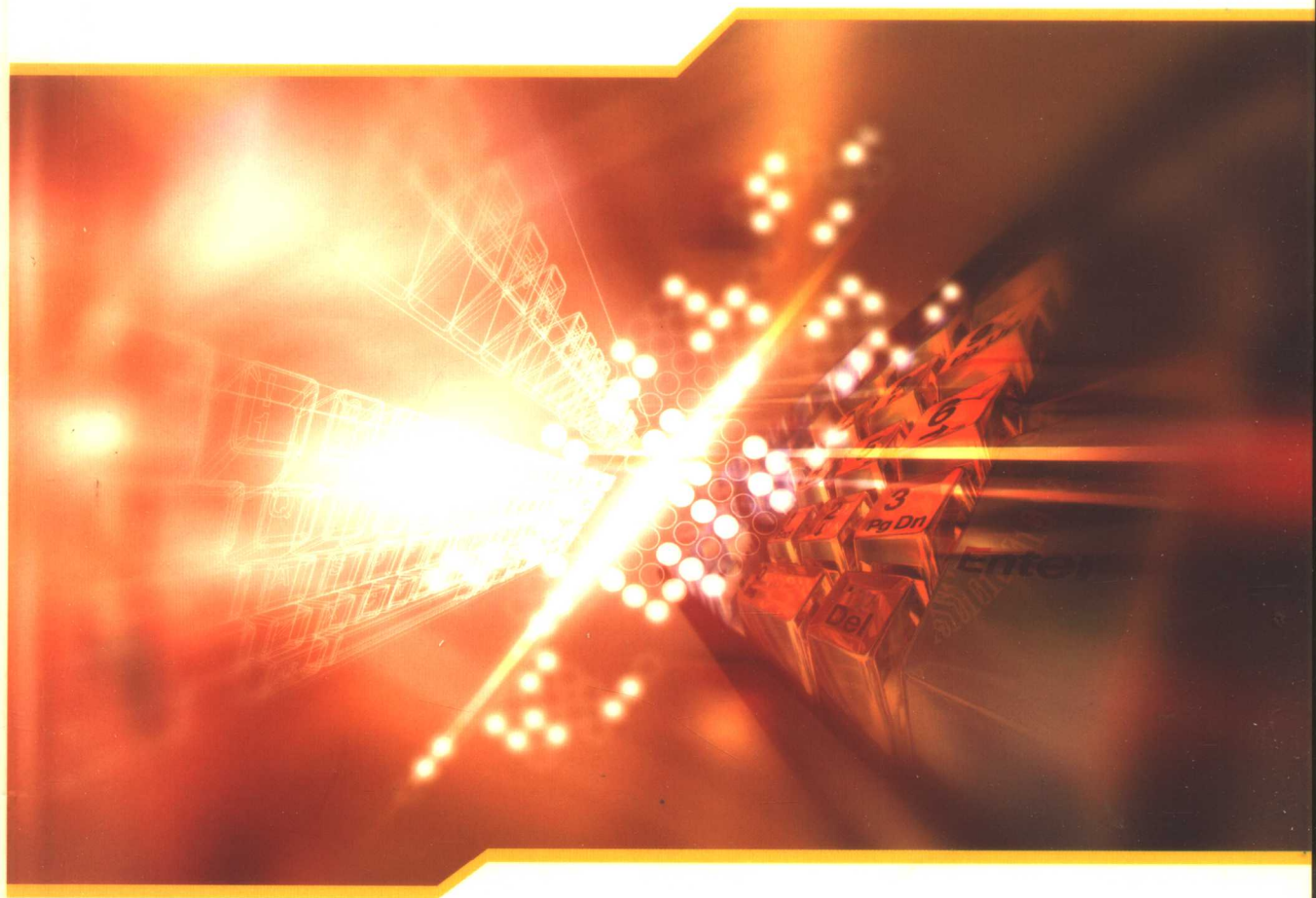


数据库应用开发 综合实例教程

SHUJUKU YINGYONG KAIFA ZONGHE SHILI JIAOCHENG

孔璐 袁珏 吴志坚 顾洪 编著



国防工业出版社

National Defense Industry Press

数据库应用开发 综合实例教程

孔璐 袁珏 吴志坚 顾洪 编著

国际工业出版社

·北京·

内 容 简 介

本书在系统介绍数据库理论的基础上,以 SQL Server、Access、Visual FoxPro 这三种通用数据库管理系统(DBMS)为开发工具,以多个实用的数据库应用系统为实例,综合讲述在多种通用的 DBMS 环境下数据库应用系统创建的各个过程,包括目标设计、系统开发背景、系统功能分析、系统功能模块设计、数据库需求分析、数据库概念结构设计、数据库逻辑结构设计、数据库结构实现、各个功能模块的创建、系统的编译和发行等。各实例按照由浅入深、由易到难的顺序编排,内容涵盖了信息管理系统、软件工程、ASP 与数据库编程等技术。

全书内容详实,讲解细致,图文并茂,书中选取的应用系统实例都是在实际开发环境中编译通过的,实用性强。本书既可作为各本、专科院校数据库课程的教材,也可供相关专业的工程技术人员及数据库应用开发人员学习参考。

图书在版编目(CIP)数据

数据库应用开发综合实例教程 / 孔璐等编著. —北京:
国防工业出版社, 2008. 1
ISBN 978 - 7 - 118 - 05348 - 7

I. 数... II. 孔... III. 关系数据库 - 数据库管理系统 -
教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2007)第 134679 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100044)

新艺印刷厂印刷

新华书店经售

*

开本 787 × 1092 1/16 印张 19 字数 533 千字

2008 年 1 月第 1 版第 1 次印刷 印数 1—4000 册 定价 33.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

前 言

数据库技术是计算机科学的一个重要分支,是当代信息处理中不可缺少的先进技术,它广泛应用于各级单位的信息管理系统、智能决策支撑系统、网络环境下的管理信息系统(MIS)、系统资源的管理与共享及信息高速公路,数据库已成为本专科院校、科研院所计算机专业人员以及各类软件开发人员必须掌握的基础技术。当前数据库应用开发大多同时面向多种数据库管理系统环境,具有数据多源性与软件多样性的特点,而现有数据库应用开发实例教程基本只局限于一种数据库管理系统,再加上数据库技术实践性很强,发展日新月异,所以结合多种通用数据库管理系统综合讲述数据库开发应用实例的书籍也应不断更新,才能更加适应社会发展对数据库技术的需求。基于以上实际情况编写了本书,并且在编写过程中注重以下几个方面:

(1) 本书根据当前数据库应用开发中 DBMS 多源性与软件多样性的特点,使读者能够在学习数据库基础理论的基础上,同时掌握综合利用几种通用 DBMS 进行应用开发的能力,针对性强,内容丰富。

(2) 各实例紧贴行业特点,需求分析详尽,着重讲解数据库设计与开发,重点突出。

(3) 实例部分逐层深入,由一个简单的系统开始、难度逐渐加深,全书各章穿插着许多数据库开发的技巧。

(4) 突破只讲编程技术不讲开发思路的桎梏,在书中渗透大量软件工程的思想,希望通过每个实例系统的开发,提供给读者一些软件设计的理念。

(5) 各实例模块清晰,代码规范,具有很好的可移植性,通用性强,同时关键步骤讲解详细,学习目标明确。

(6) 有完整源代码,实例部分还配有针对性的实训练习,可用于学生实习,也可供计算机相关专业教师指导学生毕业设计之用,实用性强。

本书注重培养读者的实际开发应用能力,选材新颖,由浅到深,内容全面、完整,是一本易懂、实用、极具参考价值的实例教程,既可作为各本、专科院校数据

库课程的教材,也可供相关专业的工程技术人员及数据库应用开发人员学习参考。

本书在编写过程中参考了大量国内外同行的著作与文章,在此表示感谢。鉴于作者知识水平有限,书中难免存在疏漏之处,欢迎广大读者批评指正。

编者
2007年9月

目 录

第 1 篇 数据库理论

第 1 章 数据库系统导论 / 1

- 1.1 数据与信息 / 1
 - 1.1.1 数据、信息与数据处理 / 1
 - 1.1.2 计算机信息系统 / 2
- 1.2 数据管理技术的发展过程 / 3
 - 1.2.1 人工管理阶段 / 3
 - 1.2.2 文件管理阶段 / 3
 - 1.2.3 数据库系统阶段 / 4
- 1.3 数据模型 / 5
 - 1.3.1 数据模型的定义 / 5
 - 1.3.2 关系模型 / 5
 - 1.3.3 E-R 数据模型 / 6
- 1.4 数据库系统结构 / 9
 - 1.4.1 数据库系统的组成 / 9
 - 1.4.2 数据库系统模式结构 / 10
 - 1.4.3 数据库管理系统 / 12
 - 1.4.4 数据库语言 / 14
 - 1.4.5 数据库管理员 / 14
 - 1.4.6 数据字典 / 14
 - 1.4.7 数据库系统的工作流程 / 15
- 1.5 习题 / 16

第 2 章 关系数据库 / 17

- 2.1 关系模型的基本概念 / 17
 - 2.1.1 笛卡儿乘积 / 17
 - 2.1.2 关系的基本概念 / 18
- 2.2 关系代数 / 19
 - 2.2.1 传统的集合运算 / 20
 - 2.2.2 专门的关系运算 / 20

2.3 习题 / 25

第 3 章 关系数据库设计理论 / 28

- 3.1 数据库设计中的问题 / 28
- 3.2 函数依赖 / 30
 - 3.2.1 属性间的联系 / 30
 - 3.2.2 函数的依赖关系 / 31
 - 3.2.3 关键字 / 32
- 3.3 规范化理论 / 33
 - 3.3.1 第一范式 / 33
 - 3.3.2 第二范式 / 34
 - 3.3.3 第三范式 / 35
 - 3.3.4 BCNF / 36
 - 3.3.5 规范化小结 / 37
- 3.4 习题 / 38

第 4 章 数据库应用系统设计 / 39

- 4.1 数据库应用系统设计概述 / 39
 - 4.1.1 数据库应用系统设计的内容和特点 / 39
 - 4.1.2 数据库系统设计步骤 / 40
- 4.2 需求分析 / 41
- 4.3 概念模型设计 / 42
 - 4.3.1 设计局部 E-R 模型 / 42
 - 4.3.2 将各局部 E-R 模型综合成总体 E-R 模型 / 43
- 4.4 逻辑结构设计 / 44
 - 4.4.1 概念模型向关系模型转换 / 44
 - 4.4.2 数据模型的优化 / 47

4.4.3	设计用户子模式 / 48
4.5	数据库物理设计 / 48
4.6	数据库实施 / 49
4.7	数据库运行与维护 / 49
4.8	数据库应用系统的功能和 界面设计 / 50
4.8.1	系统功能结构设计 / 50
4.8.2	数据库应用系统界面设计 / 51
4.9	习题 / 52
第5章	结构化查询语言——SQL / 54
5.1	SQL 概述 / 54
5.2	数据定义 / 55
5.2.1	SQL 的数据类型 / 55
5.2.2	基本表 / 56
5.2.3	索引 / 58
5.3	查询 / 59

5.3.1	Select 语句 / 59
5.3.2	单表查询 / 62
5.3.3	连接查询 / 69
5.3.4	嵌套查询 / 73
5.3.5	集合查询 / 78
5.4	SQL 的数据更新 / 79
5.4.1	用 INSERT 添加行 / 79
5.4.2	使用 UPDATE 修改行 / 80
5.4.3	使用 DELETE 删除行 / 80
5.5	视图 / 81
5.5.1	定义视图 / 81
5.5.2	视图的使用 / 83
5.5.3	更新视图 / 84
5.6	数据控制 / 85
5.6.1	授权 / 85
5.6.2	收回权限 / 86
5.7	习题 / 87

第2篇 初级实例

第6章 Visual FoxPro 数据库管理系统 / 88

6.1	Visual FoxPro 基础知识 / 88
6.1.1	常量与变量 / 88
6.1.2	表达式 / 92
6.1.3	常用函数 / 94
6.2	Visual FoxPro 数据库及其操作 / 98
6.2.1	Visual FoxPro 数据库及其 建立 / 98
6.2.2	建立数据库表 / 100
6.2.3	表的基本操作 / 102
6.2.4	索引 / 105
6.2.5	数据完整性 / 109
6.2.6	自由表 / 111
6.2.7	记录的统计与计算 / 112
6.2.8	排序 / 112
6.3	Visual FoxPro 程序设计 / 113
6.3.1	程序与程序文件 / 113

6.3.2	程序的基本结构 / 116
-------	---------------

6.4	习题 / 122
-----	----------

第7章 用 Visual FoxPro 6.0 开发一个通用的教学管理系统 / 126

7.1	需求分析 / 126
7.2	系统分析 / 126
7.2.1	数据库设计 / 126
7.2.2	系统功能设计 / 129
7.2.3	模块设计 / 130
7.3	建立系统项目和数据库 / 130
7.3.1	建立系统项目 / 130
7.3.2	建立数据库 / 131
7.4	系统中主要表单的制作 / 132
7.4.1	系统表单的设计 / 132
7.4.2	系统启动表单的制作 / 133
7.4.3	建立系统登录表单 / 134
7.4.4	制作系统主表单 / 137
7.4.5	建立单位设置表单 / 139

- 7.4.6 基本信息管理表单的制作 / 146
- 7.4.7 查询表单的制作 / 148
- 7.5 打开数据库程序 / 152
- 7.6 程序关闭设计 / 154
- 7.7 系统主程序设计 / 155
 - 7.7.1 系统主程序流程设计 / 155
 - 7.7.2 系统主程序代码 / 155
 - 7.7.3 主程序的编译与运行检验 / 157
- 7.8 应用系统的连接与可执行文件创建 / 157
- 7.9 用 Visual FoxPro 6.0 制作一个通用高校人力资源管理系统 / 158

第8章 Access 与 ASP 技术 / 160

- 8.1 Access 简介 / 160
 - 8.1.1 Access 窗口组成 / 160
 - 8.1.2 数据库窗口的使用 / 161
- 8.2 Access 数据类型 / 162
 - 8.2.1 文本字段 / 162
 - 8.2.2 备注字段 / 162
 - 8.2.3 数字字段 / 162
 - 8.2.4 日期/时间字段 / 163
 - 8.2.5 货币字段 / 163
 - 8.2.6 自动编号字段 / 163
 - 8.2.7 是/否字段 / 163
 - 8.2.8 OLE 字段 / 163
 - 8.2.9 超级链接字段 / 163
- 8.3 创建 Access 数据库 / 164
 - 8.3.1 新建数据库的目的和规划数据库表 / 164
 - 8.3.2 确定表的字段和具有唯一性的字段 / 165
 - 8.3.3 确定表之间的关系 / 165
 - 8.3.4 建立一个空的数据库 / 165
 - 8.3.5 数据库中的主要对象及关系 / 166
 - 8.3.6 使用向导创建数据库 / 166
 - 8.3.7 不使用向导创建数据库 / 168

- 8.4 ASP 简介 / 168
 - 8.4.1 使用 ASP 访问数据库 / 169
 - 8.4.2 ASP 定义 / 169
 - 8.4.3 ASP 中的运算符 / 171
 - 8.4.4 条件式 / 172
 - 8.4.5 循环 / 174
 - 8.4.6 ASP 常用内建对象 / 177
- 8.5 习题 / 177

第9章 Access & ASP 网络测试系统的设计与实现 / 178

- 9.1 系统需求分析 / 178
 - 9.1.1 了解系统需求 / 178
 - 9.1.2 分析用户需求 / 178
- 9.2 数据库设计 / 179
 - 9.2.1 数据库逻辑设计 / 179
 - 9.2.2 数据库物理设计 / 179
- 9.3 系统功能设计 / 180
- 9.4 数据库的链接和访问 / 180
 - 9.4.1 数据库驱动程序 / 180
 - 9.4.2 Connection 对象 / 181
 - 9.4.3 Recordset 对象 / 181
- 9.5 登录界面的设计与实现 / 182
 - 9.5.1 登录界面的设计 / 182
 - 9.5.2 登录界面的代码实现 / 183
- 9.6 选题界面的设计与实现 / 185
 - 9.6.1 选题界面的设计 / 185
 - 9.6.2 选题界面的代码实现 / 186
- 9.7 测试答题界面的设计与实现 / 187
 - 9.7.1 测试答题界面设计 / 188
 - 9.7.2 测试答题界面代码实现 / 189
- 9.8 自动判卷功能的设计与实现 / 193
 - 9.8.1 自动判卷界面设计 / 193
 - 9.8.2 自动判卷代码实现 / 194
- 9.9 实训练习 / 198
 - 9.9.1 完善本系统功能 / 198
 - 9.9.2 用 ASP 结合 Access 制作网上留言管理系统 / 198

第3篇 高级实例

第10章 SQL Server 数据库管理系统 / 199

10.1 SQL Server 概述 / 199

10.1.1 SQL Server 安装与配置 / 199

10.1.2 SQL Server 工具和实用
程序 / 203

10.2 SQL Server 管理 / 204

10.2.1 SQL Server 系统结构 / 204

10.2.2 SQL Server 基本操作 / 206

10.2.3 SQL Server 备份 / 210

10.2.4 SQL Server 数据库
恢复 / 213

10.2.5 SQL Server 安全性 / 214

10.3 习题 / 217

第11章 利用 SQL Server 开发院校管理
信息系统 / 218

11.1 系统设计 / 218

11.2 数据库设计 / 219

11.2.1 概念模型设计 / 219

11.2.2 逻辑模型设计 / 220

11.2.3 数据库实施 / 221

11.3 系统功能设计 / 223

11.3.1 创建工程 / 223

11.3.2 开始窗体 / 224

11.3.3 登录窗体 / 226

11.4 管理功能设计 / 230

11.4.1 系统登录管理 / 230

11.4.2 学生信息管理 / 234

11.4.3 教师信息管理 / 243

11.4.4 课程信息管理 / 251

11.4.5 成绩信息管理 / 258

11.4.6 选课信息管理 / 262

11.4.7 班级信息管理 / 270

11.4.8 单位信息管理 / 276

11.5 查询功能设计 / 281

11.5.1 课程教学信息查询 / 281

11.5.2 课程成绩查询 / 284

11.6 系统平台设计与测试 / 287

11.6.1 系统平台设计 / 287

11.6.2 系统测试 / 290

11.7 实训练习 / 293

参考文献 / 295

第 1 篇 数据库理论

第 1 章 数据库系统导论

本章重点

本章重点介绍数据库系统的基本概念，使读者了解数据库系统的特点、结构和组成。

学习目标

- (1) 了解数据管理技术的发展过程。
- (2) 了解数据模型的概念。
- (3) 了解数据库的系统结构。

1.1 数据与信息

1.1.1 数据、信息与数据处理

人类社会发展到今天，数据、信息已是人们在日常生活中必不可少的重要资源，如学校的学籍管理信息、企业的生产管理和产品销售数据、银行账目等。但是，在不同场合下，人们对数据、信息的概念理解有所不同。一般认为信息（Information）是数据（Data）的含义。数据和信息这两个术语有时可以混用，例如，数据处理也可称为信息处理；有时则必须分清，例如，信息系统就不能称为数据系统。

那么，什么是数据呢？数据就是对客观事物的一种反映或描述，例如，某人的姓名、性别、出生日期等，就是反映此人基本状况的数据。数据的形式可以是语言文字、数字、各种符号，还可以是图形、图像、声音、视频等。数据可以记录在纸上，也可以记录在磁介质、光介质和半导体存储器里。

孤立的数据是没有用的，必须将其转换成信息。信息是人围绕某个目的从相关数据中提取的有价值的意义，例如，从出生年月这个数据，可以得到此人的年龄、是否可以入学或是退休等相关信息。一般说来，信息是用数据来表示的，所以，二者形式基本相同，但概念有区别，并非任何数据都能表示信息，信息是经过消化理解的有意义的数。据。

数据处理是将数据转换成信息的过程。这个过程可以包括对数据的收集、存储、加工、分类、检索、传播等一系列活动。其目的是从大量的、已有的数据出发，根据事物之间的固有联系和运动规律，通过分析归纳、演绎推导等手段，提取出对人们有价值、有意义的信息，作为决策和行动的依。据。由此可见，信息是一种被加工成特定形式的数据，这种数据形式对于数据接受者来说是有意义的。对数据的加工可以相对简单，也可以相当复杂。简单加工包括组织、编码、分类、排序等；复杂加工则是使用统计学方法、数学模型等对数据进行深层次的处理。可以用图 1.1 简单地描述数据和信息的关。系。



图 1.1 数据和信息的关系

当两个或两个以上的数据处理过程前后相继时,前一过程称为预处理,预处理的输出作为二次数据,成为后面处理过程的输入,这时信息和数据的概念就发生了交叉,表现出相对性,如图 1.2 所示。

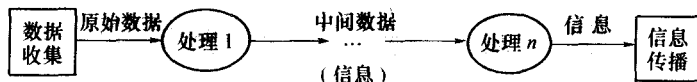


图 1.2 数据与信息相对性

1.1.2 计算机信息系统

信息的重要性已经普遍为人们所接受。在目前这样一个信息时代,人类的生产活动和日常生活的质量越来越依赖于信息,信息的利用程度已经成为各种组织机构运作效率和成败的关键因素,甚至还对个人事业及生活质量构成很大影响。要能够准确、及时地得到相关信息,就要有相当的数据处理能力,而计算机具备的强大的数据存储和加工能力,使其成为信息处理的首选工具。计算机信息系统已成为计算机技术最大的一个应用领域。

计算机信息系统是一种为了某个目标而建立的,由人员、设备、程序和数据集合构成的统一整体。它的主要功能是管理并处理相关数据,提供信息,以支持一个组织机构的运行、管理和决策。

计算机信息系统的好坏取决与它能否及时准确地为用户提供所需要的信息。在同一组织机构中,不同阶层的管理人员可以有不同的信息需求,系统应能针对各个层次需求,通过计算机实现信息支持,达到辅助管理的目的。

计算机信息系统的用户并不局限于某一个组织,它也可以是一种公用性信息服务系统,例如,美国国际航空公司的飞机航班查询订票系统,可以在全球范围内为用户提供信息查询和订票服务。

根据系统功能的不同,计算机信息系统大体可以分为以下几类:

(1) 事务处理系统(EDP),其目的是用计算机代替繁杂的手工事务处理工作,从而提高数据处理的准确性、及时性,节省人力,提高工作效率。例如,会计核算软件,就是对会计的簿记事务进行常规处理、提供数据查询、会计报表等功能,使会计部门的日常工作自动化。

(2) 管理信息系统(MIS),直接用于支持组织内部各层次的管理工作,提供业务范围内的各种数据以及对这些数据的分析结果,并可从组织的全局出发,实现数据共享,提供分析、计划、预测、控制等方面的综合信息。其目的是发挥系统的综合效益,提高管理水平。一般来说,这类系统是分层次的,一个大系统往往有若干个子系统组成。例如,一个企业的管理信息系统由财务管理系统、物资管理系统、生产管理系统、人事管理系统、销售管理系统、仓库管理系统等组成。

(3) 决策支持系统(DSS),为决策过程提供有效的信息和辅助决策手段,其主要目的是帮助决策者提高决策的科学性和有效性。这类系统必须积累大量的数据、案例、方法、模型,更进一步还可以利用知识库和专家系统。决策支持系统的服务对象是面向某种决策问题的管理人员,它协助决策者在求解问题的过程中方便地检索出相关数据,对多种方案进行选择。

(4) 查询服务系统(QSS),其目标是向公众提供信息服务。这类系统涉及面广,可以查询火车、

飞机时刻表,订票;可以查询商品信息,网上购物;可以进行学术文献检索等。在政治、金融、商业、企业、军事、地理、教育等方面都有许多可供查询的系统。

以上分类主要是按功能进行的,现代的计算机信息系统已经向着综合性、集成化、智能化方向发展。

1.2 数据管理技术的发展过程

信息系统是一类数据的密集型应用,不论哪一类信息系统,都建立在大量数据事实的基础之上,管理这种大量的、持久的、共享的数据是这类计算机应用面临的共同问题。

与其他技术的发展一样,计算机数据管理也经历了由低级到高级的发展过程。计算机数据管理随着计算机硬件(主要是外部存储设备)、软件技术和计算机应用范围的发展而不断发展,大致经历了3个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。

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。当时在硬件方面,只有卡片、纸带、磁带,而没有像硬盘这类可以快速、随机存取的外部存储设备;软件方面,没有专门的数据管理软件,数据由程序自行携带。数据和程序之间的关系如图1.3所示。

人工管理阶段数据管理的特点是:数据与程序不具有独立性和数据冗余。

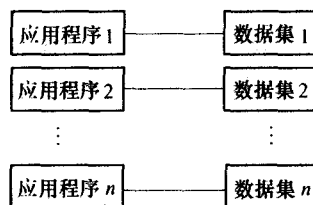


图 1.3 人工管理阶段应用程序与数据之间的对应关系

1.2.2 文件管理阶段

20世纪50年代后期至20世纪60年代中后期,计算机开始大量用于数据处理工作,大量的数据存储、检索和维护成为紧迫的需求。在硬件方面,可直接存取的磁鼓、磁盘成为联机的主要外部存储设备。在软件方面,出现了高级语言和操作系统,操作系统中的文件系统就是专门管理外部存储设备的数据管理软件。

在这一阶段,程序和数据间有了一定的独立性,程序和数据分开存储,数据文件可以长期保存在硬盘上多次存取。在文件系统的支持下,数据的逻辑结构与物理结构之间也有了一定的独立性,程序员可以集中精力在数据处理的算法上,而不必考虑数据存储的具体细节。这个阶段的数据还是面向应用程序的,数据文件基本上与各自的应用程序相对应,它们之间的关系如图1.4所示。

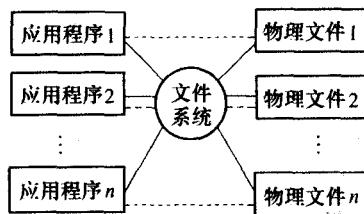


图 1.4 文件管理阶段应用程序与数据之间的对应关系

文件管理阶段存在以下5方面的问题:

- (1) 应用程序开发效率低。
- (2) 文件的设计很难满足多种应用程序的不同要求,数据的冗余是不可避免的。
- (3) 数据独立性差。
- (4) 文件系统一般不支持对文件的并发访问。
- (5) 没有对数据的统一管理。

这些问题阻碍了数据处理技术的发展,不能满足日益增长的信息需求,应用需求和计算机技术

的发展促使人们研究发展新的数据管理技术，数据库技术也就应运而生了。

1.2.3 数据库系统阶段

从 20 世纪 60 年代后期开始，计算机应用于管理的规模更加庞大，需要计算机管理的数据急剧增长，并且对数据共享的要求与日俱增。大容量磁盘系统的采用，使计算机联机存取大量数据成为可能；软件价格上升，硬件价格相对下降，使独立开发系统和维护软件成本增加。文件系统的管理方法已无法满足要求，为了解决独立性问题，实现数据统一管理，最大限度地实现数据共享，必须发展数据库技术。

数据库是通用化的相关数据集合，它不仅包括数据本身，而且包括相关数据之间的联系。数据库中的数据通常是整个信息系统全部数据的汇集，面向所有合法用户。其数据结构独立于使用数据的程序，数据库的建立、使用和维护等操作由专门的软件系统，即数据库管理系统（DBMS）统一进行。

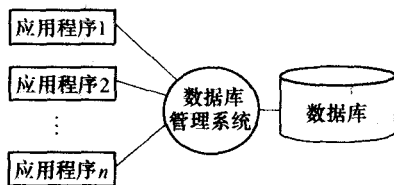


图 1.5 数据库系统数据与程序的关系

现在，数据库已成为各类信息系统的核心。在数据库管理系统支持下，数据与程序的关系如图 1.5 所示。

数据库的主要特点是：

1. 从全局观点组织数据，实现数据共享，减少数据冗余

建立数据库时，应当以面向全局的观点组织库中的数据，而不能像文件系统那样，仅仅考虑某一部门的局部应用。数据库中存放的是全组织（如企业）通用化的综合性的数据，某一类应用通常仅使用总体数据的子集，这样才能发挥数据共享的优势。

理论上，数据库中是没有冗余的，但实际上，许多数据库系统为了改善访问时间或者简化存取方法，允许存在一定的数据冗余，只是要去掉有害的或不必要的冗余。

2. 维护数据的独立性

所谓数据的独立性就是应用程序不必因为数据的存储结构的变化而修改，即应用程序与数据的结构之间不存在依赖关系，这是数据库系统所努力追求实现的一个目标，也是数据库系统结构之所以复杂的一个重要原因。数据库系统的数据独立性分为两级：

（1）物理独立性，即数据库的物理结构的变化（如物理设备的更换、物理位置的变化、存取方法的改变等）不影响数据库的逻辑结构，从而也就影响不到应用程序，不会导致应用程序的修改。

（2）逻辑独立性。数据库逻辑结构的变化（如数据定义的修改，新数据类型的增加，数据间联系的变更等）不致影响到用户原有应用程序的修改。

这两种独立性统称为数据独立性。数据独立性的目的就是要使应用程序尽可能不受数据的影响。这两种数据独立性是靠数据库管理系统来实现的，大大减轻了程序员的负担。

3. 加强对数据的保护

使用数据的用户非常关心数据的安全性和可靠性。数据的安全可靠是一个数据库是否具有实用价值的关键问题。一般较完整的数据库系统对数据的保护主要有以下 4 个方面的内容：

（1）安全保密控制，主要是数据的保密性控制，即保护库中的数据以防止不合法的使用。

（2）完整性控制。数据的完整性包括数据的正确性、有效性和一致性。一个数值型数据，如果包含了诸如字母、特殊符号，显然是不正确的，就是失去完整性的例子。此外，形式正确的数据也并非一定是有效的，例如，月份必须是 1~12 的正整数，否则便是无效的。而表示同一事实的两个数据项之间的不一致则是缺乏完整性的例子。数据库系统对数据的使用是集中控制的，因而有利于控制数据的完整性。

(3) 并发控制。数据的不相容性主要是由于数据的共享而引起的, 不同用户同时使用数据库可能引起对数据的干扰。例如, 当某个用户需要修改某一数据时, 他必须先读出这个数据, 修改后再写回去, 如果在这个用户把被修改的数据写回数据库之前, 别的用户再去读出这个正在被修改的数据, 就会引起错误。数据库管理系统的并发控制功能将排除和避免这种错误的发生, 保证数据的正确性。

(4) 数据库恢复。数据库在运行过程中很难保证不受破坏, 全局性的和局部性的破坏随时都可能发生, 因为硬件和软件故障以及用户操作上的错误都是不可避免的。为了防止由于这些故障而破坏数据的正确性, 数据库系统必须采取适当的措施来保证数据的正确性。

4. 对数据实行集中控制

由前述可知, 文件系统对数据的管理处于一种分散的状态中, 无法按照一种统一的方法来控制、维护和管理数据。在数据库中克服了文件系统的缺点, 对数据进行集中控制和管理, 一个组织或一个企业, 可以把它们所有要涉及到的数据都组织到一个数据库中, 也就是说数据库是“集成”的。但这里所说的集成并不是把数据简单地集中起来, 而是对数据进行结构上的重组, 使其能适应所有用户的要求。

1.3 数据模型

1.3.1 数据模型的定义

数据管理的一个核心问题, 就是研究如何表示和处理实体之间的联系。表示实体及实体之间联系的数据库的数据结构称为数据模型。换言之, 把数据库系统中所包含的所有记录, 按照它们之间的联系组合在一起, 构成一个整体, 这个整体的结构就称为数据库的数据模型。数据模型是数据库中数据全局逻辑结构的描述。

数据模型包括3个方面, 即模型结构、数据操作和完整性规则。其中, 模型结构是数据模型最基本的部分, 它将确定数据库的逻辑结构, 是对系统静态特性的描述。数据操作提供对数据库的操纵手段, 主要有检索和更新两大类操作。数据操作是对系统动态性的描述。完整性规则则是对数据库有效状态的约束。

数据库应用领域中的数据模型有传统的层次模型、网状模型和关系模型, 面向用户数据库设计的E-R模型和面向对象模型。本书主要介绍目前使用最广泛的关系模型和E-R模型。

1.3.2 关系模型

在层次模型和网状模型的数据库系统被开发出来之后, 人们发现层次模型和网状模型缺乏充实的理论基础, 难以开展深入的理论研究, 于是开始寻求具有较充实的理论基础的数据模型。在日常生活和工作中, 人们往往习惯于用表格的形式来组织管理和表示数据信息。例如, 图1.6就是一个大学的系情况表, 表中各系的主要信息一目了然。这样一张表可称为一个关系, 表中的每一列为关系中的一个属性(数据项), 表中的每一行为关系中的一个元组(记录)。在稍后的介绍中将给出关系严格的数学定义。实际上一个关系就相当于一个记录类型, 而这个关系的全体元组, 就是这个记录的所有值。

系号	系名	系主任	专业数
1	气象	王雨	2
2	控测	张和平	2
3	计算机	李一军	2
4	自动化	刘桂林	3
5	管理	丁娜	3
6	委培	郑国涛	5

图1.6 系情况表

关系模型是用二维表的形式来表示实体和实体间联系的数据模型。在层次模型和网状模型中, 文件中存放的是数据, 各文件之间的联系是通过指针来实现的。而在关系模型里, 描述各个实体之

间联系的信息是隐含在关系中的。

下面举例说明关系模型。设图 1.7 中的各关系和图 1.6 中的关系一起构成了某大学的教务数据库。

教师情况表						学生情况表				
职工号	姓名	年龄	系号	职称	工资	学号	姓名	年龄	系号	班级
...	...	...	...	...	...	30101	赵玉龙	19	3	301
013	张雨林	38	2	副教授	800	30102	王东	20	3	301
014	李娜	52	2	教授	1000	...	...	...	...	...
...	...	...	...	...	...	40101	汪放	21	4	401
054	丁旭东	23	5	助教	500	40102	陈立	19	4	401
...	...	...	...	...	...	...	...	...	...	...

教师任课情况表			课程情况表			学生成绩表		
职工号	课程号	班级	课程号	课程名	课时	学号	课程号	分数
...	...	...	J101	高等数学	120	30101	J101	90
013	J101	301	J102	线性代数	80	30102	J101	78
014	J102	301	J103	C 语言	72	...	...	...
...	...	...	...	...	...	30101	J103	89
						...	...	...

图 1.7 学校教务数据库

这些关系组合在一起构成了一个数据库。由于关系不是孤立的，而是相互联系的（即隐含着实体之间的联系），各个关系建立联系的方式通常有以下两种：

- （1）两个关系通过定义相同的属性反映它们之间的联系。
- （2）定义第 3 个关系，把两个关系建立起联系。

例如，关系“教师任课情况表”中的属性“职工号”，取自关系“教师情况表”，而属性“课程号”，则取自关系“课程情况表”，通过“教师任课表”这个关系就使得关系“教师情况表”和关系“课程情况表”建立起了一种联系。

由此可见，关系模型是不同于层次模型和网状模型的，其本质的差异就在于，关系模型是通过关系中的数据而不是通过指针连接来表示两实体间的联系的。

关系是元组的集合。如果表格有 n 列，则称该关系为 n 元关系。关系模型中的关系应有如下特点：

- （1）关系中的每一列属性，都是不能再分的基本字段。
- （2）各列被指定一个相异的名字。
- （3）各行不允许重复。
- （4）行、列次序无关紧要。

关系模型与层次模型和网状模型相比具有许多优点，它的用户界面简单，又有严格的数学理论作基础，已成为目前最主要的数据模型。自 20 世纪 80 年代以来，对关系数据库的理论研究和系统研究受到了高度的重视，新推出的数据库管理系统几乎都支持关系模型。

1.3.3 E-R 数据模型

1. 基本概念

E-R 数据模型（Entity Relationship Data Model，实体联系数据模型），于 1976 年由 P.P.S.Chen 提出。E-R 数据模型最成功和最广泛的应用是作为数据库概念设计的数据模型。

作为描述概念世界，建立概念模型的实用工具，E-R 模型有 3 个与概念世界相对应的基本概念

(1) 实体 (Entity)。凡是可以互相区别而又可以被人们识别的事、物、概念等统统抽象为实体。在一个单位里，具有共性的一类实体可以划分为一个实体集 (Entity Set)。例如，学生小王、小李、……都是实体，但他们都是学生。为了便于描述，可以定义“学生”这样一个实体集，所有学生都是这个集合的成员。每个学生需要描述的内容是一样的，例如，姓名、学号、性别等，但每个学生在这些描述项目上所取的值不一定一样。因此，在 E-R 数据模型中也有型与值之分，实体集作为型来定义，而每个实体是它的实例或值。

(2) 属性 (Attribute)。实体一般具有若干特征，称为实体的属性，例如，学生具有姓名、学号等属性。每个属性都有其取值的范围，在 E-R 数据模型中称为值集 (Value Set)。值集实际上相当于域。在同一实体集中，每个实体的属性及其域是相同的，但可能取不同的值。在 E-R 数据模型中，属性可以是单域的简单属性，也可以是多域的组合属性。组合属性由简单属性和其他组合属性组成。组合属性中允许包含其他组合属性，这意味着属性可以是一个层次结构。

在 E-R 数据模型中，属性可以是单值的，也可以是多值的，例如，一个人所获的学位可能是多值的。当某个属性对某个实体不适用或属性值未知时，也可用空缺符 NULL。

能唯一标识实体的属性或属性组（这组属性的任何真子集无此性质）称为实体集的实体键。如果一个实体集有多个实体键存在，则可从中选一个最常用到的作为实体的主关键字。

(3) 联系 (Relationship)。实体之间会有各种关系，例如，学生实体与课程实体之间可有选课关系，人与人之间可能有领导关系、夫妻关系等。这种实体与实体间的关系可以抽象为联系。

如果在联系中注明实体在联系中的作用，则明显地丰富了联系的语义。

与传统的数据模型相比，E-R 数据模型在实体的联系方面提供了较多的语义。在二元联系中，E-R 数据模型把联系区分为一对一 (1:1)、一对多 (1:n) 和多对多 (m:n) 3 种，并在模型中明确地给出这些语义。

不但实体可以有属性，联系也可以有属性，例如，选课这种联系可以有成绩、选修时间等属性。现实世界中还有一种特殊的联系，这种联系代表实体间的所有关系，例如，职工与家属的联系。

2. E-R 图

用 E-R 数据模型对一个单位的模拟，称为一个单位的 E-R 数据模式。E-R 数据模式可以用非常直观的 E-R 图表示，这是 E-R 数据模型广为流行的重要原因之一。E-R 图中包括实体、属性、联系 3 种基本图素。约定实体用方框表示，属性用椭圆框表示，联系用菱形框表示，如图 1.8 所示，在其内部要填上实体名、属性名和联系名。

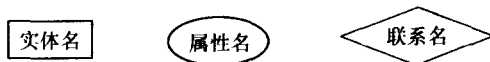


图 1.8 E-R 图的基本图素

具体画法是：把有联系的实体（方框）通过联系（菱形框）连接起来，注明联系方式，实体的属性（椭圆框）连到相应实体上。图 1.9 分别表示了两个实体间的 3 种不同的联系方式。

注意实体有属性，联系也可能有属性。例如学生与课程联系上的“成绩”，它既不是实体“学生”的属性也不是实体“课程”的属性，它是联系“学习”的属性。有时为了使 E-R 简洁明了，在图中可以省略属性，着重反映实体间的联系情况，而属性则另外以表格的形式单独列出来。图 1.11 所表示的 E-R 图均省略了属性内容。

由于 E-R 图直观易懂，在概念上表示了一个数据库的信息组织情况，所以若能画出 E-R 图，意味着彻底搞清了问题，此后就可以根据 E-R 图，结合具体 DBMS 的类型把它演变为 DBMS 所能支

持的数据模型, 这种逐步推进的方法如今已经普遍用于数据库设计中, 成为数据库设计中的一个重要步骤。

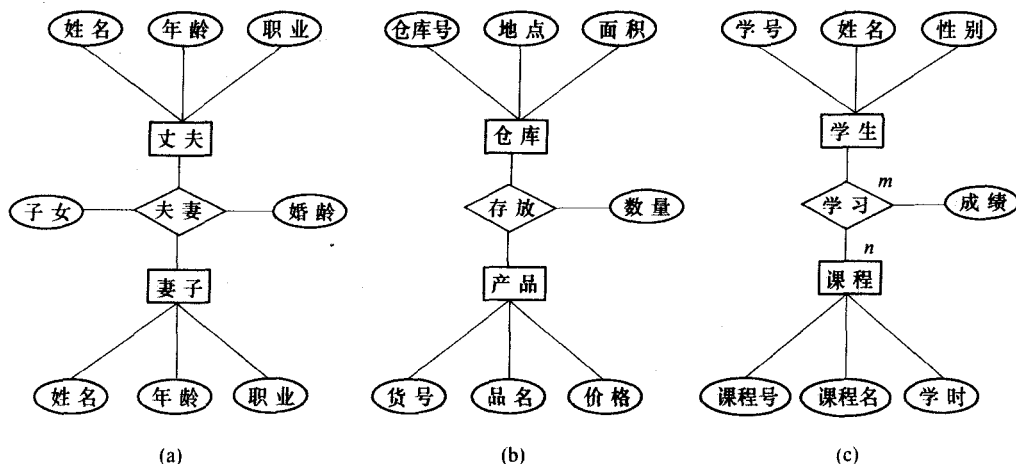


图 1.9 E-R 图的画法

(a) 丈夫与妻子的 1:1 关系; (b) 仓库与产品的 1:n; (c) 学生与课程的 m:n 关系。

现实世界的复杂性导致实体联系的复杂性。表现在 E-R 图上可归结为图 1.10 和图 1.11 所示的几种基本形式:

- (1) 两实体间的二元联系, 如图 1.10 (a) 所示, 有 3 种联系方式。
- (2) 两实体间有两种以上不同的联系, 如图 1.10 (b) 所示。
- (3) 两个以上实体间的多元联系, 如图 1.10 (c) 所示。

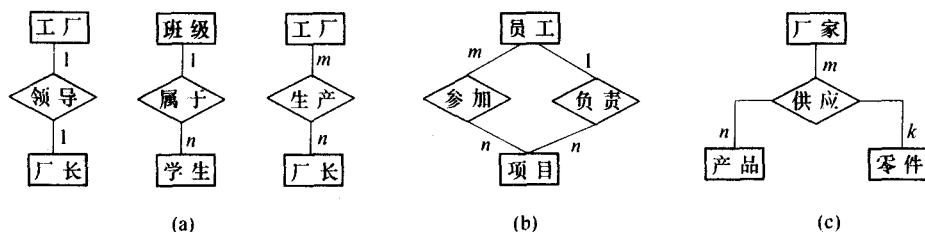


图 1.10 基本联系方式

(a) 两实体间联系; (b) 实体间的两种联系; (c) 多个实体间的多元联系。

- (4) 同一实体内部个体间的二元联系, 如图 1.11 所示。

具体设计 E-R 图时应遵循以下原则:

(1) 首先针对特定用户的应用确定实体、属性和实体间的联系, 作出反映该用户视图的局部 E-R 图。

(2) 综合各个用户的局部 E-R 图, 产生反映数据库整体概念的总体 E-R 图。在综合时, 出现于不同 E-R 图中的同名实体只能在总体 E-R 图中出现一次, 以便消除冗余。要消除那些同名异义或同义异名的现象, 以保持数据一致性。要注意消除那些冗余的联系, 因为冗余信息会影响数据的完整性, 使维护工作变得复杂化, 但有时也要折衷考虑, 因为有时必要的冗余对提高数据处理效率有利。另外, 在综合时也可以在总体 E-R 图中增加新的联

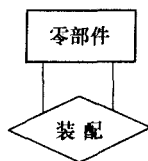


图 1.11 实体内部个体间的二元联系