal Form, NF)。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)、Boyce-Codd范式(BCNF)、第四范式(4NF)和第五范式(5NF，又称完美范式)。规范化程度较高者必是较低者的子集。满足最低要求的范式是第一范式(1NF)。一个低级范式的关系模式，通过投影分解的方法可转换成多个高一级范式的关系模式的集合，这个过程称为规范化。一般情况下，数据库只需满足第三范式(3NF)就行了。

### 1.6.2 第一范式(1NF)

在1971年至1972年，关系数据模型的创始人E.F.Codd系统地提出了第一范式(1NF)、第二范式(2NF)和第三范式(3NF)的概念。

在关系模式R的所有属性的值域中，如果每个值都是不可再分解的值，则称R属于第一范式(1NF)。第一范式的模式要求属性值不可再分成更小的部分，即属性项不能由属性组合或组属性组成。

第一范式是最低的规范化要求，它要求关系满足一种最基本的条件，它与其他范式不同，不需要诸如函数依赖之类的额外信息。

第一范式要求数据表不能存在重复的记录，即关键字取值唯一，第二个要求是每个字段都已经分到最小不可再分，关系数据库的定义就决定了数据库满足这一条。主关键字应满足下面几个条件。

- 主关键字在表中是唯一的。
- 主关键字段不存在空值。

- 每条记录都必须有一个主关键字。
- 主关键字是关键字的最小子集。

从非规范化关系转换为1NF的方法很简单，以图1-17和图1-18所示的表格为例，分别进行如图1-19和图1-20所示的转变，即可满足第一范式的关系。

编号	店名	第一季度	第二季度	第三季度	第四季度

图1-19 横向展开成第一范式关系

教师编号	教师姓名	年龄	教授科目
101	郑颖	35	数据挖掘
102	王海泉	36	计算机网络
102	王海泉	36	网络协议与安全

图1-20 纵向展开成第一范式关系

满足第一范式的关系模式有许多不必要的重复值，并且增加了修改数据时疏漏的可能性，为了避免这种数据冗余和更新数据的疏漏，就引出了第二范式。

### 1.6.3 第二范式(2NF)

如果一个关系属于第一范式(1NF)，且所有的非主关键字段都完全依赖于主关键字，则称之为第二范式。第二范式(2NF)是在第一范式(1NF)的基础上建立起来的，即满足第二范式(2NF)必须先满足第一范式(1NF)。

例如，有一个存储物品的关系有5个字段：

物品(物品ID、仓库号、物品名称、物品数量、仓库地址)

这个关系符合1NF，其中“物品ID”和“仓库号”构成主关键字，但因为“仓库地址”只完全依赖于“仓库号”，即只依赖于主关键字的一部分，所以它不符合第二范式(2NF)。这样首先存在数据冗余，因为仓库数量可能不多。其次，在更改仓库地址时，如果漏改了某一条记录，存在数据不一致性。再次，如果某个仓库的物品全部出库了，那么这个仓库地址就会丢失，所以这种关系不允许存在某个仓库中不放物品的情况。可以用投影分解的方法消除部分依赖的情况，从而达到2NF的标准。方法是从关系中分解出新的二维表，使得每个二维表中所有的非关键字都完全依赖于各自的主关键字。这里可以做如下分解，将原来的一个表分解成两个表：

物品(物品ID，仓库号，物品名称，物品数量)

仓库(仓库号，仓库地址)

这样就完全符合第二范式(2NF)了。

图1-20所示的表格虽然已经符合1NF的要求，但表中仍然存在着数据冗余和潜在的数据更新异常风险。此时，可以将表格分解成两个关系，如图1-21所示。

教师编号	教师姓名	年龄
101	郑颖	35
102	王海泉	36

教师编号	教授科目
101	数据挖掘
102	计算机网络
102	网络协议与安全

图1-21 展开成第二范式关系

## 1.6.4 第三范式(3NF)

如果一个关系属于第二范式(2NF),且每个非关键字段(非主属性)不传递依赖于主关键字,这种关系就是第三范式(3NF)。简而言之,从2NF中消除传递依赖,就是3NF。如有一个关系(姓名,工资等级,工资额),其中姓名是关键字,此关系符合2NF,但是因为工资等级决定工资额,这就叫传递依赖,它不符合3NF。同样可以使用投影分解的方法将上表分解成两个表:(姓名,工资等级)和(工资等级,工资额)。

上面提到了投影分解的方法,关系模式的规范化过程是通过投影分解来实现的。这种把低一级关系模式分解成若干个高一级关系模式的投影分解方法不是唯一的,应该在分解中满足3个条件。

- 无损连接分解,分解后不丢失信息。
- 分解后得到的每个关系都是高一级范式,不要同级甚至低级分解。
- 分解的个数最少,这就是完美要求,应该做到尽量少。

从以上内容可知,规范化的基本思想是逐步消除数据依赖中不合适的部分,使模式中的各种关系模式达到某种程度的“分离”,即“一事一地”的模式设计原则。让一个关系描述一个概念、一个实体或者实体间的一种联系。如果多于一个概念,就把它分离出去。因此,所谓规范化实质上是概念的单一化。

规范化的优点是明显的,它避免了大量的数据冗余,节省了空间,保持了数据的一致性,如果完全达到3NF,用户不会在两个以上的地方更改同一个值,而当记录会经常发生改变时,这个优点便很容易显现出来。但是,它最大的不利是,由于用户把信息放置在不同的表中,增加了操作的难度,同时把多个表连接在一起的时间花费也是巨大的。节省了时间必然付出了空间的代价;反之,节省了空间也必然要付出时间的代价,时间和空间在计算机领域中是一个矛盾统一体,它们是互相作用、对立统一的。

## 1.6.5 其他范式

一般情况下,规范化到3NF就满足需要了,规范化程度更高的还有BCNF、4NF、5NF。

### 1. BCNF

Boyce-Codd范式(BCNF)是3NF的一个子集,即满足BCNF必须满足3NF。通常情况下,BCNF被认为没有新的设计规范加入,只是对第二范式与第三范式中的设计规范要求更强,因而被认为是对第三范式的修正,使数据库冗余度更小。这也是BCNF不被称为第四范式的原因。某些资料显示,根据范式要求的递增性将其称之为第四范式是不规范的。真正的第四范式,则是在设计规范中添加了对多值依赖的要求。

BCNF的判定如下:如果关系模式R的所有属性(包括主属性和非主属性)都不传递依赖于R的任何候选关键字,那么称关系R属于BCNF。换句话说,对于关系模式R,如果每个决定因素都包含关键字(而不是被关键字所包含),则R属于BCNF。

对符合3NF的关系进行投影,消除原关系中主属性对键的部分与传递依赖,即可得到一组BCNF关系。

例如，有如下仓库存储物品的管理关系：

仓库存储管理(仓库ID, 物品ID, 管理员ID, 数量)

通常，一个管理员只在一个仓库工作，一个仓库可以存储多种物品。这个数据库表中存在如下决定关系：

(仓库ID, 物品ID) → (管理员ID, 数量)

(管理员ID, 物品ID) → (仓库ID, 数量)

所以，(仓库ID, 物品ID)和(管理员ID, 物品ID)都是“仓库管理”的候选关键字，关系中的唯一非关键字段为数量，它是符合3NF的。但是，由于存在如下决定关系：

(仓库ID) → (管理员ID)

(管理员ID) → (仓库ID)

即存在关键字段决定关键字段的情况，所以不符合BCNF。消除主属性对候选关键字的依赖关系，可以把上述关系分解为如下两个：

仓库管理 (仓库ID, 管理员ID)；

仓库存储 (仓库ID, 物品ID, 数量)。

这样就符合BCNF了。

## 2. 第四范式(4NF)

第四范式的定义用到了多值依赖，多值依赖的定义如下：设 $R(U)$ 是属性集 $U$ 上的一个关系模式。 $X$ 、 $Y$ 、 $Z$ 是 $U$ 的子集，并且 $Z=U-X-Y$ 。关系模式 $R(U)$ 中多值依赖 $X \twoheadrightarrow Y$ 成立，当且仅当对 $R(U)$ 的任一关系 $r$ ，给定的一对 $(x, z)$ 值有一组 $Y$ 的值，这组值仅仅决定于 $x$ 值而与 $z$ 值无关。

4NF是消除关系中的多值依赖。

例如，有一个用户联系方式表：

客户联系方式(CUSTID,PHONE,MOBILE)。

CUSTID为用户ID，PHONE为用户的固定电话，MOBILE为用户的移动电话。

本来，这是一个非常简单的BCNF表。主键为CUSTID，不存在传递依赖。但在某些情况下，这样的表还是不合理的。因为有的用户可能有两个固定电话，或多个移动电话。由于PHONE和MOBILE是互相独立的，而有些用户又有两个或多个值。这时此表就违反了第四范式。在这种情况下，此表的设计就会带来很多维护上的麻烦。解决这一问题的方法为，设计一个新表：

新客户联系方式(CUSTID,NUMBER,TYPE)

这样就可以对每个用户处理不同类型的多个电话号码，而不会违反第四范式。

**说明：**

第四范式的应用范围比较小，只有在某些特殊情况下，要考虑将表规范到第四范式。所以在实际应用中，一般不要求表满足第四范式。

**3. 第五范式(5NF)**

第四范式不是“最终”范式，多值依赖有助于理解并解决利用函数依赖无法理解的某些形式的信息重复。还有一些类型的概括多值依赖的约束称为连接依赖(Join Dependence)，由此引出的另外一种范式称为投影-连接范式(Project-Join Normal Form, PJNF)，有的书中也将其称为第五范式(5NF)。

5NF是在4NF的基础上做的进一步规范化。4NF处理的是相互独立的多值情况，而5NF则处理相互依赖的多值情况。

第五范式要求表必须可以分解为较小的表，除非那些表在逻辑上拥有与原始表相同的主键。例如，有一个表示销售信息的关系：

SVP(SALEID, VENDOR, PRODUCT)

SALEID代表销售人员，VENDOR代表供货商，PRODUCT代表产品。

在某些情况下，这个表中会产生一些冗余。可以将表分解为

SV(SALEID, VENDOR)

SP(SALEID, PRODUCT)

VP(VENDOR, PRODUCT)

因为一个销售人员可以销售多个产品，同一产品也可能有多个供货商，所以SVP仅有的候选码(SNO,PNO,JNO)肯定不是它的3个投影SV、SP、VP自然连接的公共属性，因此，SVP不属于5NF。

由此可见，5NF的应用就更少了，所以本书不对其做过多介绍。

规范化的过程就是在数据库表设计时移除数据冗余的过程。随着规范化的进行，数据冗余越来越少，但数据库的效率也越来越低。在数据库设计中，需要结合实际应用的性能要求，规范到合适的范式。一般情况下，规范到3NF就够了。

## 1.7 数据库语言

数据库系统提供了两种不同类型的语言：一种是数据定义语言，用于定义数据库模式；另一种是数据操纵语言，用于表达数据库的查询和更新。而实际上，数据定义和数据操纵语言并不是两种分离的语言，相反，它们构成了单一的数据库语言，如广泛使用的SQL语言。

## 1.7.1 数据定义语言(DDL)

数据库模式是通过一系列定义来说明的, 这些定义由一种称为数据定义语言(Data-Definition Language, DDL)的特殊语言来表达。例如, 下面的SQL语句描述了Students表的定义:

```
Create table Students
(Sno varchar(10),
Sname varchar(50),
Ssex varchar(4),
Sage integer,
Sdeptno integer,
Stelephone varchar(20))
```

## 1.7.2 数据操纵语言(DML)

数据操纵语言(Data-Manipulation Language, DML)使得用户可以访问或操纵那些按照某种特定数据模式组织起来的数据。数据操纵包括对存储在数据库中的信息进行检索, 向数据库中插入新的信息, 从数据库中删除信息和修改数据库中存储的信息。

通常有以下两种基本的数据操纵语言。

- 过程化DML: 要求指定需要什么数据以及如何获得这些数据。
- 陈述式DML: 也称非过程化DML, 只要求用户指定需要什么数据, 而不指明如何获得这些数据。

通常陈述式DML比过程化DML更易学易用。但是, 由于未指明如何获得数据, 因此数据库系统会指出一种访问数据的高效路径。SQL语言的DML部分是非过程化的。

查询是要求对信息进行检索的语句。DML中涉及信息检索的部分称为查询语句。例如, 下面的语句将从Students表中查询名为“邱舒娅”的用户信息:

```
SELECT * FROM Students WHERE cname='邱舒娅';
```

# 1.8 数据库设计

数据库设计是指对于一个给定的应用系统, 构造(设计)优化的数据库逻辑模式和物理结构, 并据此建立数据库及其应用系统, 使之能够有效地存储和管理数据, 满足各种用户的应用需求, 包括信息管理要求和数据操作要求。

- 信息管理要求是指在数据库中应该存储和管理的数据对象。
- 数据操作要求是指对数据对象需要进行的操作, 如查询、增加、删除、修改和统计等操作。

## 1.8.1 数据库设计的目标

数据库设计的目标是为用户和各种应用系统提供一个信息基础设施和高效率的运行环境。高效率的运行环境包括数据库数据的存取效率、数据库存储空间的利用率以及数据库系统运行管理的效率等。

## 1.8.2 数据库设计的特点

数据库设计和一般的软件系统的设计、开发、运行与维护有许多相同之处，也有其自身的一些特点。

### 1. 数据库建设的基本规律

“三分技术，七分管理，十二分基础数据”是数据库设计的特点之一。

在数据库建设中，不仅涉及技术，还涉及管理。要建设好一个数据库应用系统，开发技术固然重要，但是相比之下管理则更加重要。这里的管理不仅包括数据库建设作为一个大型的工程项目本身的项目管理，而且包括该企业的业务管理。

“十二分基础数据”则强调了数据的收集、整理、组织和不断更新是数据库建设中的重要环节。人们往往忽视基础数据在数据库建设中的地位和作用。基础数据的收集、入库是数据库建立初期工作量最大、最烦琐、最细致的工作。在以后数据库运行过程中更需要不断地把新的数据加入数据库中，使数据库成为一个“活库”，否则就成为“死库”。数据库一旦成了“死库”，系统也就失去了应用价值，原来的投资也就失败了。

### 2. 结构(数据)设计和行为(处理)设计相结合

数据库设计应该和应用系统相结合。也就是说，整个设计过程中要把数据库结构设计和对数据的处理设计密切结合起来。这是数据库设计的特点之二。

在早期的数据库应用系统开发过程中，常常把数据库设计和应用系统的设计相分离。由于数据库设计有它专门的技术和理论，因此，需要专门来讲解数据库设计。这并不等于数据库设计和在数据库之上开发应用系统是相互分离的。相反，必须强调设计过程中数据库设计和应用程序的密切结合，并把它作为数据库设计的重要特点。

## 1.8.3 数据库设计的方法

大型数据库设计是涉及多学科的综合性的技术，同时又是一项庞大的工程项目。它要求从事数据库设计的专业人员具备多方面的技术和知识，主要包括：计算机的基础知识、软件工程的原理和方法、程序设计的方法和技巧、数据库的基本知识、数据库设计技术和应用领域的知识等。这样才能设计出符合具体领域要求的数据库及其应用系统。

早期数据库设计主要采用手工与经验相结合的方法。设计的质量往往与设计人员的经验与水平有直接的关系。数据库设计是一种技艺，缺乏科学理论和工程方法的支持，设计质量难以保证。因此，人们努力探索，提出了各种数据库设计方法，其中比较著名的有以下4种。

- 新奥尔良(New Orleans)方法：该方法把数据库设计分为若干阶段和步骤，并采用一些辅助手段实现每一过程。它运用软件工程的思想，按一定的设计规程用工程化方法设计数据库。新奥尔良方法属于规范化设计法。虽然从本质上看它仍然是手工设计方法，其基本思想是过程迭代和逐步求精。
- 基于E-R模型的数据库设计方法：该方法用E-R模型来设计数据库的概念模型，是数据库概念设计阶段广泛采用的方法。
- 3NF(第三范式)设计方法：该方法用关系数据理论为指导来设计数据库的逻辑模型，是设计关系数据库时在逻辑阶段可以采用的一种有效方法。
- ODL(Object Definition Language)方法：这是面向对象的数据库设计方法。该方法用面向对象的概念和术语来说明数据库结构。ODL可以描述面向对象的数据库结构设计，可以直接转换为面向对象的数据库。

## 1.8.4 数据库设计的步骤

数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求。

数据库设计一般分为以下6个步骤。

### 1. 需求分析

进行数据库设计首先必须准确了解与分析用户需求，包括数据和处理。需求分析是整个设计过程的基础，是最困难、最耗时的一步。作为“地基”的需求分析是否做得充分与准确，决定了在其上构建数据库大厦的速度与质量。需求分析做得不好，可能会导致整个数据库设计返工重做。

需求分析的重点是调查、收集与分析用户在数据管理中的信息要求、处理要求、安全性与完整性要求。

常用的调查方法包括以下几种。

- (1) 跟班作业：通过亲身参加业务工作了解业务情况，能比较准确地理解用户的需求，但比较耗时。
- (2) 开调查会：与用户座谈来了解业务活动情况及用户需求。
- (3) 请专人介绍：请了解业务的专业人员介绍。
- (4) 询问：对某些调查中的问题，可以找专人询问。
- (5) 设计调查表：请用户填写。
- (6) 查阅记录：查阅与原系统有关的数据记录。

### 2. 概念结构设计

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。

概念模型是整个组织各个用户关心的信息结构。描述概念结构的有力工具是E-R图。数据库设计通常基于E-R模型来进行，然后转换成关系模型。

### 3. 逻辑结构设计

逻辑结构设计将概念结构转换为某个DBMS所支持的数据模型，并对其进行优化。逻辑结构设计的步骤如下。

- (1) 将概念结构转换为一般的关系、网状或层次模型。
- (2) 将转换来的关系、网状或层次模型向特定DBMS支持下的数据模型转换。
- (3) 对数据模型进行优化。
- (4) 设计用户子模式。

### 4. 物理结构设计

物理结构设计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等。物理结构设计通常分为以下两步。

- 确定数据库的物理结构：可分为确定数据的存取方法和数据的存储结构。
- 对物理结构进行评估：包括对时间效率、空间效率、维护开销和各种用户要求进行权衡，从多种设计方案中选择一个较优的方案。

### 5. 数据库实施

在数据库实施阶段，设计人员运用DBMS提供的数据库语言(如SQL)及其宿主语言，根据逻辑设计和物理设计的结果建立数据库，编制与调试应用程序，组织数据入库，并进行调试运行。

#### 1) 定义数据库结构

确定数据库的逻辑结构与物理结构后，就可以用所选用的DBMS提供的数据库定义语言(DDL)来严格描述数据库结构。

#### 2) 数据装载

数据库结构建立后，即可向数据库中装载数据。组织数据入库是数据库实施阶段最主要的工作。对于数据量不是很大的小型系统，可以用人工方式完成数据的入库，具体包括如下几个步骤。

- 筛选数据：需要装入数据库中的数据通常都分散在各个部门的数据文件或原始凭证中，所以首先必须把需要入库的数据筛选出来。
- 转换数据格式：筛选出来的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时可能很复杂。
- 输入数据：将转换好的数据输入计算机中。
- 校验数据：检查输入的数据是否有误。

对于中大型系统，由于数据量大，用人工方式组织数据入库将会耗费大量的人力物力，而且很难保证数据的正确性，因此应该设计一个数据输入子系统，由计算机辅助数据的入库工作。

#### 3) 编制与调试应用程序

数据库应用程序的设计应该与数据设计并行进行。在数据库实施阶段，当数据库结构建立好后，就可以开始编制与调试数据库的应用程序，也就是说，编制与调试应用程序是与组织数据入库同步进行的。调试应用程序时由于数据入库尚未完成，可先使用模拟数据。

#### 4) 数据库试运行

应用程序调试完成，并且已有少部分数据入库后，即可开展数据库的试运行。数据库试运行也称为联合调试，其主要工作如下。

- 功能测试：即实际运行应用程序，执行对数据库的各种操作，测试应用程序的各种功能。
- 性能测试：即测量系统的性能指标，分析是否符合设计目标。

### 6. 数据库运行和维护

数据库应用系统经过试运行后即可投入正式运行。数据库投入运行标志着开发任务的基本完成和维护工作的开始，并不意味着设计过程的终结，由于应用环境在不断变化，数据库运行过程中物理存储也会不断变化，对数据库进行评价、调整、修改等维护工作是一个长期的任务，也是设计工作的继续和提高。

在数据库运行阶段，对数据库经常性的维护工作主要是由DBA完成的，主要包括以下内容。

#### 1) 数据库的转储和恢复

定期对数据库和日志文件进行备份，以保证一旦发生故障，能利用数据库备份及日志文件备份，尽快将数据库恢复到某种一致性状态，并尽可能减少对数据库的破坏。

#### 2) 数据库的安全性、完整性控制

DBA必须对数据库的安全性和完整性控制负起责任，根据用户的实际需要授予不同的操作权限。另外，由于应用环境的变化，数据库的完整性约束条件也会变化，也需要DBA不断修正，以满足用户要求。

#### 3) 数据库性能的监督、分析和改进

目前，许多DBMS产品都提供了监测系统性能参数的工具，DBA可以利用这些工具方便地得到系统运行过程中一系列性能参数的值。DBA应该仔细分析这些数据，通过调整某些参数来进一步改进数据库性能。

#### 4) 数据库的重组和重构

数据库运行一段时间后，由于记录不断被增、删、改，会使数据库的物理存储变坏，从而降低数据库存储空间的利用率和数据的存取效率，使数据库的性能下降。这时，DBA就要对数据库进行重组，或部分重组(只对频繁增、删的表进行重组)。数据库的重组不会改变原设计的数据逻辑结构和物理结构，只是按原设计要求重新安排存储位置，回收垃圾，减少指针链，提高系统性能。DBMS一般会提供重组数据库时使用的实用程序，帮助DBA重组数据库。

数据库应用环境发生变化会导致实体及实体间的联系也发生相应的变化，使原有的数据库设计不能很好地满足新的需求，从而不得不适当调整数据库的模式和内模式，这就是数据库的重构造。DBMS都提供了修改数据库结构的功能。

重构造数据库的程度是有限的。如果应用变化太大，已无法通过重构数据库来满足新的需求，或重构数据库的代价太大时，则表明现有数据库应用系统的生命周期已经结束，应该重新设计新的数据库系统，开始新数据库应用系统的生命周期。

**提示:**

设计一个完善的数据库应用系统不是一蹴而就的，往往是上述6个阶段的不断反复。

## 1.9 Access概述

Access是美国Microsoft公司推出的关系数据库管理系统(RDBMS)，它是Microsoft Office的组成部分之一，具有与Word、Excel和PowerPoint等相似的操作界面，深受广大用户的喜爱。本书将通过Access 2021版本详细介绍Access的功能和使用技巧。

### 1.9.1 Access简介

Access是一个面向对象的、采用事件驱动的新型关系数据库。它提供了表生成器、查询生成器、宏生成器和报表设计器等许多可视化的操作工具，以及数据库向导、表向导、查询向导、窗体向导、报表向导等多种向导，使用户能够很方便地构建一个功能完善的数据库系统。

#### 1. 概述

Access能操作其他来源的数据，包括许多流行的PC数据库(如DBASE、Paradox、FoxPro)和服务于、小型机及大型机上的许多SQL数据库。此外，Access还提供了Windows操作系统的高级应用程序开发系统。与其他数据库开发系统相比，Access有一个明显的区别，就是用户不需要编写一行代码，就可以在很短的时间里开发出一个功能强大且相当专业的数据库应用程序，并且这一过程是完全可视的，如果能给它加上一些简短的VBA代码，那么开发出的程序就能与专业程序员开发的程序相媲美。

Access的最主要优点是它不用携带向上兼容的软件。无论是对于有经验的数据库设计人员还是那些刚刚接触数据库管理系统的新手，都会发现Access所提供的各种工具既实用又方便，同时还能够获得高效的数据处理能力。

#### 2. Access发展历程

Access数据库系统是一个关系数据库系统，它经历了一个长期的发展过程。

Microsoft公司在1990年5月推出Windows 3.0，该程序一经推出立刻受到了用户的欢迎和喜爱。1992年11月Microsoft公司发行了Windows关系数据库系统Access 1.0版本。从此，Access被不断改进和再设计。自1995年起，Access成为办公软件Office 95的一部分。多年来，Microsoft先后推出的Access版本有2.0、7.0/95、8.0/97、9.0/2000、10.0/2002、2003，到2007年，Access 2007以一种全新的界面和文件格式面世，取消传统菜单的操作方式而之以功能区，这是Access 2007的明显改进之一。用户可以在功能区中进行绝大多数的数据库管理相关操作。随后的几年中，在Access 2007的基础上，微软又相继推出了2010、2013、2016和2021版本。本书将介绍Access 2021版本。

## 1.9.2 Access数据库

前面曾经介绍过,数据库(database)就是数据存储的仓库,是针对特定的需求所整理和组织出的相关信息的汇集处。例如,全国居民的身份证信息、某银行的客户账户信息、12306网上订票系统的订单数据、医院患者的看病记录等。本节介绍Access数据库,让读者了解一下Access数据库的结构和Access数据库文件。

### 1. Access数据库结构

Access是关系数据库。Access数据库在精心定义的结构中存储信息。Access表可存储各种不同类型的数据,从简单的文本到诸如图片、音频或视频等复杂数据。

在Access数据库中,任何事物都可以被称为对象,也就是说,Access数据库由各种对象组成,包括表、查询、窗体、报表、宏和模块等。其中,可以利用表对象来存储信息,查询、窗体和报表提供对数据的访问,允许用户添加或提取数据,以及通过有用的方式呈现数据,利用宏对象完成自动化工作,利用模块实现复杂功能。

此外,Access数据库具备存储、组织和管理各项相关信息的功能。数据库记录了字段和记录的验证规则、各个字段的标题和说明、各个字段的默认值、各个表的索引、各个表之间的关联性、数据参照完整性等。

在Access中,数据库是数据以及相关对象的整体容器。它不仅是表的集合,而且包含很多类型的对象。当打开Access数据库时,数据库中的对象(表、查询等)会呈现出来,供用户处理使用。用户可以根据需要同时打开多个Access副本,也可以同时处理多个数据库。

### 2. Access数据库文件

由于Access数据库与传统的数据库概念有所不同,它采用特有的全环绕数据库文件结构组成数据库文件,因此,它可以以一个单独的数据库文件存储一个数据库应用系统中包含的所有对象。基于Access数据库文件的这一特点,创建一个Access数据库应用系统的过程就是创建一个Access数据库文件并在其中设置和创建各种对象的过程。

#### 知识点:

在Access中,不同版本数据库文件的扩展名也不一样,早期的Access版本数据库文件的扩展名为.mdb,从Access 2007开始,其扩展名为.accdb。

开发一个Access 2021数据库应用系统的流程:第一步工作就是创建一个Access数据库文件,其操作的结果是在磁盘上建立一个扩展名为.accdb的数据库文件;第二步工作则是在数据库中创建数据表,并建立数据表之间的关系;接着,创建其他对象,最终即可形成完备的Access 2021数据库应用系统。

整个数据库应用系统仅以一个文件存储于文件系统中,显得极为简洁,这也使得该数据库应用系统的创建和发布变得非常简单。这也是很多小型数据库应用系统开发者偏爱Access的原因之一。实际上,对于Access数据库管理系统来说,数据库是一级容器对象,其他对象均置于该容器对象之中,因此,数据库是其他对象的基础,即其他对象必须建立在数据库中。

### 1.9.3 Access数据库对象

数据库对象是Access最基本的容器对象，它是一些关于某个特定主题或目的的信息集合，具有管理本数据库中所有信息的功能。在数据库对象中，用户可以将自己的数据分别保存在彼此独立的存储空间中，这些空间称为数据表；可以使用联机窗体来查看、添加和更新数据表中的数据；可以使用查询来查找并检索所需的数据；也可以使用报表以特定的版面布局来分析及打印数据。总之，创建一个数据库对象是应用Access建立信息系统的第一步工作。

早期的Access中有7种不同类别的数据库对象，即表、查询、窗体、报表、数据访问页、宏和模块。从Access 2010开始，不再支持数据访问页对象。如果希望在Web上部署数据输入窗体并在Access中存储所生成的数据，则需要将数据库部署到Microsoft Windows SharePoint Services 3.0服务器上，使用Windows SharePoint Services所提供的工具实现所需的目标。

不同的对象在数据库中有着不同的作用：表是数据库的核心与基础，存放着数据库中的全部数据；报表、查询和窗体都是从数据库中获得数据信息，以实现用户的某一特定的需求，如查找、计算统计、打印、编辑修改等；窗体可以提供一种良好的用户操作界面，通过它可以直接或间接地调用宏或模块，并执行查询、打印、预览和计算等功能，甚至可以对数据库进行编辑修改操作。

#### 1. 表

表是数据库中用来存储数据的对象，是整个数据库系统的基础。建立和规划数据库，首先要做的就是建立各种数据表。数据表是数据库中存储数据的唯一单位，各种信息分门别类地存放在各种数据表中。Access允许一个数据库中包含多个表，可以在不同的表中存储不同类型的数据。通过在表之间建立关系，可以将不同表中的数据联系起来，以供使用。

表中的数据以行和列的形式保存，类似于Excel电子表格。表中的列称为字段，字段是Access信息的最基本载体，说明了一条信息在某一方面的属性。表中的每一行称为记录，记录是由一个或多个字段组成的。一条记录就是一个完整的信息。

在数据库中，应该为每个不同的主题建立不同的表，这样不但可以提高创建数据库的工作效率，还可以减少数据输入产生的错误。

#### 2. 查询

查询是数据库中应用得最多的对象之一。它可执行很多不同的功能，最常用的功能是从表中检索符合某种条件的数据。查询是实现数据库设计目的的体现，数据库创建完成后，数据只有被使用者查询使用才能真正体现它的价值。

查询是用来操作数据库中的数据记录的手段，利用它可以按照一定的条件或准则从一个或多个表中筛选出需要的字段，并将它们集中起来，形成动态数据集，这个动态数据集就是用户想看到的来自一个或多个表中的字段，它显示在一个虚拟的数据表窗口中。用户可以浏览、查询、打印，甚至修改这个动态数据集中的数据，Access会自动将所做的任何修改更新到对应的表中。执行某个查询后，用户可以对查询的结果进行编辑或分析，并将查询结果作为其他对象的数据源。

查询到的数据记录集合称为查询的结果集。结果集以二维表形式显示出来，但它们不是基本表。每个查询只记录该查询的查询操作方式，这样，每进行一次查询操作，其结果集显示的都是基本表中当前存储的实际数据，它反映的是查询的那个时刻数据表的情况，查询的结果是静态的。

查询对象的运行形式与数据表对象的运行形式几乎完全相同，但它只是数据表对象所包含数据的某种抽取与显示，本身并不包含任何数据。需要注意的是，查询对象必须建立在数据表对象之上。

### 3. 窗体

窗体是Access数据库对象中最灵活的一种对象，其数据源可以是表或查询。窗体有时被称为“数据输入屏幕”。窗体是用来处理数据的界面，通常包含一些可执行各种命令的按钮。可以说窗体是数据库与用户进行交互操作的最好界面。利用窗体，用户能够从表中查询、提取所需的数据，并将其显示出来。通过在窗体中插入宏，用户可以把Access的各个对象很方便地联系起来。

窗体的类型比较多，大致可以分为如下3类。

- 提示型窗体：主要用于显示文字和图片等信息，没有实际性的数据，也基本没有什么功能，主要用于作为数据库应用系统的主界面。
- 控制型窗体：使用该类型的窗体，可以在窗体中设置相应菜单和一些命令按钮，用于完成各种控制功能的转移。
- 数据型窗体：使用该类型的窗体，可以实现用户对数据库中相关数据进行操作的界面，这是Access数据库应用系统中使用得最多的窗体类型。

### 4. 报表

数据库应用程序通常要打印输出数据，在Access中，如果要对数据库中的数据进行打印，使用报表是最简单且有效的方法。利用报表可以将数据库中需要的数据提取出来进行分析、整理和计算，并将数据以格式化的方式发送到打印机。

报表以类似于PDF的格式显示数据。Access在创建报表时提供了额外的灵活性。例如，可以配置报表以便列出给定表(如Customers表)中的所有记录，也可以使报表仅包含满足特定条件的记录(如居住在“河北沧州”的所有客户)。为此，可以基于查询创建报表，该查询仅选择报表所需的记录。

用户可以在一个表或查询的基础上创建报表，也可以在多个表或查询的基础上创建报表。利用报表可以创建计算字段；还可以对记录进行分组，以便计算出各组数据的汇总等。在报表中，可以控制显示的字段、每个对象的大小和显示方式，还可以按照所需的方式显示相应的内容。

### 5. 宏

Access的宏对象是Access数据库中的一个基本对象。宏是指一个或多个操作的集合，其中每个操作实现特定的功能，如打开某个窗体或打印某个报表。某些普通的、需要多个指令连续执行的任务通过一条指令就能自动完成，而这条指令就称为宏。例如，可创建某个宏，在用户单击某个命令按钮时运行该宏，打印某个报表。因此，宏可以看作一种简化的编程语言。利用宏，用户不必编写任何代码，就可以实现一定的交互功能。

通过宏，可以实现的功能主要有以下几项。

- 打开或关闭数据表、窗体，打印报表和执行查询。
- 弹出提示信息框，显示警告。
- 实现数据的输入和输出。
- 在数据库启动时执行操作等。
- 查找数据。

Microsoft Office系列产品提供的所有工具中都提供了宏的功能。利用宏可以简化操作，使大量重复性操作自动完成，从而使管理和维护Access数据库更加简单。

宏可以是包含一个操作序列的宏，也可以是若干个宏的集合所组成的宏组。一个宏或宏组的执行与否还可以使用一个条件表达式是否成立予以判断，即可以通过给定的条件来决定在哪些情况下运行宏。

## 6. 模块

模块对象是Access数据库中的一个基本对象。在Access中，不仅可以通过从宏列表中以选择的方式创建宏，还可以利用VBA(Visual Basic for Applications)编程语言编写过程模块。

模块是将VBA的声明、语句和过程作为一个单元进行保存的集合，也就是程序的集合。创建模块对象的过程也就是使用VBA编写程序的过程。Access中的模块可以分为类模块和标准模块两类。类模块中包含各种事件过程，标准模块包含与任何其他特定对象无关的常规过程。

尽管Microsoft在推出Access产品之初就将该产品定位为不用编程的数据库管理系统，而实际上，要在Access的基础上进行二次开发来实现一个数据库应用系统，用VBA编写适当的程序是必不可少的。也就是说，若需要开发一个Access数据库应用系统，其间必然包括VBA模块对象。

## 1.10 本章小结

随着信息技术的飞速发展，需要处理的数据越来越多，将越来越多的资料存入计算机中，并通过一些编制好的计算机程序对这些资料进行管理，这些程序后来就被称为“数据库管理系统”(DBMS)，它们可以帮助管理输入在计算机中的大量数据。本章主要介绍了数据库的基本概念、数据模型、数据库管理系统、关系数据库和关系代数，以及数据库设计的内容和一般步骤等数据库基础理论知识。Access是基于关系模型的数据库管理系统。本章的知识虽然过于理论化，但掌握这些理论是学好Access的重要基础。

## 1.11 思考和练习

1. 什么是数据库？什么是数据库系统？
2. 什么是数据库管理系统？它主要有哪些功能？

3. 说出几种常用的数据模型。
4. 下列选项中, 不属于数据库系统特点的是( )。
- A. 数据共享      B. 数据完整性      C. 数据冗余度高      D. 数据独立性高
5. 数据库系统的核心是( )。
- A. 数据库      B. 数据库管理系统      C. 数据模型      D. 软件工具
6. 数据库(DB)、数据库系统(DBS)和数据库管理系统(DBMS)三者之间的关系是( )。
- A. DBS包括DB和DBMS      B. DBMS包括DB和DBS  
C. DB包括DBS和DBMS      D. DBS就是DB, 也就是DBMS
7. DBMS是指数据库管理系统, 它是位于\_\_\_\_\_和\_\_\_\_\_之间的一个数据管理软件。
8. 什么是关系模型? 它是如何表示实体和实体之间的联系?
9. 常用的关系运算有哪些? 如何区分一目运算和二目运算?
10. 关系运算中花费时间可能最长的运算是( )。
- A. 投影      B. 选择      C. 笛卡儿积      D. 除
11. 自然连接是构成新关系的有效方法。一般情况下, 当对关系R和S使用自然连接时, 要求R和S含有一个或多个共有的( )。
- A. 元组      B. 行      C. 记录      D. 属性
12. 为什么要进行关系模式规范化?
13. 第三范式与第二范式相比有哪些改进?
14. 什么是数据操纵语言? 它有什么作用?
15. 简述数据库设计的步骤。
16. 企业进销存管理系统主要实现从进货、库存到销售的一体化信息管理, 涉及商品信息、商品的供应商、购买商品的客户等多个实体。根据下面的描述创建客户实体E-R图、供应商实体E-R图、商品实体E-R图。
- 企业进销存管理系统将记录所有的客户信息, 在销售、退货等操作时, 将直接引用该客户的实体属性。客户实体包括客户编号、客户名称、简称、地址、电话、邮政编码、联系人、联系人电话、传真、开户行和账号等属性。
  - 不同的供应商可以为企业提供不同的商品, 在商品信息中将引用商品供应商的实体属性。供应商实体包括编号、名称、简称、地址、电话、邮政编码、传真、联系人、联系电话、开户行和E-mail属性。
  - 商品信息是进销存管理系统中的基本信息, 系统将维护商品的进货、退货、销售、入库等操作。商品实体包括编号、商品名称、商品简称、产地、单位、规格、包装、批号、批准文号、商品简介和供应商属性。
17. 请简述满足1NF、2NF和3NF的基本条件, 并完成以下题目:  
某信息一览表如图1-22所示, 其是否满足3NF? 若不满足, 请将其转化为符合3NF的关系。

考生编号	姓名	性别	考生学校	考场号	考场地 点	成绩	
						考试成绩	学分

图1-22 信息