

● 普通高等教育“十三五”规划教材
(计算机专业群)

数据库原理与技术

主 编 程传慧
副主编 曾 玲 杨晓艳

(第三版)



中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十

算机专业群)

数据库原理与技术

(第三版)

主 编 程传慧

副主编 曾 玲 杨晓艳



中国水利水电出版社
www.waterpub.com.cn

· 北京 ·

内 容 提 要

本书较系统、全面地叙述了数据库系统的基本概念、基本原理和基本方法。内容包括数据库概念、数据模型、存储结构、关系数据理论和关系数据库的基本概念、SQL 语言、数据库应用系统设计、SQL Server 2014 数据库的基础知识与基本开发方法。还介绍了 C/S 与 B/S 模式、数据库保护、数据仓库、数据挖掘、分布式数据库等知识。本教材强调理论联系实际,最后一章介绍了管理信息系统软件库和软件生产线,无需掌握编程语言,只需建模并在建模过程中设定参数就能完成一般管理信息系统的设计与建设。

本书可作为高等院校本、专科及在职工工学习数据库理论与技术的教材,也可供研究生和从事计算机工作的科技工作者参考。

本书配有电子教案,读者可以从中国水利水电出版社网站和万水书苑免费下载,网址为:
<http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目 (C I P) 数据

数据库原理与技术 / 程传慧主编. -- 3版. -- 北京:
中国水利水电出版社, 2017.8
普通高等教育“十三五”规划教材. 计算机专业群
ISBN 978-7-5170-5655-3

I. ①数… II. ①程… 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181140号

策划编辑: 石永峰

责任编辑: 周益丹

封面设计: 李 佳

书 名	普通高等教育“十三五”规划教材(计算机专业群) 数据库原理与技术(第三版) SHUJUKU YUANLI YU JISHU
作 者	主 编 程传慧 副主编 曾 玲 杨晓艳
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
经 售	北京万水电子信息有限公司 三河市航远印刷有限公司
排 版	184mm×260mm 16开本 15.25印张 374千字
印 刷	2001年10月第1版 2001年10月第1次印刷 2017年8月第3版 2017年8月第1次印刷
规 格	0001—3000册
版 次	34.00元
印 数	
定 价	

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

前 言

数据库是设计与建立管理信息系统的主要支撑，而管理信息系统是计算机应用最主要的内容之一。学习数据库的重点是学习数据库的基本理论、基本知识与基本方法。需要强调的是要理论联系实际，要联系管理信息系统的设计与建设实际进行：由管理信息系统的需求理解数据库系统的设计理念；由管理信息系统的设计过程理解数据库的基本组成；由管理信息系统的应用理解数据库的技术与方法；根据管理信息系统的发展研究数据库理论与技术的创新方向。孤立地学习和讲述这门课程会减少学习数据库的所得，将使其内容变得枯燥无味与难以理解，更无法将之应用于生产实际。

本书第一、二版问世以来，得到了广大读者的肯定，不少读者还提出了许多有益的建议，为新版的编写打下良好基础，我们深表感谢。本书第二版介绍了 SQL Server 2005 与 Oracle 两种数据库管理系统，突出在网络环境中数据库的应用研究，加强涉及数据库与数据表的设计与管理、SQL 语句及内嵌语言的应用、数据完整性与安全性保护、备份与恢复等方面的内容。这些内容既扩展了数据库基本理论与技术的教学内容，也使数据库理论教学与社会实践活动更紧密地结合。但不足的是，关于实验教学工具的改进较小，教的是 SQL Server，却以 VFP 为实践工具，存在理论与实际脱节的弊病。本次改版研究设计了应用 Java 语言开发的 SQL 语言学习工具程序，提供各种可视化界面，帮助读者分步写出 SQL 语句，使得更便于理解 SQL 语句的结构与设计方法，强化这一部分的教学效果。另外还设计了应用 Java 语言开发的基于 SQL Server 数据库的软件生产线。软件生产线由面向系统建模程序和管理信息系统软部件库构成。面向系统建模（System-oriented Modeling, SOM）是在面向对象建模语言——统一建模语言（Unified Modeling Language, UML）基础上设计的新一代建模语言。UML 建模语言是十分成熟的建模语言，已成为标准和规范，它简单易学，信息量丰富，表现力强，是一般软件开发的基础，被普遍使用。但是，它毕竟以类为基本元素，基于其模型无法实现软件设计自动化。面向系统建模以软部件为基本元素，尽量保留 UML 的风格，沿用其图形元素，用部件图代替类图，可以直接基于模型建立应用系统。面向系统建模图形由用例图、数据结构图（或称元数据图）、系统结构图（或称数据操作图）、系统组件图、工作流程图（时序图）组成。用例图描述哪些操作对象做哪些事；数据结构图描述数据构成及数据属性和数据约束；系统结构图描述涉及哪些数据、哪些操作、界面风格、数据联系及处理，以部件图为主要图形元素；系统组件图描述子系统构成；工作流程图描述随时间变化的处理过程，包括人员、操作、时间、权限等要素。

管理信息系统软部件是应用系统中由类与对象组合而成的，集成了多项功能，可以表现多种性能的具有自适应与即插即用特性的通用程序模块，只需要输入必要的参数，就可以让一个部件程序选择并表现某具体功能与特殊的性能。软件生产线系统提供建模工具程序，运行该程序可建立应用系统模型，在建模过程中根据提示输入必要的参数，就能在以分钟计的极短时间里搭建一个局域网上的功能比较齐全的管理系统。这个系统拥有丰富易操作的界面，充分满足用户需求的功能与良好性能，包括各种数据录入与维护的程序，满足各种需要的查

询程序和数据处理程序、各种数据导入或导出程序、多种打印与图形输出程序。将之用于数据库教学，可以不要求学习任何开发语言，不懂程序代码的语法与句法，只要求安装 Java 系统软件（JDK 6.0）和 SQL Server 数据库（SQL Server 2014 及之前版本，也可用于 Oracle、MySQL、Access、DB2、达梦等数据库），进行应用系统需求分析，就可以让学生结合数据库设计实际开发应用系统，通过实践更好地理解与掌握数据库的理论与方法；深入且具体地让学生联系应用系统需求认识数据冗余、共享、数据独立性、各类数据完整性及数据完整性保护、关键字、视图、数据安全、SQL 语言及其应用、数据表结构及其对系统设计的影响、字典表与数据整合、代码表与派生数据及其处理等基本概念、基本理论与基本方法；掌握数据库系统设计方法，从而大大提高数据库的学习质量与动手能力。

软件生产线技术具有实用价值，随着其技术的发展，能大大提高应用系统设计效率，降低成本，提高设计质量，降低维护成本，一般企业管理工作者能自己进行应用系统的维护。在管理信息系统建设时，参与原始代码设计的人员将减少，大部分开发人员的主要工作将集中到数据库设计、应用系统结构研究、系统扩展与维护等工作上来，促使数据库应用范围不断扩展。我们目前的研究还处于早期阶段，缺点与错误在所难免，希望广大读者多提宝贵意见。

本书配套 PPT 教案可在出版社网站上下载，与本书配套的实验工具程序包括 SQL 语句辅助生成系统与管理信息系统软件生产线等全部软件随与本书配套出版的实验指导发布。

本书由程传慧任主编，曾玲和杨晓艳任副主编，程学先为本书的出版提供了技术支持。参与本书编写的还有郑秋华、陈永辉、程传庆等。林姗、刘伟、胡显波、赵岚、江南、肖模艳、龚晓明、王富强、陈义、史函、刘玲玲、熊晓菁、童亚拉、周金松、祝苏薇、王嘉、黎柳柳、苏艳、蒋慧婷、陈莉、湛章恒、张军、赵普、高霞、钱涛、张俊、李珺、张慧萍、顾梦霞、贺红艳、罗红芳、陈小娟、齐赛、聂志恒、王玉民、龚文义等参加了本书编审与软件设计工作，在此表示感谢。

编者

2017 年 6 月

目 录

前言	
第1章 数据库基础知识	1
本章学习目标	1
1.1 数据处理	1
1.1.1 利用文件系统进行数据处理	1
1.1.2 从实例看数据库的数据处理技术	4
1.2 数据库技术概述	7
1.3 数据库的数据结构及存储结构	12
1.3.1 链表式数据结构	12
1.3.2 关系数据库结构概述	13
1.4 索引文件组织	16
1.4.1 索引文件	16
1.4.2 非关键字索引文件	17
1.4.3 B+树索引结构	19
本章小结	20
习题一	20
第2章 数据库设计中的数据模型	21
本章学习目标	21
2.1 数据模型	21
2.1.1 数据模型概念	21
2.1.2 数据之间的联系	22
2.1.3 实体—联系模型	23
2.2 关系数据模型	25
2.2.1 关系数据模型的概念	25
2.2.2 关系数据模型的设计	27
2.3 面向对象数据模型	28
2.3.1 UML 定义的类图	28
2.3.2 利用 Rose 建模操作	32
2.3.3 从建模到建库与建表的自动化操作	34
本章小结	35
习题二	35
第3章 关系数据库	36
本章学习目标	36
3.1 基本概念	36
3.2 函数依赖	37
3.2.1 函数依赖概念	37
3.2.2 部分函数依赖	38
3.2.3 完全函数依赖	38
3.2.4 传递函数依赖	38
3.3 候选关键字与主属性	39
3.3.1 候选关键字	39
3.3.2 主属性	40
3.4 关系规范化	40
3.4.1 问题的提出	40
3.4.2 范式	42
3.4.3 关系分解的正确性	47
本章小结	47
习题三	48
第4章 SQL Server 基础	49
本章学习目标	49
4.1 SQL Server 管理工具	49
4.2 可视化建立数据库、表、索引的操作	52
4.2.1 建立数据库	52
4.2.2 建立数据表	53
4.2.3 修改表结构	56
4.2.4 建立索引	56
4.2.5 数据维护操作	58
4.3 建立视图的操作	59
4.3.1 建立视图	59
4.3.2 使用视图	60
4.4 数据完整性保护	61
4.4.1 实体完整性保护的实现	61
4.4.2 参照完整性保护的实现	61
4.4.3 域完整性保护的实现	64
4.5 数据库安全性管理	64
4.5.1 主体与安全对象	65
4.5.2 身份验证模式	65

4.5.3 登录名的管理	66	5.10 查询优化	102
4.5.4 创建架构	67	本章小结	103
4.5.5 针对具体数据库创建用户名	68	习题五	104
4.5.6 服务器角色	68	第 6 章 T-SQL 语言程序设计	106
4.5.7 数据库角色	69	本章学习目标	106
4.5.8 权限管理	71	6.1 T-SQL 程序设计的语言元素	106
本章小结	72	6.1.1 变量	106
习题四	72	6.1.2 运算符	107
第 5 章 关系代数与 SQL 语言	74	6.1.3 表达式及常用命令	108
本章学习目标	74	6.1.4 函数	108
5.1 关系代数	75	6.1.5 流程控制语句	112
5.1.1 传统的集合运算	75	6.1.6 注释	115
5.1.2 专门的关系运算	78	6.2 SQL Server 中的存储过程	116
5.2 关系演算	81	6.2.1 存储过程的概念	116
5.3 SQL 语言概貌	82	6.2.2 存储过程的优点	116
5.4 SQL 数据定义功能	82	6.2.3 使用对象资源管理器创建存储过程	117
5.4.1 基本表的定义和修改	82	6.2.4 使用 T-SQL 命令创建存储过程	118
5.4.2 索引的建立和删除	84	6.2.5 重新命名存储过程	119
5.5 SQL 数据查询语句	84	6.2.6 删除存储过程	120
5.5.1 标准 SQL 数据查询语句格式	84	6.2.7 执行存储过程	120
5.5.2 对单一表查询语句	85	6.2.8 系统存储过程	120
5.5.3 对两个以上表的连接查询	86	6.3 SQL Server 中的触发器	121
5.5.4 嵌套查询	87	6.3.1 触发器的概念及作用	121
5.5.5 关系除法	88	6.3.2 触发器的种类	122
5.6 视图	89	6.3.3 创建触发器	122
5.6.1 建立视图的语句	89	6.3.4 触发器的原理	125
5.6.2 删除视图语句	90	6.3.5 INSTEAD OF 触发器	125
5.7 SQL Server 中 SQL 语句的加强	90	6.3.6 触发器的应用	126
5.7.1 T-SQL 语言对 SQL 定义语句 的加强	91	本章小结	128
5.7.2 涉及数据完整性的数据表结构 修改语句	94	习题六	128
5.7.3 T-SQL 语言对 SQL 查询语句 的加强	97	第 7 章 数据库管理与数据安全	129
5.8 SQL 数据更新语句	99	本章学习目标	129
5.8.1 修改 (UPDATE) 语句	99	7.1 数据库的安全性实施方法	129
5.8.2 删除 (DELETE) 语句	99	7.1.1 应用 SQL Server 语句建立登录名、 架构与用户	129
5.8.3 插入 (INSERT) 语句	100	7.1.2 SQL 语言访问权限控制	130
5.9 嵌入式 SQL	101	7.2 事务处理	133
		7.2.1 事务的基本概念	133
		7.2.2 事务处理过程分析	134

7.2.3 SQL 的事务管理	134	9.1.4 分类与决策树	177
7.3 并发控制	135	9.1.5 聚类	183
7.3.1 并发处理产生的三种不一致性	135	9.2 数据仓库	185
7.3.2 封锁	137	9.2.1 数据仓库的概念	185
7.4 数据库的备份与恢复	139	9.2.2 联机事务处理	186
7.4.1 故障的类型	139	9.2.3 联机分析技术概述	187
7.4.2 事务日志	140	9.2.4 数据仓库的架构	189
7.4.3 恢复	141	9.2.5 数据收集	190
7.4.4 数据的转储	141	9.2.6 SQL Server 中的数据仓库组件	192
7.5 SQL Server 中的数据导入和导出	142	9.3 分布式数据库	192
7.5.1 使用 T-SQL 进行数据导入、导出	142	9.3.1 分布式数据库系统概述	193
7.5.2 使用 SQL Server 2014 数据导入、导出向导	143	9.3.2 分布式数据存储	193
7.5.3 利用对象资源管理器导入、导出	146	9.3.3 分布式数据的查询处理	194
7.6 SQL Server 应用系统开发环境	146	9.3.4 分布式数据库系统中的事务处理	195
7.6.1 SQL Server 应用系统的两种系统结构	146	9.3.5 数据对象的命名方式与目录表的管理	198
7.6.2 ODBC	147	9.3.6 更新传播	200
7.6.3 JDBC	148	本章小结	200
本章小结	150	习题九	200
习题七	150	第 10 章 管理信息系统软部件库与软件生产线	202
第 8 章 数据库应用系统设计	152	本章学习目标	202
本章学习目标	152	10.1 管理信息系统软件生产线	202
8.1 概述	152	10.2 管理信息系统软部件库及数据库桌面系统	206
8.2 数据库结构设计	153	10.2.1 数据库桌面系统概述	206
8.2.1 数据库结构设计步骤	153	10.2.2 表格式数据维护部件程序功能、性能与操作说明	207
8.2.2 需求分析	153	10.2.3 单记录式数据维护部件程序功能、性能与操作说明	210
8.2.3 概念结构设计	158	10.2.4 查询类部件程序功能、性能与操作说明	215
8.2.4 逻辑结构设计	160	10.2.5 数据处理类部件程序功能、性能与操作说明	217
8.2.5 数据库物理设计	161	10.2.6 数据导入导出部件程序功能、性能与操作说明	219
8.3 应用程序结构设计	163	10.2.7 打印报表部件程序功能、性能与操作说明	220
本章小结	166	10.3 用例图	222
习题八	166		
第 9 章 数据库新技术介绍	168		
本章学习目标	168		
9.1 数据挖掘	168		
9.1.1 数据挖掘技术概述	168		
9.1.2 公式发现	169		
9.1.3 关联规则	173		

10.3.1 功能	222	10.6 系统结构图	229
10.3.2 主要图形元素或按钮	222	10.6.1 功能	229
10.3.3 主要操作	223	10.6.2 主要图形元素或按钮	229
10.4 数据结构图	224	10.6.3 主要操作	230
10.4.1 功能	224	10.7 生成应用系统	231
10.4.2 主要图形元素或按钮	225	10.8 工作流程图	231
10.4.3 主要操作	226	10.8.1 功能	231
10.5 系统组件图	227	10.8.2 主要图形元素	232
10.5.1 功能	227	10.8.3 主要操作	233
10.5.2 主要图形元素或按钮	227	本章小结	234
10.5.3 主要操作	228	参考文献	236

第 1 章 数据库基础知识

本章学习目标

本章联系数据库管理系统的发展历程介绍数据库管理系统的基本概念，包括文件管理系统进行数据管理的缺点，数据库管理系统的优点，数据库管理系统的组成，数据库的数据结构和存储结构等。通过本章学习，读者应该掌握以下内容：

- 数据库管理系统与传统的文件管理系统的主要区别与各自的特点
- 数据库管理系统（DBMS）的组成与功能
- 数据库的数据结构和存储结构
- 索引的概念，索引文件类型
- 索引文件结构与使用
- B+树结构及应用

当今时代是信息技术飞速发展的时代。所谓信息，是以数据为载体的客观世界实际存在的事物、事件或概念在人们头脑中的反映。信息系统是以计算机为核心，以数据库为基础，对信息进行收集、组织、存储、加工、传播、管理和使用的系统。数据库能借助计算机保存和管理大量复杂的数据，快速而有效地为多个不同的用户和各种应用程序提供需要的数据，以便人们能更方便更充分地利用这些宝贵的信息资源。

数据管理是指数据的收集、整理、组织、存储、查询、维护和传送等各种操作，是数据处理的基本环节，是任何数据处理任务必有的共性部分。因此应当开发出通用而又方便好用的软件，把数据有效地管理起来，以便最大限度地减轻应用人员的负担。数据库技术正是针对这一目标逐渐完善的一门计算机软件技术。它所研究的问题就是如何科学地组织和存储数据，如何高效地获取和处理数据，如何更广泛、更安全地共享数据。

1.1 数据处理

1.1.1 利用文件系统进行数据处理

根据计算机软件和硬件的发展，数据管理技术的发展大体上分为三个阶段：人工管理阶段、文件系统阶段和数据库系统阶段。

1. 人工管理阶段（20 世纪 50 年中期以前）

20 世纪 50 年代中期以前，计算机主要用于科学计算。当时尚无操作系统与高级语言，软件采用机器语言编写。在科学计算公式中用到一些数据，但数据量很小，一般不存在对它们添加、修改、删除等维护操作，也不要求检索，因此无论从软件开发环境，还是从应用需求上都

将它们与程序紧密结合在一起,对这些数据不需要共享,而且当时也不允许共享。

2. 文件系统阶段 (20 世纪 50 年代后期至 60 年代)

20 世纪 50 年代后期到 60 年代中期,随着计算机科学的发展,这时硬件方面已有了磁盘、磁鼓等直接存取存储设备;软件方面出现了操作系统及高级语言。操作系统中有专门的文件管理软件,称为文件系统,处理方法上不仅有了文件批处理,而且能够联机实时处理。计算机应用也从单纯科学计算、控制,扩大到电子数据处理系统,包括电算系统(例如工资系统与成本会计计算系统)、统计系统(例如国民经济数据统计系统)、数据更新系统(例如飞机预订订票系统)等,数据量大大增加,且同一组数据往往要求用于不同的计算和统计之中,以供不同客户查询、存储修改变动了的数据、根据用户的需要添加新的数据、删除一些过时无用的数据等维护操作。为便于对数据进行维护,也提供不同用户查询的需求,人们利用文件系统将数据从程序中分离出来形成专门的数据文件。例如将两件商品(Commodity)的有关数据:商品名称(WareName)、规格(Specification)、单价(Unitprice)、说明(Illuminate)等输入到一个文件中的 C 语言程序,如例 1.1 所示。

【例 1.1】C 语言中将 Commodity 的有关数据: WareName、Specification、Unitprice、Illuminate 等输入到一个文件中的程序。

```
# include "stdio.h"
main()
{ file *fp;
  fp = fopen("Commodity.c","w");
  fputs("Silverware ",fp);
  fputs("Wwmottle      ",fp);
  putw(40,fp);
  fputs("11111111      ",fp);
  fputs("Chinaware   ",fp);
  fputs("Popularware   ",fp);
  putw(100,fp);
  fputs("222222222222    ",fp);
  fclose(fp);
}
```

如要显示文件中数据,可使用例 1.2 所示的程序。

【例 1.2】求显示文件 Commodity.c 中的数据的程序。

```
#include "stdio.h"
#define SIZE 2
struct Commodity _type
{ char WareName[12];
  char Specification[16];
  int Unitprice;
  char Illuminate[20];
}stud[SIZE];
main()
{ int i;
  file *fp;
  fp=fopen("Commodity.c","r");
```

```
for(i=0;i<SIZE;i++)
{
    fread(&stud[i], sizeof(struct Commodity_type),1,fp);
    printf("%28s%20s %8d %200s %\n", stud[i].WareName, stud[i].Specification,
stud[i].Unitprice, stud[i].Illuminate);
}
```

如果要按一定条件显示一定范围的数据,上述程序需略作修改:对每一组数据逐一判别是否满足要求,再按预定范围显示。如果要修改文件中的数据,要首先使用类似于例 1.2 的程序,以读的方式打开文件,将文件中数据读入到变量中,修改变量中数据的值,再以类似于例 1.1 的程序以写的方式打开文件,将修改后的数据输回到文件中。数据处理全过程如图 1.1 所示。

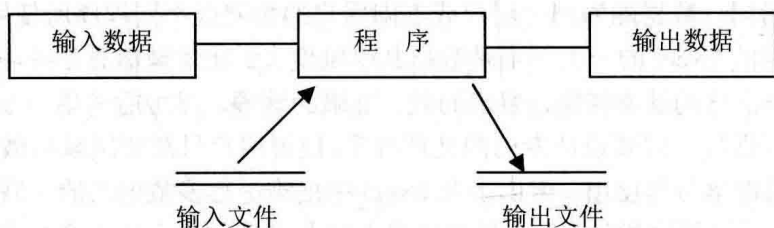


图 1.1 传统数据处理模式

使用数据文件实现了数据与程序相分离,分别采用两个文件各自存放数据与程序,我们将之称为实现了数据的物理独立。

利用文件系统解决程序和数据的存取操作,程序员只需关心文件的逻辑结构,无须关心如何转为物理存储,从而使程序设计变得简单,还可专门对数据文件进行管理,单独对数据进行操作或维护。不同的程序只要以读的方式打开数据文件,并按数据存放的格式将数据取出便可使用这些数据,在一定的范围内可以做到数据为不同程序所共享。数据可长期保存,大大方便了用户的使用。在这一阶段,文件越来越多样化和结构化,出现了方便查询与直接存取的索引文件、方便多条件查询的倒排表文件、将内容联系起来的链接文件、线性表文件等各种可适应不同应用需要的文件。

使用数据文件的缺点是如果要使用数据和维护数据,必须知道数据存放的格式,即要知道数据存取的逻辑结构。例如在前述例 1.2 中,必须知道共存放了两件商品的记录,且每个商品的记录包括四个数据:WareName、Specification、Unitprice、Illuminate。其中第一、二、四是字符串类型数据,宽度为 12、16 与 20;第三个数据是整型数据。数据个数、数据类型与宽度必须与数据文件中的数据一一对应,否则,程序所读出的数据将会出错。其原因显然是因为数据结构和程序语句紧密相关,或者说数据和程序之间缺少数据逻辑独立性。

在实际系统中,由于数据值及逻辑结构都可能不断变化,如果每次变化都要求所有应用程序作相应修改,其工作量之大实在令人无法承受。而且,对这些数据文件要么控制不让打开,一旦能够打开,每个程序都可取用全部数据,所有人使用数据的权限都相同,完全无安全性可言。由于数据文件中除了数据不再有其他信息,也就无法对数据作统一的控制和管理。因为以上原因,这样的系统中数据共享就只能局限在一定范围内。同一数据在多个地方同时存放,称为数据冗余。重复次数多称为数据冗余度大。文件系统中数据冗余度大且无法有效控制,一方面浪费存储空间,降低运行效率;另一方面降低系统可靠性、正确性,降低系统价值,对系统

进行修改、维护都麻烦。

同一数据在多个地方同时存放,那么同一数据在不同存放地的值可能不相同,称为数据不一致,这将会降低信息价值,甚至造成重大损失。

从数据处理需求来看,计算机辅助管理的内容逐步向纵深发展,应用面愈来愈广,部门的壁垒开始被打破,数据不再只是用于计算、统计,还作为信息的载体被存储,成为人们宝贵的信息资源,用于检索、统计、预测及决策。要求对人工管理过程广泛予以模拟;要求收集并保存大量的数据;要求不断对历史上收集的数据进行筛选、分析和提炼;要求为决策提供大量的数据,并产生不同决策方案以供决策者参考;要求随时将各类信息向各方面发布等等。这就要求系统效率更高,实现更大范围的数据共享。

数据共享是指同一数据能为同一用户或不同用户的相同或不同程序所使用。

目前广泛使用的 Excel 的一大弱点是数据共享难度大。在管理信息系统中,信息数据是实现管理规范、标准化的基本依据,要求高效、准确的管理,尽可能简单、方便地操作。实际应用系统希望能根据每一需要设计专门的处理程序,使得用户只需要用鼠标做极少的点击就能操作,能十分容易地学习与使用。但由于在 Excel 中很难进行多数据表的关联操作,无法直接进行代码变换,很难达到这样的目标。管理信息系统强调确保数据的安全与准确,对不同操作人员应当提供不同操作权限。但对于 Excel 数据文件,要么不能打开,而一旦打开,任何人对所有内容就都可读、可修改。学习与使用 Excel 确实很简单,不需要预先建立数据结构,这是优点所在,但也是其弱点的根源。

3. 数据库系统阶段(20 世纪 60 年代后期)

20 世纪 60 年代后期以来,计算机用于管理的规模更为庞大,应用越来越广泛,数据量急剧增长,同时多种应用、多种语言互相覆盖,共享数据的要求越来越强烈。这时硬件也飞快发展,有了大容量磁盘,硬件价格下跌,软件价格上升,为编制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处理。在这种背景下,以文件系统作为管理手段已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,就出现了数据库技术,出现了统一管理数据的专门软件系统:数据库管理系统。数据库系统使前述例 1.1、例 1.2 的工作变得十分简单,也解决了文件系统中所存在的许多问题。

1.1.2 从实例看数据库的数据处理技术

当前关系数据库管理系统的产品主要有 Oracle、DB2、Informix、Sybase ASE、SQL Server、MySQL、达梦、Access、Visual FoxPro 等,Oracle 是大型数据库,SQL Server 是可以应用于网上的小型数据库,Access、Visual FoxPro 是桌面式数据库,达梦是我国自行开发的数据库管理系统等。

如果在 SQL Server 中要完成前述任务,首先要建立数据存储结构。可先建立数据库,例如建立名为 Waremanage 的数据库,当打开该数据库后,可使用下述命令建立空表结构:

```
CREATE TABLE Commodity(WareName CHAR(12),Secfcaton CHAR(16),  
Unitprice INT,Illuminate TEXT);
```

如要存入两件商品数据,可使用下述命令:

```
INSERT INTO Commodity VALUES("Silverware", "Wwmottle",40, "1111111111111111");
```

```
INSERT INTO Commodity VALUES("Chinaware", "Popularware",100,
"22222222222222222222222222222222");
```

与 C 语言程序不同,在输入商品名称、规格、说明等数据时,数据值的宽度不到预定的字符宽度时,无须加填充空格;在数据维护过程中对数据格式要求不是那么严格。

例如要显示表 **Commodity** 中的内容,可使用下述命令:

```
SELECT * FROM Commodity
```

如要求按一定条件显示某一定范围内的内容,例如只显示单价为 40 的商品名称、规格与单价,可使用下述命令:

```
SELECT WareName,Secfcaton, Unitprice FROM Commodity WHERE Unitprice =40
```

显然,采用数据库系统管理软件后,有以下优点:

(1) 程序极为简单,在 C 语言中需要大量语句才能实现的功能,目前仅一两句即可完成。对数据的每一种维护,往往也只需要一两个语句即可实现。

(2) 程序中不需要了解数据的数量和顺序,只需要知道所需要的数据的名字,就可以指名道姓地进行操作。

(3) 可直接对数据的某一部分分量进行操作,而无须知道全面的数据结构及其他分量的个数、名字和数据类型。

(4) 只要初始定义的数据结构所需的那一部分分量的名字和数据类型不发生变化,程序将无须随数据结构改变而修改。

这些优点为用户和程序员均带来方便,也使数据广泛共享真正成为可能。

数据库系统管理软件是如何实现上述功能的呢?

通过实验可以发现,数据库将所管理的数据集中到一到多个文件中保存。在存放数据时,会将数据分为不同类型,采用不同格式存放:以二进制码格式存放如整型、浮点型这样一些可能需要进行算术运算与比较大小的数据以及图形、声音等量大但结构复杂的数据,以 ASCII 码存放如字符、文本等这样一些不需要进行算术运算但需要对之查询或比较的数据。SQL Server 数据库中基本数据类型及意义如表 1.1 所示。

表 1.1 SQL Server 基本数据类型说明

类型名称	类型代码	意义	宽度	说明
bigint	7F7F	大整型	8	介于 -2^{63} 到 $2^{63}-1$ 的整型数据
binary	5353	二进制	*50	定长二进制数据,最长 8000
bit	6868	位	1	1 或 0 的整数数据
char	5151	字符串	*10	最大长度为 8000 的固定长度非 Unicode 字符数据,一个中文字用两个 char 字符表示
datetime	3D3D	时间	8	介于 1753.1.1~9999.12.31 的日期时间数
decimal	6A6A	十进制	9	精确数据类型, $-10^{38}+1\sim 10^{38}-1$,固定精度和小数位的数字数据
float	3E3E	浮点	8	$-1.79E+308\sim 1.79E+308$ 的浮点数
image	2222	图像	16	变长二进制数据,最长 $2^{31}-1$ 字节
int	3838	整型	4	介于 $-2^{61}\sim 2^{61}-1$ 的整型数据
money	3C3C	货币	8	介于 $-2^{63}\sim 2^{63}-1$ 之间货币数值

续表

类型名称	类型代码	意义	宽度	说明
nchar	1111	字符串	*10	最大长度为 4000 的固定长度的 Unicode 码字符数据，一个中文字用一个 nchar 字符表示
ntext	6363	文本	16	可变长度的 Unicode 码字符数据，最大长度由内存限定
numeric	6C6C	数字	9	精确数据类型，功能上等同于 decimal
nvarchar	1919	字符串	*50	最大长度为 4000 的可变长度的 Unicode 码字符数据
real	3B3B	逻辑	4	-3.40E+38~3.40E+38 的浮点数
smalldatetime	3A3A	小时间	4	1900.1.1~2079.6.6，日期时间数据
smallint	3434	小整型	2	从 -2^{15} ~ $2^{15}-1$ 的整型数据
smallmoney	7A7A	小货币	4	-214748.36~+214748.36 货币数值
sql_variant	6262	SQL 变量		支持 SQL Server 的多种数据类型
text	2323	文本	16	可变长度的非 Unicode 字符数据，最大长度由内存限定
timestamp	4343	时间容器	8	数据库范围的数字，随更新而更新
tinyint	3030	微整型	1	从 0 到 255 的整型数据
uniqueidentifier	2424		16	全局唯一标识符（GUID）
varbinary	5B5B	（变）二进制	*50	可变长二进制数据，最长 8000
varchar	5151	（变）字符串	*50	最大长度为 8000 的可变长度的非 Unicode 字符数据

其中宽度数据前加*的为默认值，具体设计时应当根据实际情况修改。未加星号的数字为系统自动设置的定长宽度，用户无需设置。

目前使用最多的是关系数据库，其中数据以记录为单位等长、顺序存放，各记录中对应数据项的数据存放时所占据空间的宽度相同，其数据结构可形象看作一个由行和列构成的二维表格，每一数据项为一列，称为字段，每一行称为一条记录，包括一件事物的有关数据。例如一个关于商品的关系数据库数据表的结构与内容如表 1.2 所示。

表 1.2 商品表结构与内容

商品代码	商品名称	单价	数量	备注
m01	笔记本	6000	10	MSI GL62M 7RD-223CN 15.6 英寸(i7-7700HQ)
m02	电脑	4000	20	DIY 组装机 i5 7500/GTX1060/240G

SQL Server 数据库是目前常用的一种关系数据库，近期的有 2008、2012、2014、2016 等不同版本，在数据存储等基本操作中特性与操作方法相似。如果用它存放数据，首先建立数据库，生成一个数据文件和一个日志文件。数据以数据表为单位存放到数据文件之中。操作时先设计数据表结构，建立数据表。例如，可以设计“商品代码”的数据为字符类型（char），宽度为 8，“商品名称”的数据为字符类型（nchar），宽度为 24，“单价”为 numeric 类型，长度为 10，其中小数 2 位、“数量”为整型，系统默认为 32 位二进制数、“备注”为文本类型（ntext），允许换行，长度不限。利用 SQL Server 数据库提供的可视化操作界面可以定义及建立数据表，之后通过实验可以发现关于所定义的结构内容被按一定格式写到了数据文件中。之后，可以利

用 SQL Server 数据库提供的可视化操作界面录入各记录数据,通过实验可以发现所输入的数据也以一定格式存放到数据文件中。实验证明,SQL Server 数据库中数据库文件分成多个相关部分,其中有一部分存放所定义的数据结构的内容,再有一部分按顺序存放数据,还有一部分以链表结构形式存放文本类型等数据。

Oracle 是另一种应用普遍的大型数据库产品,其较新版本是 Oracle 11g,除具有关系数据库功能外,还有较强的数据仓库功能,适应性很广。它的数据存储容量可达 8TB,实际数据容量只受操作系统限制。其数据存储方式与 SQL Server 不同,数据库下设表空间(一种逻辑结构),表空间内包括多个文件,表、索引、数据字典分布在这些文件中。数据字典存放关于数据逻辑结构的定义。表空间将用户数据、数据字典、索引信息、回滚数据(为保证在并发式共享数据情况下数据正确性而生成的一种供恢复用的临时性数据)分开,使其具有良好可扩展性、数据安全性、应用灵活性、使用高效率等。

从上面描述中可见,在关系数据库中对于类似于文本(存放履历、手册、纯文本文件等数据)、图像(存放相片、图形、语音、非纯文本类型及其他二进制数据)等类型的数据采用特殊方式存放。这是因为,在同一个数据表的不同记录中,这些类型字段中的数据可能有,可能无,实际数据的长度差别很大,小的为 0,大的到兆甚至 G。如用等长方式存储,许多空间被空置,占用存储空间太多,不便管理也影响效率,因而对这类数据普遍采用链表结构存放。而对于其他类型的数据则采用等长顺序存放的方式。因此,一般数据库的数据文件都设计有顺序结构与链表结构等两种不同文件结构,分别存放一般数据与文本与图片类型数据。在顺序文件部分,以等长记录方式存放一般数据的记录,在其中文本类型、图形类型等类型字段的位置上只存放指针,指向链表结构部分中的相应内容。

归纳以上内容,数据库都要求预先定义数据逻辑结构,并用专门文件或指定文件的一部分存储关于结构的描述,程序员编写程序时就无须了解数据的全局结构,而只需关心他所涉及的那部分数据项,其他事项可利用数据库管理系统软件(DBMS)来帮助完成。DBMS 可以分析数据全局与各数据项结构,分析一条记录总长度,及每一个数据项的名字、类型、从第几个字节开始及共占据多少宽度等等,之后就可自动从数据区中根据需要提取数据。这样一些烦琐的工作由软件自动完成,程序设计的工作就将大大简化,数据结构的变化对程序的影响也将大大减少。由于有对数据结构整体描述的内容,在其中还可加入其他内容,例如关于记录的标识属性,关于一个数据项数据的合理范围,关于数据使用权限等等,可借之实现对数据存储、使用、传送的控制,使数据安全地使用。这样一种设计初步实现了数据与数据逻辑结构描述(称为模式)的分离。这是数据逻辑独立的基础。

1.2 数据库技术概述

数据库指有组织的、动态地存储的、结构化的、相互关联的数据的集合。

1. 数据库系统

数据库系统一般由数据库、支持数据库运行的软件与硬件、数据库管理系统、应用系统、数据库管理员和用户构成。

数据库是通过综合多个用户的文件中的数据,除去不必要的冗余,建立必要的联系之后建立的数据存储库。集成、共享、存储、信息是数据库的要素,其重要特点是联系。

设计一个数据库应用系统,首先要了解系统对数据和功能的需求;接着对全系统涉及的所有数据进行分析 and 整理、分析数据之间的联系,用一定的数据模型来表示;再求得系统全局的数据结构,用一定语言加以表示及定义;注意考虑数据的存放位置、数据量大小、对安全保密性、数据正确性、防错纠错措施等方面的要求,设计并在计算机中建立数据库结构;在此基础上开发相应维护和使用数据的应用程序。在设计程序时要注意使用方便,操作简单,系统高效可靠。

要尽可能减少数据冗余。所谓数据冗余是指同一数据在多个不同的地方存放。例如,同一个人的基本情况,如果在人事管理部门的系统中存放、在财务部门也存放,在生产管理部门也存放,那就存在明显冗余。另外,如果一组数据在一个表中多次重复,也是冗余。它不仅导致数据量的增加,使系统处理速度变慢,效率降低,而且易发生错误。多一个数据表,就需要多一套维护程序,多一些发生错误的可能,会影响全系统的性能。在实际设计中,应尽量减少数据冗余,控制冗余度。需要正确定义全局数据结构。

用某一种数据库语言对全局数据结构的定义称为这种数据库的概念模式,简称模式。例如设计关于“学生”的数据,定义为二维结构的表,表名为 Student,包括 Name、Num、Age 三个数据分量,数据类型分别为字符型、整型、整型,所占宽度分别为 8、4、4 个字节。这些内容就是对关系数据库“学生”的模式定义的主要内容。关系数据库的模式除包括数据库名、数据结构方式、记录的构成等内容外,还包括记录的标识性数据分量、数据范围及使用权限等内容。在关系数据库中标识性数据分量指能唯一标志一条记录的数据分量。模式常常简单地被表示为:模式名(数据项 1,数据项 2,...),例如 Student (Name,Num,Age)。上述数据分量我们称之为字段,数据分量名称为字段名,每条记录该数据分量的值称为字段值。标识性数据分量称关键字。

要增强数据的共享性、尽量减少数据冗余,还需要进一步提高数据的逻辑独立性,减少应用程序对全局性数据结构的依赖,让应用程序只和局部数据结构相关。为此,可进一步定义概念模式的逻辑子集,称为子模式。

在数据库系统管理软件中,有专门定义数据模式的语句,例如,建立学生(学号,姓名,年龄)和成绩(学号,课名,分数)两个表(名字均改用英文字符表示)的语句:

```
CREATE TABLE Student (Num i, Name c(8), age i)
CREATE TABLE Result (Num i, LessonName c(28), Fraction i)
```

在数据库系统管理软件中,还有建立视图的语句。例如欲建立视图 Sr,列举所有不及格的学生,可以用一条语句实现:

```
CREATE VIEW Sr AS SELECT Student.Name,Result.LessonName,Result.Fraction FROM
Student, Result (WHERE Student.Num =Result.Num AND Result.Fraction <60)
```

有了这个视图之后,在关系数据库系统中可将 Sr 视同一般表,通过它对源表 Student 和 Result 进行查询操作。该视图实现了两个源表的联系,相当于源表模式的子集。它使用的字段名可以与源表不同,以后若源库结构发生变化,包括上述相应字段名变化,我们可修改视图定义来局部适应这些变化,而不需要修改程序。视图和表不同之处在于它并没有真正地存储数据,它所存取的数据必须依附于所关联的数据表,是一种虚的映射关系。

还有一些数据库,在关于全局数据结构的子集定义中允许改变对应的数据类型、宽度,还可加入关于权限控制方面的内容,这种用一定数据库语言对局部数据结构的描述称为子模式