

普通高等教育“十三五”规划教材

SQL Server 数据库原理及实验教程

主 编 毋建宏 李鹏飞

副主编 卫 昆 朱烨行

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统地讲解了数据库技术及应用的基础知识,并将引导读者利用 SQL Server 2014 进行数据库的管理与开发实践。全书共 19 章,在介绍数据库系统相关概念与关系数据库有关知识的基础上,详细讲解安装和管理 SQL Server 2014、管理数据库和数据表、数据更新和查询、数据完整性、创建和操作索引与视图、T-SQL 编程、存储过程、触发器的开发、事务管理、数据库安全管理、数据库备份与恢复、数据库的导入和导出,最后结合某物流仓储管理系统开发案例,给出完整的数据库应用系统分析、设计与实施的方法步骤。本书提供了示例数据库,并附有章后习题,可方便读者学习使用。

本书既可作为高等院校信息管理与信息系统、电子商务、物流管理等经管类专业的数据库技术课程教材,也可作为想学习 SQL Server 2014 数据库的初学者及有一定数据库基础的技术人员的参考用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

SQL Server 数据库原理及实验教程 / 毋建宏, 李鹏飞主编. —北京: 电子工业出版社, 2020.1
ISBN 978-7-121-38217-8

I. ①S… II. ①毋… ②李… III. ①关系数据库系统—高等学校—教材 IV. ①TP311.132.3

中国版本图书馆 CIP 数据核字(2020)第 008651 号

责任编辑: 王志宇 文字编辑: 靳 平

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 23 字数: 602.8 千字

版 次: 2020 年 1 月第 1 版

印 次: 2020 年 1 月第 1 次印刷

定 价: 55.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254523, wangzy@phei.com.cn。

前 言

目前,数据库系统广泛应用于经济管理等领域,数据库技术已成为各种计算机应用系统开发的支撑技术。作为功能强大的关系数据库管理系统,Microsoft SQL Server 得到了广泛应用,是从事数据库应用开发与管理人员需要掌握的技术。“数据库技术及应用”是一门具有较强理论性和实践性相结合的专业基础课程,学习该课程应把理论知识与实际应用结合起来。经济管理等非理工科专业读者迫切需要一本能满足他们对数据处理与数据库应用理论学习及实验指导的教材。

本书着眼于数据库应用所需的基本原理知识和操作技能,注重对操作过程中易出现问题的讲解,配以经济管理中的应用案例,以1个实例为主线,贯穿各个实验的操作内容,并配合代码,对操作进行全面细致讲解。本书内容由浅入深、由简到繁,从基础操作到复杂管理、从单个技术应用到综合案例实践,特别适合经济管理专业等不具备深入计算机技术知识的读者学习。

通过对本书的学习,可使读者了解数据库基本理论、SQL Server 2014 数据库基本技术和经济管理应用中数据库系统操作与设计的方法,培养读者应用、管理及设计数据库的能力。本书既可作为高等院校信息管理与信息系统、电子商务、物流管理等经管类专业的数据库技术课程教材,也可作为企业信息化管理人员及其他 SQL Server 2014 应用人员参考用书。

本书对应课程的教学共需要 48~64 学时。当教学采用 64 学时,各章参考学时见下表;当教学采用 48 学时,第 11~14 章、第 19 章内容可作为课下的选读内容。

章 号	教 学 内 容	讲 授 学 时	上机实验学时
第 1 章	数据库系统概述	4	0
第 2 章	关系数据库基本原理	4	0
第 3 章	SQL Server 2014 数据库的安装及管理	1	1
第 4 章	管理 SQL Server 数据库	2	1
第 5 章	管理 SQL Server 数据表	2	1
第 6 章	数据更新	2	1
第 7 章	数据查询	3	2
第 8 章	高级查询	4	2
第 9 章	数据完整性	2	1
第 10 章	索引与视图	2	2
第 11 章	T-SQL 程序设计与游标	2	2
第 12 章	存储过程	2	2
第 13 章	触发器	2	1
第 14 章	函数	1	1
第 15 章	事务管理	2	1

续表

章 号	教 学 内 容	讲 授 学 时	上机实验学时
第 16 章	数据库安全管理	2	2
第 17 章	数据库备份与恢复	1	1
第 18 章	数据库的导入和导出	1	1
第 19 章	数据库应用系统的设计与开发	1	2
学时总计		40	24

本书由毋建宏、李鹏飞担任主编，卫昆、朱烨行担任副主编，其中毋建宏编写第 1、2、5、13、14、19 章，李鹏飞编写第 3、4、6 章，卫昆编写第 7~12 章，朱烨行编写第 15~18 章，全书由毋建宏统稿。本书的编写得到西安邮电大学教务处、电子工业出版社的大力支持，在此表示由衷的感谢。

本书受到“陕西高校青年创新团队”项目和西安邮电大学教务处教材建设项目资助。在本书的编写过程中，融入了编者长期以来数据库方面教学与研究的成果，广泛参考、吸收了众多学者的研究成果，借鉴了国内外大量的出版物和资料，由于编写体例的限制未在文中一一注明，只在最后参考文献中列出，同时在编写过程中得到许多同行专家的支持，在此谨向各位学者表示由衷的敬意和感谢；本科生罗洋、潘佳妮、潘景恒、廉佳颖、樊敏、任倩、黄美成、张璇等同学对本书的文字整理和插图做了大量工作，在此深表谢意。由于编者的水平有限，书中难免存在疏漏和错误，敬请广大读者予以批评、指正。

编 者

目 录

第 1 章 数据库系统概述	1
1.1 数据库的基本概念	1
1.1.1 数据	1
1.1.2 数据库	2
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	2
1.2.1 人工管理阶段	3
1.2.2 文件系统阶段	3
1.2.3 数据库技术阶段	4
1.3 数据库系统的特点	5
1.4 数据库系统的三级模式结构	6
1.5 数据模型	8
1.5.1 数据模型的分层	8
1.5.2 数据模型的三要素	8
1.5.3 概念模型与 E-R 图	9
1.5.4 常用的数据模型	11
1.6 小结	14
习题 1	14
第 2 章 关系数据库基本原理	15
2.1 关系模型概述	15
2.1.1 关系数据结构	15
2.1.2 关系模型的数据操作	18
2.1.3 关系模型的完整性约束	19
2.2 关系代数	22
2.2.1 集合运算	22
2.2.2 关系运算	22
2.3 关系规范化理论	24
2.3.1 函数依赖的基本概念	24
2.3.2 数据依赖对关系模式的影响	25
2.3.3 关系模式的规范化	26
2.3.4 关系模式的分解	31

2.4	数据库设计方法	32
2.4.1	数据库设计的过程	32
2.4.2	E-R 图转换为关系数据库模式	34
2.5	小结	34
	习题 2	35
第 3 章	SQL Server 2014 数据库的安装及管理	36
3.1	SQL Server 2014 简介	36
3.1.1	SQL Server 的发展历史	36
3.1.2	SQL Server 2014 的新特性	37
3.2	SQL Server 2014 的安装	37
3.2.1	安装要求	37
3.2.2	版本选择	37
3.2.3	安装过程	38
3.3	SQL Server 2014 的常用管理工具	47
3.3.1	SQL Server 配置管理器	47
3.3.2	SQL Server Management Studio	48
3.3.3	其他管理工具	49
3.4	SQL 和 T-SQL 的概述	49
3.4.1	SQL 的发展与特点	49
3.4.2	T-SQL 概述	49
3.5	小结	50
	习题 3	50
第 4 章	管理 SQL Server 数据库	51
4.1	SQL Server 数据库概述	51
4.1.1	SQL Server 数据库的结构	51
4.1.2	SQL Server 系统数据库	51
4.2	创建数据库	52
4.2.1	使用 SSMS 创建数据库	52
4.2.2	使用 T-SQL 语句创建数据库	53
4.3	修改数据库	54
4.3.1	使用 SSMS 查看及修改数据库属性	54
4.3.2	使用 T-SQL 语句修改数据库	55
4.4	删除数据库	56
4.4.1	使用 SSMS 删除数据库	56
4.4.2	使用 T-SQL 语句删除数据库	57
4.5	分离数据库和附加数据库	58
4.5.1	分离数据库	58
4.5.2	附加数据库	59

4.6	生成 SQL 脚本	60
4.7	小结	61
	习题 4	61
第 5 章	管理 SQL Server 数据表	62
5.1	SQL Server 数据表概述	62
5.1.1	表的概念	62
5.1.2	表的结构	62
5.1.3	列的数据类型	62
5.2	创建数据表	64
5.2.1	使用 SSMS 创建数据表	64
5.2.2	使用 T-SQL 语句创建数据表	65
5.3	修改数据表	66
5.3.1	使用 SSMS 查看数据表属性信息及修改数据表	67
5.3.2	使用 T-SQL 语句修改数据表	69
5.4	删除数据表	70
5.4.1	使用 SSMS 删除数据表	70
5.4.2	使用 T-SQL 语句删除数据表	70
5.5	小结	71
	习题 5	71
第 6 章	数据更新	72
6.1	插入数据	72
6.1.1	通过 SSMS 插入数据	72
6.1.2	用 INSERT 语句插入数据	73
6.2	修改数据	76
6.2.1	通过 SSMS 修改数据	76
6.2.2	用 UPDATE 语句修改数据	77
6.3	删除数据	78
6.3.1	通过 SSMS 删除数据	79
6.3.2	用 DELETE 语句删除数据	79
6.4	小结	81
	习题 6	81
第 7 章	数据查询	82
7.1	SELECT 语句的结构与执行	82
7.1.1	SELECT 语句的语法结构	82
7.1.2	SELECT 语句各子句的顺序及功能	83
7.1.3	SELECT 语句各子句的执行	83
7.2	基本查询	84
7.2.1	简单查询	84

7.2.2	条件查询	90
7.2.3	查询结果排序	99
7.2.4	数据统计查询	101
7.3	小结	108
	习题 7	108
第 8 章	高级查询	109
8.1	连接查询	109
8.1.1	基本连接	109
8.1.2	JOIN 关键字	111
8.1.3	内部连接	111
8.1.4	外部连接	112
8.1.5	交叉连接	114
8.1.6	自连接	114
8.2	集合查询	115
8.2.1	联合查询	115
8.2.2	集合交集	117
8.2.3	集合差	118
8.3	子查询	118
8.3.1	单值子查询	119
8.3.2	带有 ALL、ANY、SOME 运算符的子查询	120
8.3.3	带有 IN 运算符的子查询	121
8.3.4	带有 EXISTS 运算符的子查询	123
8.3.5	在 FROM 子句中使用子查询	124
8.4	小结	125
	习题 8	125
第 9 章	数据完整性	126
9.1	数据完整性概述	126
9.2	使用约束实施数据完整性	127
9.2.1	主键约束	127
9.2.2	外键约束	129
9.2.3	非空约束	132
9.2.4	唯一性约束	133
9.2.5	默认值约束	135
9.2.6	检查约束	136
9.3	使用规则实施数据完整性	138
9.3.1	创建规则	139
9.3.2	查看规则	139
9.3.3	绑定与松绑规则	141

9.3.4 删除规则	142
9.4 使用默认值实施数据完整性	143
9.4.1 创建默认值	143
9.4.2 查看默认值	143
9.4.3 绑定与松绑默认值	145
9.4.4 删除默认值	146
9.5 小结	147
习题 9	147
第 10 章 索引与视图	148
10.1 索引概述	148
10.1.1 索引的概念	148
10.1.2 索引的分类	149
10.2 索引操作	150
10.2.1 创建索引	150
10.2.2 查看及修改索引	154
10.2.3 删除索引	155
10.3 视图概述	157
10.3.1 视图的概念	157
10.3.2 视图的作用	157
10.3.3 视图的限制	157
10.4 视图操作	157
10.4.1 创建视图	157
10.4.2 修改视图	161
10.4.3 删除视图	162
10.5 视图应用	163
10.5.1 在 SSMS 界面中操作视图记录	163
10.5.2 视图中的数据更新	164
10.6 小结	166
习题 10	166
第 11 章 T-SQL 程序设计与游标	167
11.1 数据与表达式	167
11.1.1 常量与变量	167
11.1.2 运算符与表达式	169
11.2 流程控制语句	173
11.2.1 语句块和注释	173
11.2.2 分支语句	175
11.2.3 循环语句	178
11.2.4 批处理	179

11.3	游标	180
11.3.1	游标概述	180
11.3.2	声明游标	180
11.3.3	打开游标	181
11.3.4	读取游标	182
11.3.5	关闭与释放游标	183
11.3.6	使用游标修改和删除数据	183
11.4	小结	185
习题 11		185
第 12 章	存储过程	186
12.1	存储过程概述	186
12.1.1	存储过程的概念	186
12.1.2	存储过程的种类	186
12.2	创建和管理存储过程	187
12.2.1	创建存储过程	187
12.2.2	执行存储过程	189
12.2.3	查看存储过程	191
12.2.4	修改存储过程	192
12.2.5	删除存储过程	194
12.3	带参数的存储过程	195
12.3.1	存储过程的参数类型	195
12.3.2	创建和执行带输入参数的存储过程	195
12.3.3	创建和执行带输出参数的存储过程	196
12.3.4	存储过程的返回值	197
12.4	小结	199
习题 12		199
第 13 章	触发器	200
13.1	触发器的概述	200
13.1.1	触发器的概念	200
13.1.2	触发器的作用	201
13.1.3	触发器的类型	201
13.1.4	触发器应用的两个逻辑表	202
13.2	创建和管理 DML 触发器	202
13.2.1	创建 DML 触发器	202
13.2.2	其他类型的 DML 触发器	213
13.2.3	修改触发器	213
13.2.4	查看触发器	213
13.2.5	删除触发器	214

13.2.6 禁用和启用触发器·····	215
13.3 创建 DDL 触发器·····	217
13.3.1 DDL 触发器类型·····	218
13.3.2 创建 DDL 触发器·····	218
13.4 小结·····	218
习题 13·····	219
第 14 章 函数·····	220
14.1 系统内置函数·····	220
14.1.1 聚合函数·····	220
14.1.2 配置函数·····	226
14.1.3 游标函数·····	227
14.1.4 日期和时间函数·····	229
14.1.5 数学函数·····	232
14.1.6 元数据函数·····	238
14.1.7 字符串函数·····	240
14.1.8 文本和图像处理函数·····	245
14.2 用户自定义函数·····	246
14.2.1 标量值函数·····	246
14.2.2 内嵌表值函数·····	247
14.2.3 多语句表值函数·····	248
14.2.4 用户自定义函数的注意事项·····	248
14.2.5 查看用户定义函数·····	249
14.2.6 删除用户定义函数·····	250
14.3 小结·····	251
习题 14·····	251
第 15 章 事务管理·····	252
15.1 事务概述·····	252
15.1.1 事务的概念·····	252
15.1.2 事务的特性·····	252
15.1.3 事务的运行模式·····	253
15.1.4 多事务的并发问题·····	253
15.2 事务管理与应用·····	256
15.3 锁机制·····	258
15.3.1 锁的简介·····	258
15.3.2 隔离级别·····	260
15.3.3 查看锁和死锁·····	262
15.3.4 封锁协议·····	265
15.4 小结·····	268

习题 15	268
第 16 章 数据库安全管理	270
16.1 SQL Server 的安全机制	270
16.1.1 身份验证模式	270
16.1.2 更改身份验证模式	270
16.2 创建、管理登录名和数据库用户	272
16.2.1 创建登录名	272
16.2.2 管理登录名	276
16.2.3 创建和管理数据库用户	278
16.3 管理角色	279
16.3.1 角色的种类	279
16.3.2 管理服务器角色	280
16.3.3 管理数据库角色	282
16.4 管理权限	285
16.4.1 权限的种类	285
16.4.2 授予权限	286
16.4.3 禁止与撤销权限	287
16.4.4 查看权限	288
16.5 小结	289
习题 16	290
第 17 章 数据库备份与恢复	291
17.1 数据库备份	291
17.1.1 数据库备份概述	291
17.1.2 创建和管理备份设备	292
17.1.3 备份数据库操作	296
17.2 数据库恢复	300
17.2.1 数据库的恢复模式	300
17.2.2 配置恢复模式	302
17.2.3 恢复数据库操作	303
17.3 小结	309
习题 17	310
第 18 章 数据库的导入和导出	311
18.1 导入和导出概述	311
18.2 导入数据	311
18.3 导出数据	315
18.4 小结	320
习题 18	320

第 19 章 数据库应用系统的设计与开发	321
19.1 数据库设计的基本步骤	321
19.2 采用 ADO.NET 组件访问 SQL Server	322
19.2.1 ADO.NET 组件简介	322
19.2.2 连接式访问数据库	323
19.3 采用 JDBC 访问 SQL Server	324
19.3.1 JDBC 简介	324
19.3.2 JDBC 连接 SQL Server 数据库的步骤	325
19.4 某物流仓储管理系统开发案例	328
19.4.1 需求分析	328
19.4.2 系统分析	330
19.4.3 系统设计	337
19.4.4 主要功能模块实现	345
19.5 小结	350
习题 19	351
参考文献	352

电子工业出版社有限公司
版权所有
盗版必究

第 1 章

数据库系统概述

本章将从数据库的基本概念入手，在论述数据、数据库、数据库管理系统和数据库系统的基础上，介绍数据管理技术的产生与发展、数据库技术的特点、数据库系统的三级模式结构和常用的数据模型，使人们对数据库有一个初步认识，掌握其核心概念，并理解这些概念在数据库技术中的地位和作用。

1.1 数据库的基本概念

1.1.1 数据

数据（Data）是描述事物的符号记录。日常生活中，人们对世界的认识就是通过各种感官获取事物的特征值来辨别事物的。这些不同种类的特征值就是人们描述不同事物的符号记录。

在计算机科学中，数据是所有能输入计算机并被计算机程序处理的符号介质的总称。在计算机系统中，各种字母、数字符号的组合、语音，图形，图像等统称为数据。数据经过加工后就成为信息。数据是数据库存储的对象，也是数据库管理系统处理的对象。

例如，对一个员工的基本情况描述包括张磊、员工号 02021501、男、1998 年 5 月出生、陕西西安人、工作于黄河机器制造厂、在家用电器部门、喜欢打篮球。为了有效管理员工，我们经常通过表格对描述的事物进行归类，如表 1-1 所示。

表 1-1 员工基本信息表

员 工 号	姓 名	性 别	出生年月	籍 贯	单 位	部 门	爱 好
02021501	张磊	男	1998 年 5 月	陕西西安	黄河机器制造厂	家用电器	篮球
.....							

从表 1-1 可以看出，数据的表达包含两个方面的内容，即语义和数据，语义如表头所示，解释了数据的含义；数据则是实际的数据值，两者结合有效地描述了一个具体事物。将某类事物的基本特征抽象出来后就形成关于这个事物的基本数据语义描述，称为事物的特征。按照特征具体描述一个员工就是关于其数据的值，称为一条记录。记录是计算机中表示和存储数据的一种格式，这样存储的数据是有结构的，可称为结构化数据。在对数据进行结构化处理后能非常方便地进行数据管理和数据管理，这是数据库的主要功能之一。

1.1.2 数据库

数据库 (Database, DB) 是指长期存储在计算机内的、有组织的、可共享的结构化数据集合。数据库往往是一个单位或一个应用领域的通用数据处理系统, 它存储的是属于企业和事业部门、团体和个人有关数据的集合。数据库中的数据是从全局观点出发建立的, 按一定的数据模型进行组织、描述和存储。其结构基于数据间的自然联系, 可提供一切必要的存取路径。数据库中的数据不再针对某个应用, 而是面向组织的所有应用, 其具有整体的结构化特征。

数据库中的数据供众多用户所共享, 因此摆脱了具体程序的限制和制约。不同的用户可以按各自的用法使用数据库中的数据; 多个用户可以同时共享数据库中的数据资源, 即不同的用户可以同时存取数据库中的同一个数据, 这就是数据库的共享性, 共享性不仅满足了各用户对信息内容的要求, 同时满足了各用户之间信息通信的要求。

1.1.3 数据库管理系统

数据库管理系统 (Data Base Management System, DBMS) 是数据库系统的核心构成。它是一个系统软件, 负责数据库中的数据组织、数据操纵、数据维护、数据控制及保护、数据服务等。

数据库管理系统作为操纵和管理数据库的大型软件, 用于建立、使用和维护数据库。它对数据库进行统一的管理和控制, 以保证数据库的安全性和完整性。用户通过该系统访问数据库中的数据, 数据库管理员也通过它进行数据库的维护工作。它可使多个应用程序和用户用不同的方法在同一时刻或不同时刻去建立、修改和询问数据库。数据库管理系统提供数据定义语言 (Data Definition Language, DDL) 与数据操作语言 (Data Manipulation Language, DML), 供用户定义数据库的模式结构与权限约束, 从而实现了对数据的追加、删除等操作。

数据库管理系统的主要类型有 3 种: 层次型数据库管理系统、网状数据库管理系统和关系数据库管理系统, 其中, 关系数据库管理系统应用最为广泛。

1.1.4 数据库系统

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, 能够有组织地、动态地存储大量相关数据, 提供数据处理和信息资源共享的便利手段。数据库系统由数据库 (数据)、数据库管理系统 (软件)、计算机硬件、操作系统及数据库管理员组成。

在数据库、数据库管理系统和数据库系统三者之中, 数据库管理系统是数据库系统的组成部分, 数据库又是数据库管理系统的管理对象, 也就是说数据库系统包括数据库管理系统, 数据库管理系统包括数据库。

1.2 数据管理技术的产生和发展

数据管理技术是指对各种数据进行有效分类、组织、编码、存储、检索、维护和应用, 是数据处理的中心问题。随着人类社会的发展和技术应用的深入, 尤其是计算机技术

的快速发展，数据管理技术在广度和深度上都得到了极大的发展。在应用需求的推动下，计算机硬件、软件的发展使数据管理技术水平得到了极大提高。一般来讲，数据管理技术经历了三个发展阶段：人工管理阶段、文件系统阶段和数据库技术阶段。

1.2.1 人工管理阶段

20 世纪 50 年代中期以前，计算机主要用于科学计算。当时存储设备比较落后，还没有磁盘等直接存取设备、外部存储器，只有磁带、卡片和纸带等。在软件方面，既没有操作系统，也没有管理数据的专门软件。数据处理方式主要采用批处理。

在人工管理阶段，数据是面向应用程序的，每个应用程序都需要设计独立的数据集合。程序与数据之间的关系如图 1-1 所示。由于数据结构需要不同的应用程序定义和管理，缺乏专门的系统负责数据的管理，多个应用程序涉及相同的数据时需要重新定义。因此，该阶段的数据管理具有以下特点。

- (1) 数据不被保存。
- (2) 没有专用的数据管理软件。
- (3) 数据共享性低。
- (4) 数据独立性差。

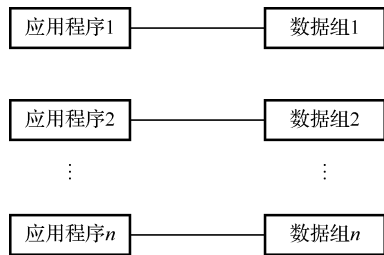


图 1-1 人工管理阶段程序与数据之间的关系

1.2.2 文件系统阶段

从 20 世纪 50 年代后期到 60 年代中期，计算机开始广泛应用于信息管理领域。大量的数据存储、检索和维护成为迫切需求，数据结构和数据管理技术迅速发展起来。在硬件方面，有了磁盘、磁鼓等直接存储设备；在软件方面，出现了高级语言和操作系统，并且操作系统中有了专门管理数据的文件系统，同时出现了新的联机实时处理方式。在此基础上，人们开始大规模使用文件系统来进行数据管理。

用文件系统管理数据的特点如下。

(1) 数据长期保留。可以将数据长期保留在外存上被反复处理，即可以经常对数据进行查询、修改和删除等操作。

(2) 数据的独立性。由于有了操作系统，所以可以利用文件系统专门的数据管理，使程序员能够集中精力在算法设计上，而不必过多地考虑细节。如要保存数据时，只需给出保存指令，而不必让所有的程序员都精心设计一套程序，控制计算机物理地实现保存数据。在读取数据时，只需给出文件名，而不必知道文件具体的存放地址。文件的逻辑结构和物理存储结构由系统进行转换，程序与数据有了一定的独立性。数据的改变不一定

要引起程序的改变。例如，保存的文件中有 100 条记录，使用某一个查询程序，当文件中有 1000 条记录时，仍然使用保留的这个查询程序。

(3) 可以实时处理。由于有了直接存取设备，也就有了索引文件、链接存取文件、直接存取文件等，所以既可以采用顺序批处理，也可以采用实时处理方式。数据的存取以记录为基本单位。

文件系统阶段与人工管理阶段相比，在数据管理方法和管理模式上有了很大进步，但一些根本性问题仍没有得到彻底解决，主要表现在以下两个方面。

(1) 数据的共享性差，冗余度大。

一个数据文件只能对应于同一程序员的一个或几个程序，当不同的应用程序具有部分相同的数据时，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。

(2) 缺乏数据独立性。

要对现有的数据增加一些新的应用会很困难，系统不容易扩充。数据和程序相互依赖，一旦改变数据的逻辑结构，就必须修改相应的应用程序。应用程序发生变化时，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。因此，数据和程序之间缺乏独立性。

由此可见，文件系统仍然是一个不具有弹性的、无结构的数据集合，即文件与文件、文件与程序之间是孤立的，不能反映现实世界事物之间的内在联系。在文件系统阶段，程序与数据之间的关系如图 1-2 所示。

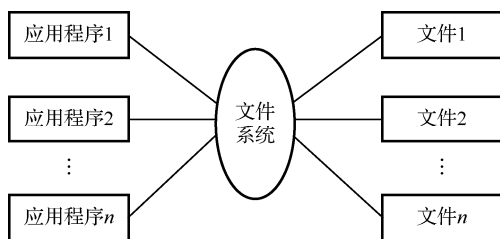


图 1-2 文件系统阶段程序与数据之间的关系

1.2.3 数据库技术阶段

20 世纪 60 年代后期，数据管理进入数据库技术阶段。该阶段计算机应用于管理的规模更加庞大，数据量急剧增加，数据共享的要求也越来越强烈。与此同时，计算机硬件、软件有了进一步的发展。在处理方式上，联机实时处理的需求增加，出现了分布式处理。文件系统的数据管理方法已无法适应应用系统的需要。为解决多用户、多个应用程序共享数据的需求，产生了数据库管理系统。

尤其是 1970 年美国 IBM 公司的 E.F.Codd 连续发表论文，提出了关系模型，使数据库技术在概念、原理和方法等方面均得到了迅速发展。数据库系统克服了文件系统的缺陷，提供了对数据更高级、更有效的管理。数据库技术的目标是解决数据冗余问题，实现数据的独立性和共享性，同时解决由于数据共享而带来的数据完整性、安全性及并发控制等一系列问题。为实现这个目标，数据库的运行必须由一个软件系统来控制，这个软件系统称为数据库管理系统，其将程序员进一步解脱出来。程序员不再需要考虑程序中的数据

是不是因为改动而造成不一致，也不用担心由于应用功能的扩充、程序重写而导致数据结构重新变动。这个阶段的数据管理技术具有以下优点。

- (1) 实现了数据的整体结构化。
- (2) 数据的存取方式更加灵活。
- (3) 数据的共享性高、冗余少、易扩充。
- (4) 数据的独立性高。
- (5) 统一的数据控制功能。

在数据库技术阶段，应用程序与数据之间的关系如图 1-3 所示。

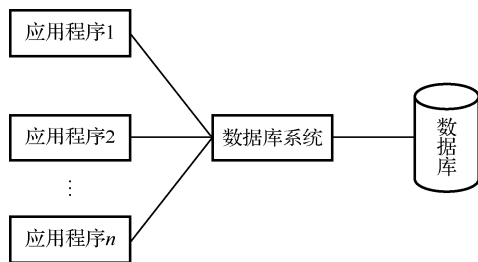


图 1-3 数据库技术阶段应用程序与数据之间的关系

1.3 数据库系统的特点

数据库系统的特点包括以下 4 点。

1. 数据结构化

数据库系统实现了整体数据的结构化，这是数据库最主要的特征之一。这里所说的“整体”结构化是指在数据库中，从全局、整体观点来组织数据，数据不再是仅针对某个应用、某个程序，而是面向全组织，为多种应用所共享，从而避免不必要的冗余；从内部结构上来讲，在数据库中，所有的数据将采用统一的数据结构，数据之间存在联系，从而有利于数据库管理系统的统一操作。因此数据结构化不只是数据内部的结构化，而是整体式结构化。

2. 数据的共享性高，冗余少，易扩充

由于数据是面向整体的，故数据可以被多个用户、多个应用程序共享使用，能够极大地减少数据冗余，节约存储空间，避免数据之间的不相容性与不一致性。

数据冗余的减少可以节约存储空间，使数据的存储、管理和查询都容易实现；使数据统一，避免产生数据不一致的问题；便于数据维护，避免数据统计错误。同时数据库系统可以通过数据模型和数据控制机制来提高数据的共享性，从而使得系统现有用户或程序可以共同享用数据库中的数据；当系统需要扩充时，再开发的新用户或新程序还可以共享原有的数据资源，多用户或多程序可以在同一时刻共同使用同一数据，最终保证了应用系统能够便捷地实现系统扩充。

3. 数据的独立性高

数据和程序之间相互的依赖性低、独立性高的特性称为数据独立性。数据与程序独立，把数据的定义从程序中分离出去，加上存取数据由 DBMS 负责，从而简化了应用程

序的编制，大大减少了对应用程序的维护和修改。

数据的独立性包括数据的物理独立性和逻辑独立性。

(1) 数据的物理独立性 (Physical Data Independence) 是指应用程序对数据存储结构的依赖程度。数据物理独立性高是指当数据的物理结构发生变化时，应用程序不需要修改也可以正常工作。数据库系统之所以具有数据物理独立性高的特点，是因为数据在磁盘上如何存储由 DBMS 管理，用户程序不用了解，应用程序要处理的只是数据的逻辑结构，数据库管理系统能够提供数据的物理结构与逻辑结构之间的映像 (Mapping) 或转换功能。这样一来当数据的物理存储结构改变时，用户程序不用改变。

(2) 数据的逻辑独立性 (Logical Data Independence) 是指应用程序对数据全局逻辑结构的依赖程度，逻辑独立性高即用户的应用程序与数据库的逻辑结构是相互独立的，当数据库系统的数据逻辑结构改变时，它们对应的应用程序不需要做任何修改。

数据库中的数据逻辑结构分全局逻辑结构和局部逻辑结构两种：全局逻辑结构是指全系统总体的数据逻辑结构，它是按全系统使用的数据、数据的属性及数据联系来组织的；局部逻辑结构是指具体一个用户或程序使用的数据逻辑结构，它是根据用户自己对数据的需求进行组织的。

4. 数据由 DBMS 统一管理和控制

数据库的共享是并发 (Concurrency) 共享的，即多个用户可以同时存取数据库中的数据，甚至可以同时存取数据库中的同一个数据，这个机制必须通过 DBMS 的统一管理和控制才能实现。DBMS 的功能如下。

(1) 数据的安全性控制 (Security)，即保护数据库的安全，以防止不合法使用造成的数据泄露、破坏和更改。数据安全性受到威胁是指出现用户看到了不该看到的数据、修改了无权修改的数据、删除了不能删除的数据等现象。数据安全性被破坏有两种情况：用户有超越自身拥有的数据操作权的行为；出现了违背用户操作意愿的结果。

(2) 数据的完整性控制 (Integrity)，即保证数据的正确性、有效性和相容性，防止不符合语义的数据输入或输出所采用的控制机制。数据的完整性控制包括两项内容：提供进行数据完整性定义的方法，用户利用该方法定义数据应满足的完整性条件；提供进行检验数据完整性的功能，特别是在数据输入和输出时，系统应能够自动检查其是否符合已定义的完整性条件，以避免错误的数据进入数据库或从数据库中流出，造成不良后果。数据完整性的高低是决定数据库中数据可靠程度和可信程度的重要因素。

(3) 数据库的并发访问控制 (Concurrency)，即排除由于数据共享，用户并行使用数据库造成的数据不完整和系统运行错误的问题。

(4) 数据库的故障恢复 (Recovery)，即通过记录数据库运行的日志文件和定期做数据备份工作，保证数据在受到破坏时，能够及时使数据库恢复到正确状态。

1.4 数据库系统的三级模式结构

数据库领域公认的数据库系统的标准结构是三级模式，包括外模式、模式、内模式。数据库系统通过三级模式结构有效组织和管理数据，提高了数据库的逻辑独立性和物理独立性。用户级对应外模式，概念级对应模式，物理级对应内模式，使不同级别的用户对数据库形成不同的视图。所谓视图，就是指观察、认识和理解数据的范围、角度和方法，是数据库

在用户应用中的反映，不同层次（级别）用户所接触的数据是不同的，如图 1-4 所示。

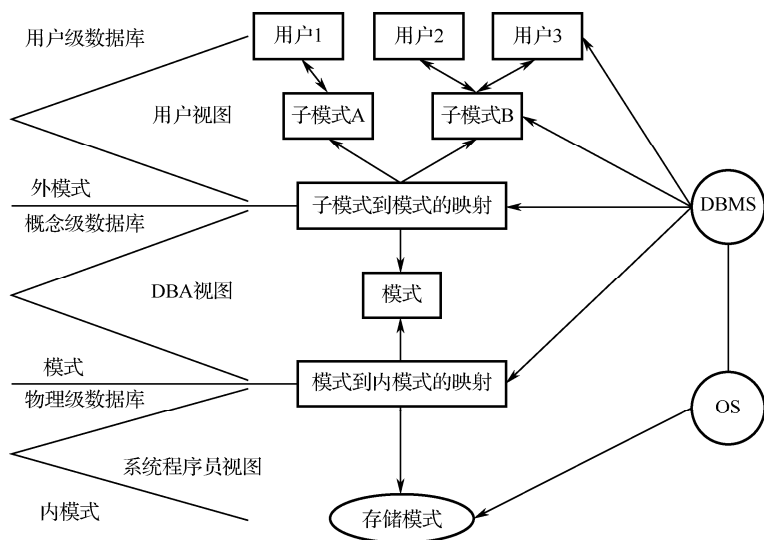


图 1-4 数据库的三级模式和两级映射

1. 三级模式

(1) 外模式：是指应用程序用到的部分数据的逻辑结构。一个数据库可以有多个外模式。

(2) 模式：是指数据库管理员用到的视图，又称概念模式，就是在 DBMS 的可视化界面中看到的数据库对象。一个数据库只有一个概念模式。

(3) 内模式：是数据库的最低层模式，是数据物理结构和存储方式的描述。一个数据库只有一个内模式。

2. 两级映射

(1) 外模式/模式的映射。

模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构。对应于同一模式可以有任意多个外模式。对应于任意一个外模式，数据库都有一个外模式/模式映射，它定义了外模式和模式之间的对应关系。当模式发生变化时，由数据库管理员对外模式/模式映射做相应的改变，可以使外模式保持不变。应用程序是依照外模式编写的，所以不必修改，保证了数据与程序的逻辑独立性。

(2) 模式/内模式的映射。

数据库中有一个模式，也只有一个内模式，所以模式/内模式的映射是唯一的，它定义了数据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发生变化时，由数据管理员对模式/内模式映射进行相应改变，可以使模式保持不变，应用程序也不发生变化，从而保证了数据与程序的物理独立性。

3. 三级模式的关系

(1) 模式是数据库的核心与关键。

(2) 内模式依赖于模式，独立于外模式和存储设备。

(3) 外模式面向具体的应用，独立于内模式和存储设备。

(4) 应用程序依赖于外模式，独立于模式和内模式。

1.5 数据模型

数据模型是一组严格定义的概念集合，这些概念精确地描述了系统的静态特征、动态特征和完整性约束条件，是数据特征的抽象，是数据库系统中用于提供信息表示和操作手段的形式架构。数据模型包括数据结构、数据操作和完整性约束三要素。计算机处理的只能是数据，因此解决实际问题时，人们需要把具体的事物转换成计算机能够处理的数据，即把现实世界中具体的人、物、活动、概念等用数据模型进行抽象、描述和表示。

1.5.1 数据模型的分层

数据模型应具有以下三方面的基本要求：①能比较真实地抽象现实世界；②能被人们理解；③便于计算机实现。根据数据抽象的级别不同，将数据模型划分为以下三层：

(1) 概念模型：又称信息模型，是指按用户的观点来对数据和信息建模，是对现实世界的第一层抽象。它既是独立于计算机系统的模型，又是用户和数据库设计人员之间进行交流的有力工具。概念模型有较强的语义表达能力，并且概念简单、清晰，易于用户理解。概念模型的表示方法最为著名的是实体-联系（E-R）方法。

(2) 逻辑模型：是数据抽象的第二层，用于描述数据库数据的全局逻辑结构，按计算机的观点对数据建模，主要用于 DBMS 的实现，不同的 DBMS 提供的逻辑模型各不相同。它涉及计算机系统和数据库管理系统，有严格的形式化定义，以便于在计算机系统中实现。目前最常用的数据模型一般有层次模型、网状模型、关系模型和面向对象模型 4 种。

(3) 物理模型：是数据抽象的底层，用于描述数据在计算机中的物理存储结构和存取方法。它的结构由 DBMS 的设计决定，并且与操作系统、计算机硬件相关，具体实现是 DBMS 的任务。对于一般用户而言，有基本的概念了解就行，不必考虑物理细节。

为了把现实世界中的具体事物进行抽象，最后用数据进行描述，并存储到计算机中，必须借助数据模型的一步步抽象来完成。首先将现实世界抽象为信息世界，然后将信息世界抽象为数据世界，最后再将数据世界映射为机器世界，如图 1-5 所示。因此数据模型的建模过程实际上就是将要描述的事物从现实世界到信息世界，再到数据世界，最后到机器世界逐步转换的过程。

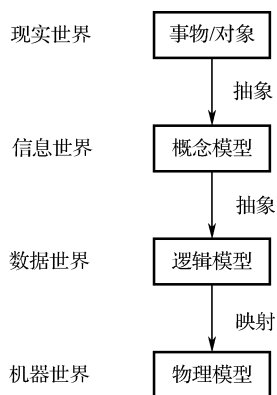


图 1-5 现实世界事物的抽象过程

1.5.2 数据模型的三要素

1. 数据结构

数据结构是定义如何构造一组数据库的规则，也是一种物理数据模型。数据结构由数据元素和数据项组成。数据元素表示一个事物的一组数据；数据项是构成数据元素的数据。例如，要描述货物信息，可包括货物的编号、名称、规格、数量等数据。货物的编号、名称、规格、数量等数据就构成了货物信息描述的数据项，其中数据项的一组数据就

构成了货物信息的一个数据元素。数据结构是对系统静态特性的描述。

2. 数据操作

数据操作是对数据库中各种对象（型）的实例（值）允许执行的操作及有关的操作规则，主要含有检索、更新（包括插入、删除、修改）两类操作，是对系统动态特性的描述。简而言之，对数据集中的数据元素进行的某种处理就是数据操作。

3. 数据的完整性约束

数据的完整性约束是指一组约束规则的集合，是数据及其联系所具有的制约和储存规则，用于限定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、相容。这些数据的约束条件与具体的应用有关，可能很简单，也可能很复杂。

1.5.3 概念模型与 E-R 图

1. 概念模型的基本特点

概念模型独立于计算机系统，不涉及信息在系统中的表示，重点描述某特定组织所关心的信息结构，强调语义表达，实现对现实世界的第一层抽象。因此，概念模型应具有以下特点。

- (1) 语义表达能力强。
- (2) 易于理解和交流。
- (3) 与 DBMS 无关。
- (4) 易于向逻辑数据模型转换。

概念模型是对信息世界的建模，因此，概念模型应该能够方便、准确地表示信息世界中的常用概念。

2. E-R 图

概念模型的表示方法很多，其中最常用的是 P.P.S.Chen 于 1976 年提出的实体-联系模型（Entity-Relationship Model, E-R 模型）。该模型用 E-R 图来描述现实世界，如表 1-2 所示。

表 1-2 E-R 图中各图形的含义

对象类型	E-R 图的表示方法	E-R 图的表示图示	员工示例
实体	用矩形表示，矩形内写明实体名称		
属性	用椭圆形表示，椭圆内写明属性名称，并用无向边将其与实体连接起来		
联系	用菱形表示，菱形内写明联系名称，用无向边分别与有关实体连接起来，并在无向边旁标明联系的类型		

在信息世界中，常用的概念如下。

1) 实体（Entity）

实体是现实世界中可区别于其他对象的一件“事物”或“物体”，实体可以是可触及的对象，如一个员工、一本书、一辆汽车；也可以是抽象的事件，如一堂课、一次比赛等。在实体联系图中，用矩形框表示实体。具有相同类型及性质的实体集合构成实体集，如所有订单的集合就定义为订单实体集。

2) 属性 (Attributes)

实体是通过一组属性来描述的。属性是实体集的每一个实体都具有的特征描述, 同一实体集中所有实体都具有相同的属性。如员工实体集中所有的员工都有员工号、姓名、年龄、性别、部门等方面的属性。在实体联系图中, 用椭圆表示实体属性。

属性有“型”“值”之分, “型”即为属性名, 如姓名、年龄、性别是属性的型; “值”即为属性的具体内容, 如(990001、张立、20、男、家用电器) 这些属性值的集合表示了一个员工实体。

3) 实体型 (Entity Type)

若干个属性组成的集合可以表示一个实体的类型, 简称实体型, 如员工(员工号、姓名、年龄、性别、部门) 就是一个实体型。

4) 实体集 (Entity Set)

同型实体的集合称为实体集, 如所有的员工、所有的部门等。

5) 键 (Key)

能唯一标识一个实体的属性或属性集称为实体的键, 如员工的员工号。由于员工的姓名可能有重名, 故不能作为员工实体的键。

6) 域 (Domain)

属性值的取值范围称为该属性的域, 如员工号的域为 6 位整数、姓名的域为字符串集合、年龄的域为小于 40 的整数、性别的域为(男, 女)。

7) 联系 (Relationship)

在现实世界中, 事物内部及事物之间是有联系的, 这些联系同样也要抽象和反映到信息世界中来。可以将信息世界抽象为实体型内部的联系和实体型之间的联系。实体型内部的联系通常是指组成实体的各属性之间的联系; 实体型之间的联系通常是指不同实体集之间的联系。在实体联系图中, 用菱形框表示实体间的联系。

反映实体型及其联系的结构形式称为实体模型, 它是现实世界及其联系的抽象表示。两个实体型之间的联系有以下 3 种类型。

(1) 一对一联系 (1:1)。

实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应, 反之亦然, 则称实体集 A 与实体集 B 为一对一的联系, 记作 1:1, 如工厂与厂长、观众与座位、病人与床位, 如图 1-6 (a) 所示。

(2) 一对多联系 (1:n)。

实体集 A 中的一个实体与实体集 B 中的多个实体相对应, 反之, 实体集 B 中的一个实体至多与实体集 A 中的一个实体相对应。记作 1:n, 如工厂与员工、公司与职员、省与市, 如图 1-6 (b) 所示。

(3) 多对多联系 (m:n)。

实体集 A 中的一个实体与实体集 B 中的多个实体相对应, 反之, 实体集 B 中的一个实体与实体集 A 中的多个实体相对应。记作 (m:n), 如仓库与产品、工厂与产品, 如图 1-6 (c) 所示。

多对多联系是一类比较复杂的联系, 一般数据库管理系统不直接支持多对多联系, 因此总是将一个多对多联系转换为两个一对多联系。例如, 仓库和产品之间的联系就是一个多对多联系, 此时要在它们之间增加一个储存联系, 仓库和储存之间、产品与储存之间就

构成了两个一对多的联系。

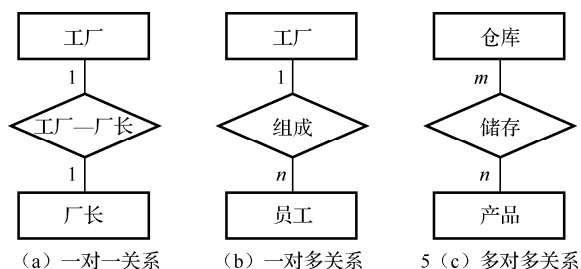


图 1-6 两实体之间的联系

3. 概念结构的设计

对于每一个实体集，可指定一个码为主码。如果用矩形框表示实体集、用椭圆形表示属性、用线段连接实体集与属性，则当一个属性或属性组合被指定为主码时，在实体集与属性的连接线上标记一斜线，可以用图 1-7 描述产品储存管理系统中的实体集及每个实体集涉及的属性。

若实体集 A 和实体集 B 之间存在各种关系，通常把这些关系称为“联系”。实体集及实体集联系的图就表示为 E-R 模型；从分析用户项目涉及的数据对象及数据对象之间的联系出发，到获取 E-R 图的过程称为概念结构设计。联系用菱形表示，通过直线与实体相连，这样构成的图就是 E-R 图，它是 E-R 模型的描述方法。

例如，在储存系统中，一类产品同时有若干个仓库储存（产品的属性主要包括产品号、产品名、价格、质量等），而一个仓库可以同时保管多类产品（仓库的属性主要包括仓库号、仓库名、地址等），则产品与仓库之间具有多对多联系，如图 1-8 所示。

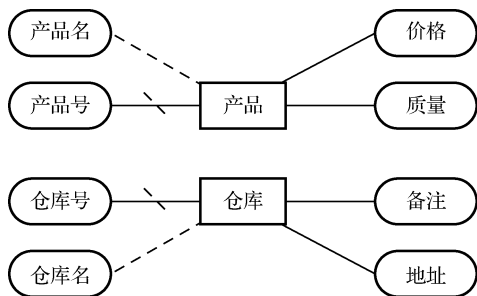


图 1-7 产品和仓库实体集属性的描述

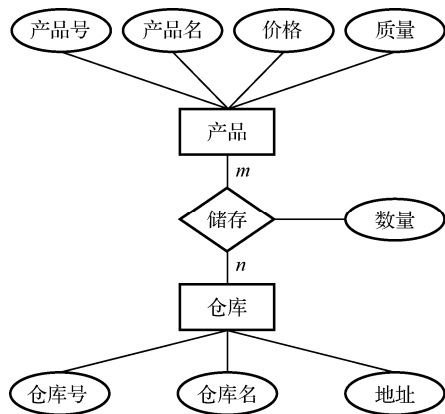


图 1-8 产品储存系统概念模型

1.5.4 常用的数据模型

概念模型完成了现实世界到信息世界的抽象，接下来要做的就是将概念模型进一步抽象为逻辑模型（数据模型）。数据库设计中，数据模型的选择至关重要，数据模型的好坏直接影响数据库的性能，因此数据模型的选择是设计数据库的一项首要任务。在数据库技术发展过程中产生了 3 种主要的数据模型：层次模型（Hierarchical Model）、网状模型

(Network Model) 和关系模型 (Relational Model)。这 3 种数据模型的根本区别在于数据结构不同, 即数据之间联系的表示方式不同。

- 层次模型用“树结构”来表示数据之间的联系。
- 网状模型用“图结构”来表示数据之间的联系。
- 关系模型用“二维表”来表示数据之间的联系。

1. 层次模型

层次模型是数据库系统中最早出现的数据模型, 采用层次模型的数据库典型代表是 IBM 公司的 IMS (Information Management System, 数据库管理系统)。

现实世界中, 许多实体之间的联系都表现出一种很自然的层次关系, 如家族关系、行政机构等。层次模型用一棵“有向树”的数据结构来表示各类实体, 以及实体间的联系。在树中, 每个结点表示一个记录类型, 结点间的连线(或边)表示记录类型间的关系。每个记录类型可包含若干个字段, 记录类型描述的是实体, 字段描述实体的属性, 各个记录类型及其字段都必须命名。如果要存取某个记录型的记录, 可以从根结点起, 按照有向树层次向下查找。

如图 1-9 所示, 结点 A 为根结点, D、E、F 为叶结点, B、C 为兄弟结点。

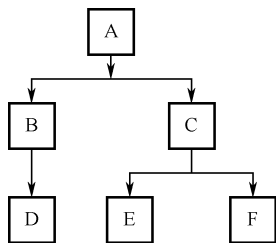


图 1-9 有向树

1) 层次模型的特征

- (1) 有且仅有一个结点没有双亲, 该结点就是根结点。
- (2) 根以外的其他结点有且仅有一个双亲结点, 这就使层次数据库系统只能直接处理一对多的实体关系。
- (3) 任何一个给定的记录值只有按其路径查看时, 才能显示其全部意义, 没有一个子女记录值能够脱离双亲记录值而独立存在。

2) 层次模型的主要优点

- (1) 比较简单, 只需很少几条命令就能操纵数据库, 比较容易使用。
- (2) 结构清晰, 结点间联系简单, 只要知道每个结点的双亲结点, 就可知道整个模型结构。现实世界中许多实体间的联系本来就呈现出一种很自然的层次关系。
- (3) 它提供了良好的数据完整性支持。

3) 层次模型的主要缺点

- (1) 不能直接表示两个以上的实体型间的复杂联系和实体型间的多对多联系, 只能通过引入冗余数据或创建虚拟结点的方法来解决, 易产生不一致性。
- (2) 对数据插入和删除的操作限制太多。
- (3) 查询子女结点必须通过双亲结点。

2. 网状模型

现实中事物之间的联系更多是非层次关系，用层次模型表示这种关系很不直观，网状模型则克服了这个弊病，可以清晰表示这种非层次关系。20 世纪 70 年代，数据系统语言研究会 CODASYL (Conference On Data System Language) 下属的数据库任务组 DBTG (Data Base Task Group) 提出了一个系统方案，即 DBTG 系统，也称 CODASYL 系统，成为网状模型的代表。

在数据库中，把满足以下两个条件的基本层次联系的集合称为网状模型。

- 有一个以上的结点没有双亲。
- 至少有一个结点可以有多个双亲。

即允许两个或两个以上的结点没有双亲结点，允许某个结点有多个双亲结点，此时有向树变成了有向图，该有向图描述了网状模型。网状模型中每个结点表示一个记录型（实体），每个记录型可包含若干个字段（实体的属性），结点间的连线表示记录类型（实体）间的父子关系。

1) 网状模型的优点

- (1) 能更为直接地描述客观世界，可表示实体间的多种复杂联系。
- (2) 具有良好的性能和存储效率。

2) 网状模型的缺点

- (1) 结构复杂，其数据定义语言极其复杂。
- (2) 数据独立性差，由于实体间的联系本质上是通过存取路径表示的，因此应用程序在访问数据时要指定存取路径。

3. 关系模型

关系模型是发展较晚的一种模型，1970 年美国 IBM 公司研究员 E.F.Codd 发表了题为《大型共享数据库的数据关系模型》(A Relation Model of Data for Large Shared Data Banks) 的论文，首次提出了数据库系统的关系模型理论。他在文中解释了关系模型，定义了关系代数运算，研究了数据的函数相关性，定义了关系的第三范式，开创了数据库的关系方法和数据规范化理论的研究。关系模型源于数学，他把数据看成一张二维表中的元素，这个二维表就是关系。在关系模型中，用关系（数据表格）表示实体及实体之间的联系。

通俗地讲，关系就是二维表，表格中的每一行称为一个元组，也就是一条记录，记录中的每一列是一个属性值集，列名就是属性名，这与前文所提的实体属性或记录的字段意义相当。关系数据库中，记录的集合构成关系，记录值之间不再通过指针联系，关系或记录之间的联系靠连接字段值来处理，理解关系和连接字段的思想对于理解关系数据库至关重要。

20 世纪 80 年代以来，计算机厂商新推出的数据库管理系统几乎都支持关系模型，非关系系统的产品也都加上了关系接口。数据库领域当前的研究工作都是以关系方法为基础的。关系数据库已成为目前应用最广泛的数据库系统，如 Oracle、Informix、Sybase、SQL Server 等都是关系数据库系统。

1.6 小结

本章介绍了数据库的基本概念，概述了数据管理技术的产生与发展，简述了数据库系统的特点，数据库系统的三级模式和两级映射的系统结构，确保了数据库系统有较高的逻辑独立性和物理独立性。

数据模型分为三层：概念模型、逻辑模型和物理模型，它的三要素是数据结构、数据操作和数据的完整性约束。

最常用的概念模型的表示方法是实体-联系模型，用 E-R 图来描述。3 种常用的数据模型：层次模型、网状模型和关系模型。

习题 1

- 1-1 简述数据、数据库、数据库管理系统、数据库系统的概念。
- 1-2 论述文件系统与数据库系统的区别与联系。
- 1-3 简述数据库系统的特点。
- 1-4 解释概念模型中的术语：实体、实体型、实体集。
- 1-5 试述数据与程序的物理独立性。