

普通高等教育“十三五”规划教材



# 数据库系统 原理与实践

SHUJUKU XITONG YUANLI YU SHIJIAN

陈红顺 黄秋颖 周鹏 主编

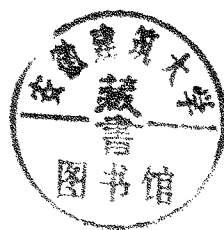
中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

普通高等教育“十三五”规划教材

# 数据库系统原理与实践

陈红顺 黄秋颖 周 鹏 主 编

刘 军 副主编



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

## 内 容 简 介

本书全面介绍了数据库系统概论、数据模型、关系数据库和关系数据库标准语言 SQL、数据库安全性、数据库完整性、关系规范化理论、数据库设计、Transact-SQL 编程、关系查询处理和查询优化、并发控制、数据库恢复技术、数据库编程接口、ORM 技术及应用系统开发实例。全书理论与实践并重,提供丰富的示例代码,并配有相应的思考题与实验。

本书内容丰富,系统性强,注重理论联系实际,适合作为高等本科院校(特别是应用型本科院校)计算机相关专业的数据库课程的教材,也可以作为相关技术人员的自学参考书。

### 图书在版编目(CIP)数据

数据库系统原理与实践 / 陈红顺, 黄秋颖, 周鹏主编. —北京:  
中国铁道出版社, 2018. 7  
普通高等教育“十三五”规划教材  
ISBN 978-7-113-23333-4

I. ①数… II. ①陈… ②黄… ③周… 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 (2017) 第 320645 号

书 名: 数据库系统原理与实践  
作 者: 陈红顺 黄秋颖 周 鹏 主编

---

策 划: 韩从付 读者热线: (010) 63550836  
责任编辑: 刘丽丽 冯彩茹  
封面设计: 付 巍  
封面制作: 刘 颖  
责任校对: 张玉华  
责任印制: 郭向伟

---

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)  
网 址: <http://www.tdpress.com/51eds/>  
印 刷: 北京柏力行彩印有限公司  
版 次: 2018 年 7 月第 1 版 2018 年 7 月第 1 次印刷  
开 本: 787 mm×1 092 mm 1/16 印张: 15.5 字数: 387 千  
书 号: ISBN 978-7-113-23333-4  
定 价: 45.00 元

---

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 51873659

# 前 言

## PREFACE

数据库技术产生于 20 世纪 60 年代末,主要研究如何存储、使用和管理数据。经过近半个世纪的发展,已经形成了完整的理论和技术体系,并已成为计算机科学的一个重要分支。

全书共分为 15 章,主要包括数据库系统概论、数据模型、关系数据库、关系数据库标准语言 SQL、数据库安全性、数据库完整性、关系规范化理论、数据库设计、Transact-SQL 编程、关系查询处理和查询优化、并发控制、数据库恢复技术、数据库编程接口、ORM 技术以及应用系统开发实例等。内容基本覆盖了关系数据库系统的原理、设计及开发应用技术。

全书注重理论联系实际,希望能使读者在掌握数据库系统基本原理和数据库设计方法的同时,进一步掌握数据库应用系统的开发方法。书中提供了丰富的示例和代码,所有代码均在 Microsoft SQL Server 2014 上调试通过,便于读者自学和练习。

本书由陈红顺、黄秋颖、周鹏任主编,刘军任副主编。具体编写分工如下:第 1 章、第 5 章、第 8 章、第 9 章和第 13 章由陈红顺编写,第 4 章、第 11 章、第 14 章和第 15 章由黄秋颖编写,第 2 章、第 3 章、第 7 章和第 12 章由周鹏编写,第 6 章、第 10 章由刘军编写。全书由陈红顺负责统稿。

本书是北京师范大学珠海分校校级质量工程项目精品教材的建设成果,同时得到了北京师范大学珠海分校信息技术学院各位领导的大力支持,在此一并表示感谢。

由于编者水平有限,加之时间仓促,书中难免存在疏漏和不足之处,敬请读者批评指正。

编 者

2018 年 4 月于珠海

# 目 录

## CONTENTS

### 第 1 章 数据库系统概论 ..... 1

1.1 基本概念 .....	1
1.1.1 数据 .....	1
1.1.2 数据库 .....	1
1.1.3 数据库管理系统 .....	2
1.1.4 数据库系统 .....	2
1.2 数据管理技术的产生和发展 .....	3
1.2.1 人工管理阶段 .....	3
1.2.2 文件系统阶段 .....	3
1.2.3 数据库系统阶段 .....	4
1.3 数据库系统结构 .....	6
1.3.1 三级模式结构 .....	6
1.3.2 两级映像和数据独立性 .....	7
1.4 数据库系统的组成 .....	8
本章小结 .....	9
练习题 .....	9
实验 1 熟悉 SQL Server .....	10

### 第 2 章 数据模型 ..... 11

2.1 数据模型概述 .....	11
2.2 数据模型的组成要素 .....	12
2.2.1 数据结构 .....	12
2.2.2 数据操作 .....	12
2.2.3 完整性约束条件 .....	12
2.3 概念模型 .....	12
2.3.1 信息世界的基本概念 .....	12
2.3.2 实体之间的联系 .....	13
2.3.3 概念模型的表示方法 .....	13
2.4 逻辑模型 .....	14
2.4.1 层次模型 .....	14
2.4.2 网状模型 .....	15
2.4.3 关系数据模型 .....	16
2.4.4 面向对象模型 .....	17
2.5 物理模型 .....	18

本章小结 .....	18
练习题 .....	19
实验 2 使用 SQL Server Management Studio .....	19

### 第 3 章 关系数据库 ..... 21

3.1 关系数据结构及形式化定义 .....	21
3.1.1 关系 .....	21
3.1.2 关系模式 .....	24
3.1.3 关系数据库 .....	24
3.2 关系操作 .....	25
3.2.1 基本的关系操作 .....	25
3.2.2 关系数据语言的分类 .....	25
3.3 关系模型的完整性 .....	26
3.3.1 实体完整性 .....	26
3.3.2 参照完整性 .....	26
3.3.3 用户定义完整性 .....	26
3.4 关系代数 .....	27
3.4.1 传统的集合运算 .....	27
3.4.2 专门的关系运算 .....	28
3.4.3 关系代数表达式 .....	33
本章小结 .....	33
练习题 .....	33
实验 3 关系代数表达式 .....	34

### 第 4 章 关系数据库标准 语言 SQL ..... 36

4.1 SQL 概述 .....	36
4.1.1 SQL 的产生与发展 .....	36
4.1.2 SQL 的特点 .....	37
4.1.3 SQL 运行环境 .....	37
4.2 网上书城示例数据库 .....	38
4.3 数据定义 .....	41
4.3.1 数据库的定义与删除 .....	41

4.3.2	基本表的定义、删除 与修改 .....	42	5.3	角色和权限 .....	69
4.3.3	模式的定义与删除 .....	44	5.3.1	权限 .....	69
4.4	数据更新 .....	45	5.3.2	授权与回收 .....	69
4.4.1	插入数据 .....	45	5.4	视图机制 .....	71
4.4.2	修改数据 .....	45	5.5	数据加密 .....	71
4.4.3	删除数据 .....	45	5.5.1	替换方法 .....	71
4.5	数据查询 .....	46	5.5.2	置换方法 .....	72
4.5.1	SELECT 语句的结构 .....	46	5.6	审计 .....	72
4.5.2	单表查询 .....	46	5.7	SQL Server 安全管理 .....	73
4.5.3	多表连接查询 .....	51	5.7.1	概述 .....	73
4.5.4	嵌套查询 .....	53	5.7.2	服务器身份验证 .....	74
4.5.5	集合查询 .....	55	5.7.3	数据库用户 .....	76
4.5.6	基于派生表的查询 .....	56	5.7.4	角色 .....	77
4.6	带子查询的数据更新 .....	56	本章小结 .....	78	
4.6.1	带有子查询的数据插入 .....	56	练习题 .....	79	
4.6.2	带子查询的数据修改 .....	57	实验 7 数据库安全性 .....	79	
4.6.3	带子查询的数据删除 .....	57	<b>第 6 章 数据库完整性 .....</b>	<b>80</b>	
4.7	索引 .....	57	6.1	数据库完整性概述 .....	80
4.7.1	索引类型 .....	58	6.2	实体完整性 .....	81
4.7.2	建立索引 .....	58	6.2.1	定义实体完整性 .....	81
4.7.3	删除索引 .....	58	6.2.2	实体完整性检查和 违约处理 .....	81
4.8	视图 .....	58	6.3	参照完整性 .....	81
4.8.1	定义视图 .....	59	6.3.1	定义参照完整性 .....	81
4.8.2	查询视图 .....	60	6.3.2	参照完整性检查和 违约处理 .....	82
4.8.3	更新视图 .....	60	6.4	用户定义完整性 .....	83
4.8.4	视图的作用 .....	61	6.4.1	非空约束 .....	84
本章小结 .....	61		6.4.2	唯一值约束 .....	84
练习题 .....	62		6.4.3	默认值约束 .....	84
实验 4 创建数据库 .....	62		6.4.4	CHECK 约束 .....	85
实验 5 数据查询 .....	63		6.5	完整性约束命名子句 .....	85
实验 6 数据更新和视图 .....	64		6.6	触发器 .....	85
<b>第 5 章 数据库安全性 .....</b>	<b>66</b>		6.6.1	触发器的基本概念 .....	85
5.1	安全性概述 .....	66	6.6.2	创建触发器 .....	86
5.1.1	安全性定义 .....	66	6.6.3	触发器的管理 .....	87
5.1.2	安全性措施 .....	66	本章小结 .....	87	
5.1.3	安全标准简介 .....	67	练习题 .....	87	
5.2	用户管理 .....	68			

实验 8 数据库完整性 .....	88	8.4.1 E-R 模型向关系模型 的转换 .....	117
<b>第 7 章 关系规范化理论 .....</b>	<b>89</b>	8.4.2 数据模型的优化 .....	118
7.1 规范的必要性 .....	89	8.4.3 设计用户子模式 .....	118
7.1.1 存在的问题 .....	89	8.5 物理结构设计 .....	119
7.1.2 解决方法 .....	90	8.5.1 物理结构设计概述 .....	119
7.2 函数依赖 .....	91	8.5.2 存取方法选择 .....	119
7.2.1 函数依赖 .....	91	8.5.3 存储结构确定 .....	120
7.2.2 码 .....	92	8.6 实施、运行与维护 .....	121
7.2.3 函数依赖集的闭包 .....	93	8.6.1 数据库的实施 .....	121
7.2.4 函数依赖的推理规则 .....	93	8.6.2 数据库的维护 .....	122
7.2.5 属性集的闭包 .....	94	本章小结 .....	123
7.2.6 最小函数依赖集 .....	95	练习题 .....	123
7.2.7 候选键的求解方法 .....	97	实验 10 数据库设计 .....	123
7.3 关系的范式 .....	97	<b>第 9 章 Transact-SQL 编程... 124</b>	
7.3.1 关系的范式简介 .....	97	9.1 批处理 .....	124
7.3.2 关系的范式定义 .....	98	9.1.1 批处理使用规则 .....	124
7.4 多值依赖及第四范式 .....	101	9.1.2 批处理错误处理 .....	125
7.4.1 多值依赖 .....	102	9.1.3 批处理示例 .....	125
7.4.2 第四范式 .....	103	9.2 注释 .....	125
7.5 关系模式的分解 .....	103	9.3 常量与变量 .....	126
本章小结 .....	106	9.3.1 常量 .....	126
练习题 .....	106	9.3.2 变量 .....	127
实验 9 关系规范化 .....	107	9.4 流程控制语句 .....	129
<b>第 8 章 数据库设计 .....</b>	<b>108</b>	9.4.1 选择结构 .....	130
8.1 数据库设计概述 .....	108	9.4.2 循环结构 .....	131
8.1.1 数据库设计的任务 与特点 .....	108	9.4.3 其他流程控制语句 .....	132
8.1.2 数据库设计的方法 和步骤 .....	110	9.5 游标 .....	133
8.2 需求分析概述 .....	111	9.5.1 游标概述 .....	133
8.2.1 需求分析的任务与方法 .....	111	9.5.2 游标的基本操作 .....	133
8.2.2 需求分析 .....	112	9.5.3 游标使用实例 .....	135
8.2.3 需求分析的结果 .....	113	9.6 函数 .....	136
8.3 概念结构设计 .....	114	9.6.1 函数概述 .....	136
8.3.1 概念模型 .....	114	9.6.2 系统提供的内置函数 .....	136
8.3.2 概念模型设计 .....	114	9.6.3 用户函数的创建 .....	138
8.4 逻辑结构设计 .....	117	9.6.4 用户函数的使用与 管理 .....	140
		9.7 存储过程 .....	140

9.7.1 存储过程概述 .....	140	11.3.2 封锁协议 .....	162
9.7.2 存储过程的定义 .....	141	11.4 活锁和死锁 .....	164
9.7.3 存储过程的执行与管理 .....	142	11.4.1 活锁 .....	164
本章小结 .....	143	11.4.2 死锁 .....	165
练习题 .....	144	11.5 并发调度的可串行性 .....	167
实验 11 Transact-SQL 编程 .....	144	11.6 两段锁协议 .....	168
<b>第 10 章 关系查询处理和</b>		11.7 封锁的粒度 .....	169
<b>    查询优化 .....</b>	<b>145</b>	11.7.1 多粒度封锁 .....	170
10.1 关系数据库系统的查询处理 .....	145	11.7.2 意向锁 .....	170
10.1.1 查询处理步骤 .....	145	本章小结 .....	171
10.1.2 实现查询操作的 算法示例 .....	147	练习题 .....	172
10.2 关系数据库系统的查询优化 .....	149	实验 13 数据库并发控制 .....	172
10.2.1 查询优化概述 .....	149	<b>第 12 章 数据库恢复技术 .....</b>	<b>173</b>
10.2.2 一个实例 .....	150	12.1 故障种类 .....	173
10.3 代数优化 .....	151	12.2 恢复的实现技术 .....	174
10.3.1 关系代数表达式等价 变换规则 .....	152	12.2.1 数据转储 .....	175
10.3.2 查询树的启发式优化 .....	153	12.2.2 登记日志文件 .....	175
10.4 物理优化 .....	154	12.3 恢复策略 .....	177
10.4.1 基于启发式规则的 存取路径选择优化 .....	154	12.3.1 事务故障的恢复 .....	177
10.4.2 基于代价估算的优化 .....	155	12.3.2 系统故障的恢复 .....	177
本章小结 .....	156	12.3.3 介质故障的恢复 .....	178
练习题 .....	157	12.4 具有检查点的恢复技术 .....	178
实验 12 查询处理和查询优化 .....	157	12.5 SQL Server 备份与恢复 .....	179
<b>第 11 章 并发控制 .....</b>	<b>159</b>	12.5.1 备份的基本概念 .....	179
11.1 事务 .....	159	12.5.2 备份操作和备份命令 .....	181
11.1.1 事务的概念 .....	159	12.5.3 恢复操作和恢复命令 .....	183
11.1.2 事务的性质 .....	159	本章小结 .....	185
11.2 并发控制概述 .....	160	练习题 .....	185
11.2.1 丢失修改 .....	161	实验 14 数据库备份和恢复 .....	185
11.2.2 不可重复读 .....	161	<b>第 13 章 数据库编程接口 .....</b>	<b>186</b>
11.2.3 读“脏”数据 .....	161	13.1 ODBC 编程 .....	186
11.3 封锁及封锁协议 .....	162	13.1.1 ODBC 概述 .....	186
11.3.1 基本锁类型 .....	162	13.1.2 ODBC 编程接口 .....	188
		13.1.3 ODBC 开发实例 .....	191
		13.2 JDBC 编程 .....	195
		13.2.1 JDBC 概述 .....	195
		13.2.2 JDBC 主要接口 .....	198



13.2.3 JDBC 开发实例 .....	202	练习题 .....	223
13.3 ADO.NET 编程 .....	203	实验 16 使用 ORM 框架 .....	223
13.3.1 ADO.NET 概述 .....	203	<b>第 15 章 应用系统开发实例 ....</b>	<b>224</b>
13.3.2 ADO.NET 对象模型 .....	204	15.1 系统简介 .....	224
13.3.3 ADO.NET 开发实例 .....	206	15.2 用例模型分析 .....	224
本章小结 .....	208	15.2.1 用户分析 .....	224
练习题 .....	208	15.2.2 用例分析 .....	225
实验 15 连接数据库 .....	208	15.3 数据库设计 .....	225
<b>第 14 章 ORM 技术 .....</b>	<b>209</b>	15.4 系统设计 .....	228
14.1 ORM 技术概述 .....	209	15.4.1 系统架构设计 .....	228
14.2 Hibernate 简介 .....	210	15.4.2 功能模块设计 .....	229
14.2.1 Hibernate 的对象关系		15.4.3 系统详细设计 .....	229
映射机制 .....	210	15.5 系统实现 .....	232
14.2.2 Hibernate 的主要组件 .....	211	15.5.1 系统登录及主界面 .....	232
14.2.3 Hibernate 简单例子 .....	212	15.5.2 专业信息管理 .....	234
14.3 Entity Framework 简介 .....	216	15.5.3 课程信息管理 .....	234
14.3.1 Entity Framework		15.5.4 班级信息管理 .....	235
实现原理 .....	217	15.5.5 教师信息管理 .....	235
14.3.2 Entity Framework		15.5.6 学生信息管理 .....	236
主要组件 .....	217	15.5.7 教学班级管理 .....	236
14.3.3 DbContext 和 Entity 类 .....	218	15.5.8 学生成绩管理 .....	237
14.3.4 Entity Framework		本章小结 .....	238
简单例子 .....	218	练习题 .....	238
本章小结 .....	222	实验 17 应用系统开发 .....	238

# 第 1 章

## 数据库系统概论

数据库是数据管理的最新技术，是计算机科学的重要分支。数据库技术是信息系统的核心和基础，广泛应用在各行各业。

本章主要介绍数据库系统的基本概念、数据管理技术的产生和发展历史、数据库系统结构和数据库系统的组成。

### 1.1 基本概念

数据、数据库、数据库管理系统和数据库系统是与数据库技术密切相关的 4 个基本概念。

#### 1.1.1 数据

数据库中存储的基本对象是数据 (Data)。一提到数据，人们在头脑中的直觉反应就是数字。其实，数字也仅仅是数据的一种表现形式。从计算机的角度来看，数据是指能够被计算机存储和处理的符号。实际上，数据的表现形式多种多样，不仅有数字、文字，还可以是图形、图像和声音等。

通常，与数据紧密相关的、但很容易混淆的概念是信息 (Information)。一般认为，数据是信息的载体；而信息需要通过特定的符号来表达，也就是说，数据是信息的具体表现形式。例如，某个学生的成绩，可以用数字 98 (假设是百分制) 来表达，也可以 A (假设是等级制，共有 A、B、C、D 和 E 5 个等级) 来表达，98 (或 A) 就是学生成绩的符号载体。

#### 1.1.2 数据库

数据库 (Database, DB)，顾名思义，就是存放数据的仓库，只是这个仓库存储在计算机的存储设备上。

通常，数据库还满足以下特点：① 数据库中的数据是按照一定格式组织起来存放的，通常是按照某种数据模型组织起来的，如建立在关系模型之上的数据库称为关系型数据库；

② 数据库中的数据通常是长期存储的，以方便用户查询；③ 数据库中的数据可以为多个用户和应用系统共享使用；④ 数据库还具有冗余度低、独立性高的特点。

总结起来，在计算机科学中，数据库是指长期存储在计算机内的、有组织的、可共享的数据集合。

### 1.1.3 数据库管理系统

数据库管理系统 (Database Management System, DBMS) 是专门用来管理数据库的计算机软件，以实现数据库的统一管理和控制。数据库管理系统是计算机系统的重要基础软件，在计算机系统中的地位非常重要，如图 1-1 所示，数据库管理系统位于应用程序和操作系统之间。通常，数据库管理系统支持一种或几种数据模型，如支持关系模型的数据管理系统称为关系型数据库管理系统 (Relational Database Management System, RDBMS)。常见的关系数据库管理系统包括甲骨文公司的 Oracle、赛贝斯公司的 Sybase、IBM 公司的 Informix 和 DB2、微软公司的 SQL Server 和 Access 及开源的 MySQL 和 PostgreSQL 等。

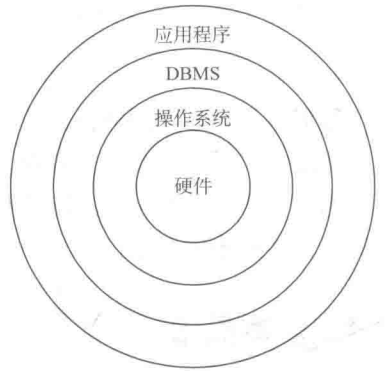


图 1-1 数据库管理系统在计算机系统的位置

数据库管理系统是一个非常复杂的大型软件，为应用程序提供了访问数据库的各种接口，主要包括数据定义、数据操作、数据控制、事务管理和其他功能。

(1) 数据定义功能。数据定义是指定义数据库中的各种对象，如表、视图、存储过程等。这些功能一般通过数据库管理系统提供的数据库定义语言 (Data Definition Language, DDL) 来实现。

(2) 数据操纵功能。数据操纵是指对数据库中数据进行查询、插入、删除和更新操作，这些操作一般通过数据库管理系统的数据操纵语言 (Data Manipulation Language, DML) 来实现。

(3) 数据控制功能。数据控制是指控制数据库用户对数据的访问权限，不同用户可以给予不同的权限，以保障数据库的安全。数据控制功能一般通过数据库控制语言 (Data Control Language, DCL) 来实现。

(4) 事务管理功能。事务管理功能保证数据库中的数据可以供多个用户并发使用而不会产生相互干扰，也能在数据库发生故障时进行正确的恢复。

(5) 其他功能。主要包括数据存储、数据转储与重组、网络通信、数据传输、系统性能监视与调整等。

### 1.1.4 数据库系统

数据库系统 (Database System, DBS) 是指引入数据库技术后的计算机系统。一般由数据库、数据库管理系统 (及相关实用工具)、应用系统和数据库管理员构成。为了保证数据库正常、高效地运行，除了数据库管理系统外，还需要专门人员对数据库进行维护，这些专门人员称为数据库管理员 (Database Administrator, DBA)。

## 1.2 数据管理技术的产生和发展

数据管理是指对数据进行分类、组织、编码、存储、检索和维护等，它是数据处理的核心问题。而数据处理是指对各种数据进行收集、存储、加工和传播等一系列活动的总和。

数据库技术正是应数据管理任务的需要而产生的。在应用需求的推动下，在计算机硬件、软件发展的基础上，数据管理技术经历了人工管理、文件系统和数据库系统3个阶段。

### 1.2.1 人工管理阶段

1946年，世界上第一台电子计算机ENIAC (Electronic Numerical Integrator And Computer) 在美国宾夕法尼亚大学诞生。从那时起，一直到20世纪50年代中期，计算机主要用于科学计算。当时的计算机，外部存储器只有打孔卡片、纸带和磁带，这些存储介质只支持数据的顺序访问，不支持随机存取；软件方面则没有操作系统，没有管理数据的专门软件；数据处理的方式是批处理。因此，每个应用程序需要根据自己的实际需要，定义数据的物理结构、逻辑结构和存取方法。

人工管理数据具有以下特点：

(1) 数据不长期保存。由于当时的计算机主要用于科学计算，一般不需要将数据长期保存起来，只是在进行某一科学计算任务时才输入数据，任务完成后不再保存。

(2) 应用程序管理数据。数据需要由应用程序自己设计、定义和管理，没有专门的软件负责数据的管理工作。应用程序不仅要规定数据的逻辑结构，还要设计对应的物理结构（包括存储结构、存取方法和输入/输出方式等）。

(3) 数据不共享。由于数据是面向应用程序的，因此，一组数据只能对应一个应用程序。当多个应用程序需要相同的数据时，各应用程序必须各自定义数据的逻辑结构和物理结构。因此，程序与程序之间有大量的冗余数据。

(4) 数据不具有独立性。一旦数据的逻辑结构或物理结构发生变化，应用程序则必须做相应修改，这无疑加重了程序员的负担。

在人工管理阶段，程序与数据之间的对应关系可用图1-2来表示。

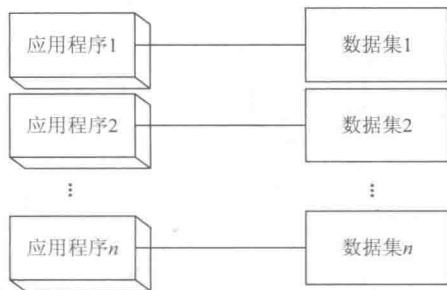


图 1-2 人工管理阶段应用程序和数据之间的对应关系

### 1.2.2 文件系统阶段

1956年，IBM公司向世界展示了第一台磁盘存储系统IBM 350 RAMAC，虽然它的体积相当于两个冰箱的体积，而存储容量只有5 MB，但它却是现代硬盘的雏形。与磁带等顺序

存储介质不同，磁盘上的磁头可以直接移动到盘片上的任何位置，从而成功实现对数据的随机存取。磁盘技术的突破变革了计算机数据管理的方式。在软件方面，操作系统中已经有了专门的数据管理软件，一般称为文件系统。在数据处理方式上，除了批处理，还可以联机实时处理。20世纪50年代后期到60年代中期，应用程序主要依靠文件系统来管理数据。

使用文件系统管理数据具有以下特点：

(1) 数据可以长期保存。由于计算机大量用于数据处理，数据需要长期保存在外部存储设备上以便反复进行查询、修改、插入和删除等操作。

(2) 由文件系统管理数据。由专门的软件（即文件系统）进行数据管理，文件系统把数据组织成相互独立的数据文件，利用“按文件名访问、按记录进行存取”的管理技术，可以对文件进行修改、插入和删除等操作。文件系统实现了记录内部的结构化，但整体上无结构。应用程序和数据之间由文件系统提供存取方法进行转换，使应用程序和数据之间有了一定的独立性，程序员不必过多地考虑物理细节，可将精力集中于算法。还有，数据在存储上的改变，不一定需要反映在程序上，从一定程度上节省了维护程序的工作量。

但是，文件系统仍然存在以下缺点：

(1) 数据共享性差，冗余度大。在文件系统中，一个（或一组）文件基本上对应于一个应用程序。当不同的应用程序需要具有相同部分的数据时，也必须建立各自的文件，而不能共享相同的数据，因此数据的冗余度大，浪费存储空间。同时，由于相同的数据重复存储、各自管理，容易造成数据的不一致性，给数据的修改和维护带来了困难。

(2) 数据独立性差。文件系统文件是为某一特定应用服务的，文件的逻辑结构对该应用程序来说是优化的，因此要想对现有的数据再增加一些新的应用会很困难，系统不容易扩充。一旦数据的逻辑结构改变，必须修改应用程序，修改文件结构的定义。应用程序的改变（如应用程序改用不同的高级语言编写）也会引起文件数据结构的变化，因此，数据与程序之间仍然缺乏独立性。

在文件系统阶段，应用程序与数据之间的关系如图1-3所示。

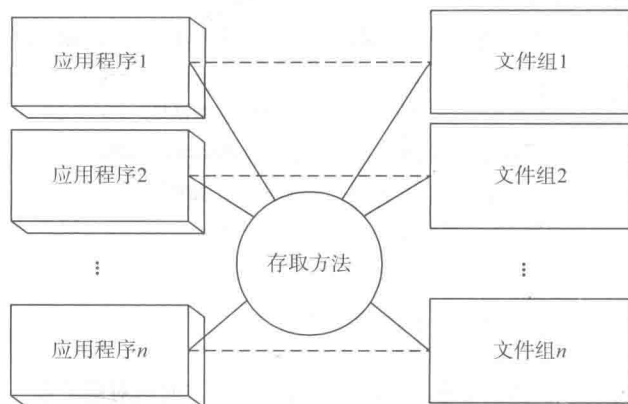


图1-3 文件系统阶段应用程序与数据之间的对应关系

### 1.2.3 数据库系统阶段

20世纪60年代，数据管理领域发生的三件大事标志着数据库系统的诞生。

### 1. IMS 系统

IBM 公司从 1966 年开始为“阿波罗”登月计划设计 IMS (Information Management System) 系统。IMS 面临的挑战是如何有效地存储和管理“土星 5 号”运载火箭和“阿波罗”太空飞船所产生的大量数据。1968 年 8 月 14 日,IMS 的首个版本在 IBM 2740 终端上正式发布。IMS 系统的最大贡献是提出了层次数据模型。

### 2. DBTG 系统

CODASYL (Conference on Data Systems Languages, 数据系统语言研究会) 是一个由数据处理领域专家组成的行业协会,成立于 1959 年。该组织开发了著名的 COBOL 语言。为了给 COBOL 增加数据处理能力,CODASYL 下属的数据库任务小组 (Database Task Group) 于 1969 年 10 月发布的一份技术报告,首次提出了网状数据模型,并对其进行了严格的定义。这就是后来人们所说的 DBTG 报告。

### 3. Codd 发表论文

1970 年,IBM 公司 San Jose 实验室的研究员 E. F. Codd 发表了著名的论文 *A Relational Model of Data for Large Shared Data Banks*。Codd 在该论文中首次提出关系数据模型,为关系数据库在整个 20 世纪 80 年代的迅猛发展奠定了坚实的理论基础。

20 世纪 60 年代后期以来,计算机硬件方面,大容量硬盘已经出现,且硬盘价格下降;软件则价格上升,为编写和维护系统软件及应用程序所需的成本相对增加;在处理方式上,联机实时处理要求更多,并开始提出和考虑分布式处理。在这种背景下,以文件系统作为数据管理手段已经不能满足应用的需求,于是为解决多用户、多应用共享数据的需求,使数据为尽可能多的应用服务,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——数据库管理系统。

从文件系统到数据库系统,标志着数据管理技术的飞跃。与人工管理和文件系统相比,数据库系统具有以下几方面的特点:

#### 1. 数据结构化

数据库系统实现了整体数据的结构化,这是数据库的主要特征之一,也是数据库系统与文件系统的本质区别。

在文件系统中,每个文件内部是有结构的,即文件由记录构成,每个记录由若干属性组成,但文件与文件之间是毫无结构化的。在数据库系统中,数据的最小存储单位是数据项,使得数据的管理更加灵活;数据的结构用数据模型描述,无须程序定义和解释,把数据和程序独立分开。这样可以保证数据不再仅仅针对某一个应用,而是面向全组织;不仅数据内部结构化,整体也是结构化的,数据之间具有联系,从而实现了数据的真正结构化。

#### 2. 数据的共享性高、冗余度低,易扩充

数据库系统从整体角度看待和描述数据,数据面向整个系统,可以被多个用户、多个应用共享使用。数据共享有利于减少数据冗余,节约存储空间;可以避免数据之间的不相容性与不一致性,且使系统易于扩充。

#### 3. 数据独立性高

数据独立性是一个重要的概念,包括数据的物理独立性和数据的逻辑独立性。

物理独立性是指用户的应用程序与存储在磁盘上的数据库中数据是相互独立的。当数据

的物理存储改变时,应用程序不用改变。

逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的。数据的逻辑结构改变了,用户程序可以不变。

数据独立性是由 DBMS 的二级映像功能来保证的。

#### 4. 数据由 DBMS 统一管理和控制

数据库系统关于数据控制的功能几乎都由 DBMS 提供,主要包括:

- (1) 数据的安全性保护。保护数据,以防止不合法的使用造成数据的泄密和破坏。
- (2) 数据的完整性检查。将数据控制在有效范围内,或保证数据之间满足一定的关系。
- (3) 并发控制。对多用户的并发操作加以控制和协调,防止相互干扰而破坏数据的一致性。
- (4) 数据库恢复。将数据库从错误状态恢复到某一已知的正确状态。

数据库系统阶段应用程序与数据之间的对应关系如图 1-4 所示。

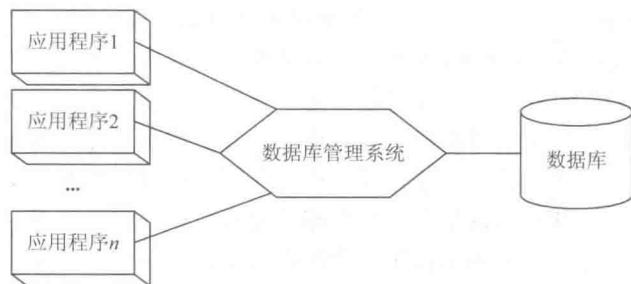


图 1-4 数据库系统阶段应用程序与数据之间的对应关系

## 1.3 数据库系统结构

数据库系统采用三级模式结构,三级模式之间形成了两级映像,从而实现了较高的数据独立性。

### 1.3.1 三级模式结构

模式是数据中全体数据的逻辑结构和特征的描述,它涉及类型的定义,而不考虑具体的值。模式的一个具体值称为模式的一个实例。同一个模式可以有多个实例。

模式是相对稳定的,而实例是相对变动的,因为数据库中的数据总在不断地更新;模式反映的是数据的结构与联系,而实例反映的是数据库某一时刻的状态。

虽然目前 DBMS 产品多种多样,支持不同的数据模型,使用不同的数据库语言,建立在不同的操作系统之上,数据的存储结构也各不相同,但它们在总体结构上一般都采用三级模式结构。所谓三级模式结构,是指数据库系统由外模式、模式和内模式三级构成,如图 1-5 所示。

#### (1) 外模式

外模式又称用户模式,它是数据库用户能够看见和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,是与某一具体应用有关的数据的逻辑表示。

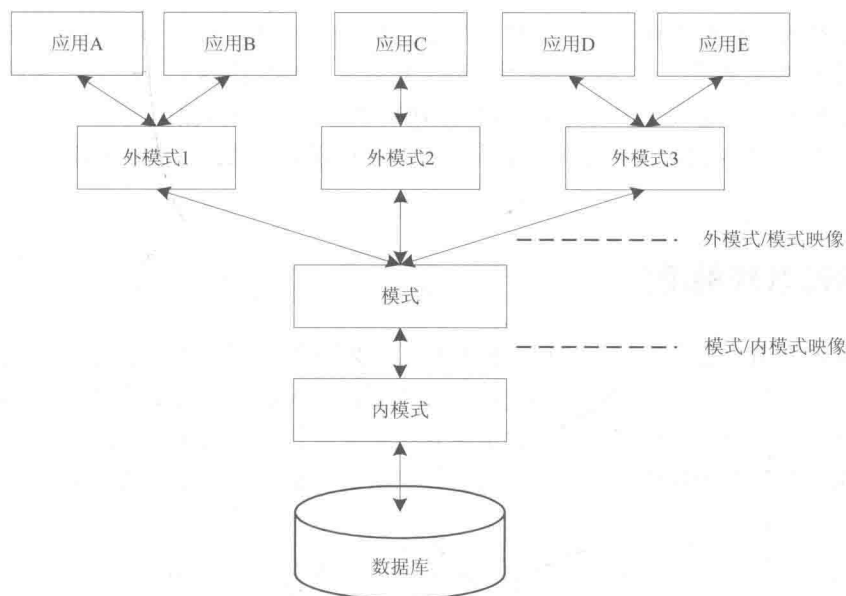


图 1-5 数据库系统的三级模式结构

一个数据库可以有多个外模式。由于不同用户在应用需求、看待数据的方式、对数据保密的要求等方面存在差异，可以为不同的用户分别建立外模式。同时，同一外模式可以为同一用户的多个应用系统所使用，但一个应用程序只允许使用一个外模式。

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

### (2) 模式

模式也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述。它是数据库系统模式结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体应用程序无关。

模式实际上是数据库中数据在逻辑级上的视图。一个数据库只有一个模式。数据库的模式以某一种数据模型为基础，统一考虑了所有用户的需求，并将这些需求有机地结合成一个逻辑整体。定义模式时，不仅要定义数据的逻辑结构，还需要定义数据之间的联系及相关的完整性、安全性要求。

### (3) 内模式

内模式也称存储模式，它是数据库中数据的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。

## 1.3.2 两级映像和数据独立性

三级模式结构是对数据库中数据的 3 个层次的抽象，它将数据的具体组织细节交给 DBMS 去处理，使用户能在抽象的逻辑层面上管理数据，而不必关心数据库的内部组织结构。DBMS 在三级模式之间提供两级映像，以实现这 3 个层次的联系与转换。

(1) 外模式/模式映像。外模式/模式映像定义了该外模式与模式之间的对应关系，对于每一个外模式，都有一个对应的外模式/模式映像。当模式改变时，由数据库管理员对各个外模式/模式映像做相应的改变，可使外模式保持不变，由于应用程序是依据数据的外模式编写



的，可以不必修改，从而实现了程序与数据的逻辑独立性。

(2) 模式/内模式映像。由于数据库只有一个内模式，所以模式/内模式映像是唯一的。它定义了数据库的全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变时，只需要通过改变模式/内模式映像保持模式不变，应用程序不必修改，从而实现程序与数据的物理独立性。

## 1.4 数据库系统的组成

数据库系统是引入数据库技术的计算机系统，一般由数据库、数据库管理系统（及相应的实用工具）、应用程序和数据库管理员组成，如图 1-6 所示。数据库是按照一定的组织形式保存在某种存储介质上的数据集合；数据库管理系统是管理数据库的专门软件；应用程序是指以数据库中的数据为基础的程序；数据库管理员负责整个数据库系统的正常运行。

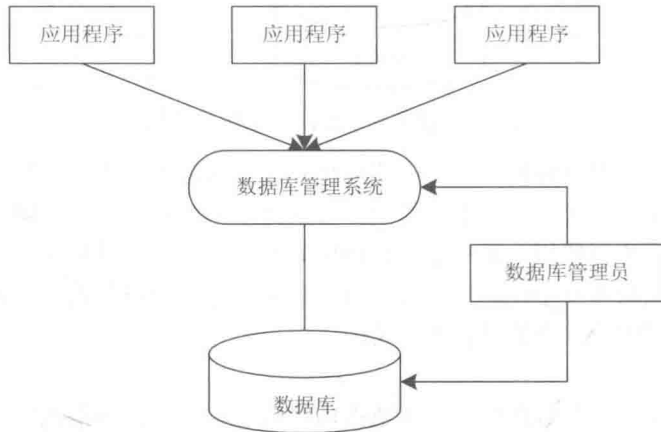


图 1-6 数据库系统的组成

下面主要从硬件平台、软件 and 人员几方面介绍数据库系统的组成。

### 1. 硬件平台

由于数据库中的数据量一般都比较大，而且 DBMS 具有丰富的功能使其自身的规模也很大，因此，整个数据库系统对硬件资源的要求很高。必须有足够大的内存，才能运行操作系统、数据库管理系统和应用程序，而且还需要足够大的硬盘空间存储数据库，最好还要有足够的存储备份数据的磁盘或光盘。

### 2. 软件

数据库系统中的软件主要包括：

(1) 数据库管理系统（及相关实用工具）。数据库管理系统是整个数据库系统的核心，是建立、使用和维护数据库的系统软件。同时，一般随数据库管理系统一起发行的还包括数据库厂商提供的一系列相关实用工具。

(2) 操作系统。数据库管理系统的很多底层操作都是靠操作系统完成的，数据库的安全控制等功能也是和操作系统共同实现。因此，数据库管理系统要和操作系统协同工作才能完成很多功能。不同的数据库管理系统对操作系统的需求不尽相同，比如，SQL Server 只支持