



普通高等教育“十一五”国家级规划教材

教育部“高等学校教学质量与教学改革工程”立项项目

王彤 王良 尚文倩 编著

数据库技术及应用 课程实践

计算机科学与技术专业实践系列教材

清华大学出版社





普通高等教育“十一五”国家级规划教材

计算机科学与技术专业实践系列教材

教育部“高等学校教学质量与教学改革工程”立项项目

数据库技术及应用 课程实践

王彤 王良 尚文倩 编著

清华大学出版社
北京

内 容 简 介

本教材以一个虚拟的小型电子商务网站“网上多媒体音乐商店”(简称 NetMusicShop 项目)的开发为主线,设计了一系列以任务驱动的相关实验,内容覆盖了数据库应用的主要技术,包括数据库设计、数据库管理、数据库编程和 Web 数据库开发等。随着与课堂教学内容同步的实验任务的逐一完成,NetMusicShop 项目经历了需求分析、系统设计、系统实现的完整开发过程,并在不断调整、迭代中得以完善。各种分散抽象、看似杂乱的知识点,融入项目的开发中,被组织成一个相互关联的有机整体,实现了知识的融会贯通。而学习者则遍历了“数据库技术及应用”课程中的全部知识点,并通过亲身的实践,更好地体会到知识与技术间的相互关联,巩固了知识,提高了应用开发能力。

本书的实验环境为 Microsoft SQL Server 2008 和 Microsoft Visual Studio 2008。为了顺利开展实验,建议读者最好具备“网页设计”和“C++ 语言”两门课程的学习基础。

本书可作为高等学校学生学习“数据库技术及应用”、“数据库管理与开发”、“数据库原理及应用”、“数据库系统概论”等相关课程的实验指导教材,也可供承担数据库课程教学的教师、开发数据库应用系统的编程爱好者和工程技术人员参考。

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。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据库技术及应用课程实践/王彤等编著. —北京:清华大学出版社,2011.6

(计算机科学与技术专业实践系列教材)

ISBN 978-7-302-25088-3

I. ①数… II. ①王… 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(2011)第 046562 号

责任编辑:焦虹

责任校对:李建庄

责任印制:杨艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:22.5

字 数:560 千字

版 次:2011 年 6 月第 1 版

印 次:2011 年 6 月第 1 次印刷

印 数:1~3000

定 价:33.00 元

产品编号:036856-01

普通高等教育“十一五”国家级规划教材
计算机科学与技术专业实践系列教材

编 委 会

主 任：王志英

副 主 任：汤志忠

编 委 委 员：陈向群 樊晓桢 邝 坚

吴 跃 张 莉

前 言

课程实践(包括课程实验和课程设计)是大学教育中最重要也是最基础的实践教学环节,其开展的好坏直接影响到后继课程的学习和后继实践活动(如大学生创新计划、课外科技活动、社会实践及毕业设计等)的质量。作为一门实践性很强的课程,数据库的教学更是如此。

本教材是作者在多年从事数据库课程教学和教改研究工作的基础上,为本科计算机专业、信息管理与信息系统专业及其他相关专业的学生学习“数据库技术及应用”、“数据库管理与开发”、“数据库原理及应用”、“数据库系统概论”等课程编写的实验教材。目的是通过完整的开发案例,使学生能够在实践中加深对数据库基础知识、基本原理的理解和对数据库技术的认识,循序渐进地掌握数据库的开发方法和应用技巧,最终能设计开发出一个功能相对完整的小型数据库应用系统,并能较规范地进行课程设计的文档编写,提交符合要求的课程设计报告。

1. 教材内容组织

本教材紧紧围绕着一个小型数据库应用系统的开发过程中所涉及的知识点来设计、编写实验内容,力求做到目标明确、任务具体、知识技能应用到位,以点带面让知识融会贯通。全书共分6章,其主要内容如下。

第1章 绪论

本章对几种典型的数据库应用系统结构进行了分析比较,重点介绍了B/S结构的数据库应用系统的设计和开发步骤;在此基础上,给出了开发数据库应用系统时几种常用的解决方案。

第2章 “网上多媒体音乐商店”的分析与设计

从这章开始,将以一个虚拟的电子商务网站“网上多媒体音乐商店”(以下简称NetMusicShop项目)的开发为主线,介绍数据库设计、数据库管理、数据库编程和Web数据库应用系统开发的相关知识,使读者掌握实现界面友好、“图文声”并茂的多媒体数据库应用系统的基本技巧。

NetMusicShop项目模拟了一个简化的网上音像制品的购物平台。尽管它的功能比较简单,但其实现技术却覆盖了“数据库技术与应用”课程中的全部知识点。通过本章及后续章节的学习,读者可以了解并完成一个简单的数据库应用系统从分析、设计到实现的完整开发过程。

教学建议:如果在实际授课时,数据库设计部分被安排在后续的教学环节中,本章的学习可以先简化为对“网上多媒体音乐商店”项目基本需求的了解,教师可先演示一下Demo程序(登录清华大学出版社网站即可下载相关教辅资料),在认可NetMusicShop数据库基本表结构,并了解其数据项含义和表之间相互关系的前提下,直接进入下一章的学习。待学生基本掌握了数据库基本对象的管理技术后,再回过头来详读本章,解决“为什么要这样设计NetMusicShop数据库”和“如何规范化地设计和开发数据库应用系统”的问题。

第3章 数据库管理技术

本章将通过8个实验,学习 Microsoft SQL Server 关系型数据库管理系统的安装和基本管理技术,并以第2章完成的数据库设计为依据,搭建起 NetMusicShop 项目的数据库平台。需要完成的实验任务有:

实验 3.1 SQL Server 2008 数据库管理系统的安装。

实验 3.2 数据库的创建与管理。

实验 3.3 数据表的创建与管理。

实验 3.4 数据查询。

实验 3.5 视图的创建与管理。

实验 3.6 数据库的安全性。

实验 3.7 数据库的完整性。

实验 3.8 数据库的备份与恢复。

第4章 数据库编程技术

本章将通过5个实验,学习 Microsoft SQL Server 关系型数据库管理系统的编程技术,从而强化 NetMusicShop 数据库的后台管理功能,使开发的系统更加高效灵活和安全可靠。需要完成的实验任务有:

实验 4.1 T-SQL 程序设计基础。

实验 4.2 存储过程。

实验 4.3 触发器。

实验 4.4 游标。

实验 4.5 并发控制。

第5章 Web 数据库开发技术

在第3章和第4章中,已经学习了 Microsoft SQL Server 数据库的管理和编程技术,完成了后台数据库服务器上 NetMusicShop 数据库的建设。在本章中,将通过8个实验,学习如何在 Microsoft Visual Studio 2008 环境下进行 Web 数据库应用系统客户端的开发,从而完成“网上多媒体音乐商店”网站的建设,最终实现 NetMusicShop 项目的全部开发任务。需要完成的实验任务有:

实验 5.1 Visual Studio 2008 系统的安装和使用。

实验 5.2 C# 语言编程基础。

实验 5.3 网页设计基础。

实验 5.4 ASP.NET 基本控件的使用。

实验 5.5 利用 ADO.NET 接口访问数据库。

实验 5.6 构建多层结构的 ASP.NET 网站。

实验 5.7 利用数据绑定控件访问数据库。

实验 5.8 ASP.NET 网站的高级开发技术。

第6章 “数据库技术及应用”课程设计

本章首先对“网上多媒体音乐商店”项目开发的全过程进行了总结,之后给出“数据库技术及应用”的课程设计要求、课程设计报告书写规范及课程设计综合训练选题。

说明:以上第3、4、5章中的每个实验均由实验目的、实验环境、知识要点、实验内容、实

验步骤、练习思考和拓展提高 7 部分内容组成。实验题目尽量与课堂讲授的知识点相一致,以便进行同步实验时参照;所有实验任务,包括练习思考和拓展提高都围绕着如何运用相关知识,循序渐进地开发 NetMusicShop 项目来设计。

2. 本书特色

- (1) 以项目开发为主线,以任务驱动实验,力求使学生在实践中应用知识、掌握技能。
- (2) 实验目的明确,实验步骤清晰,可操作性强,便于自学。
- (3) 基础实验、拓展实验与综合实验相结合,循序渐进,兼顾不同层次学生的实验能力。
- (4) 实验内容充实,既相互关联又相对独立。教师可根据学生基础、实验学时灵活选择、搭配实验题目。
- (5) 文档资料齐全,提供全部电子版的实验数据库、素材和开发源代码,方便教师备课。
- (6) 与时俱进,采用 Microsoft SQL Server 2008 和 Visual Studio 2008 等较新版本的软件作为数据库应用系统的运行环境和开发平台,体现了开发技术的先进性。

本书由王彤主编,王良、尚文倩、巩微老师参加了部分章节的编写;王永滨、张鹏洲、张弛、温宇俊老师编写了部分实验案例,并对本书的结构编排、内容选取提出了很好的建议。研究生刘随阳、李九青,本科生刘鸿嘉、马晓杰等同学参与了部分实验案例的设计和开发,在此一并表示衷心的感谢。

由于编者水平有限,书中的疏漏与错误在所难免,敬请广大读者批评指正,在此表示衷心的感谢!作者的电子邮箱为:wangtong@cuc.edu.cn。

作 者

目 录

第 1 章 绪论	1
1.1 数据库应用系统的体系结构	1
1.1.1 单用户数据库系统	1
1.1.2 主从式数据库系统	1
1.1.3 分布式数据库系统	1
1.1.4 客户机/服务器两层结构数据库系统	2
1.1.5 三层结构数据库系统	3
1.1.6 浏览器/服务器多层结构数据库系统	3
1.2 数据库应用系统的开发步骤	5
1.2.1 系统需求分析	5
1.2.2 系统设计	6
1.2.3 系统实现	9
1.2.4 系统运行与维护	9
1.3 数据库应用系统的开发平台	10
1.4 练习思考	11
本章参考文献	11
第 2 章 “网上多媒体音乐商店”的分析与设计	12
2.1 NetMusicShop 需求分析	12
2.1.1 NetMusicShop 数据流图	13
2.1.2 NetMusicShop 数据字典	16
2.2 NetMusicShop 设计	27
2.2.1 NetMusicShop 数据库结构设计	27
2.2.2 NetMusicShop 软件结构设计	32
2.3 NetMusicShop 实现	36
2.3.1 NetMusicShop 的组织结构	37
2.3.2 NetMusicShop 的任务分解	37
2.3.3 教学建议	39
2.4 练习思考	39
2.5 拓展提高	40
本章参考文献	40
第 3 章 数据库管理技术	41
3.1 SQL Server 2008 数据库管理系统的安装	41

3.2	数据库的创建与管理	63
3.3	数据表的创建与管理	70
3.4	数据查询	81
3.5	视图的创建与管理	91
3.6	数据库的安全性	96
3.7	数据库的完整性	110
3.8	数据库的备份与恢复	121
	本章参考文献	134
第4章	数据库编程技术	135
4.1	T-SQL 程序设计基础	135
4.2	存储过程	143
4.3	触发器	151
4.4	游标	165
4.5	并发控制	172
	本章参考文献	185
第5章	Web 数据库开发技术	187
5.1	Visual Studio 2008 系统的安装和使用	187
5.2	C# 语言编程基础	201
5.3	网页设计基础	216
5.4	ASP.NET 基本控件的使用	235
5.5	利用 ADO.NET 接口访问数据库	255
5.6	构建多层结构的 ASP.NET 网站	274
5.7	利用数据绑定控件访问数据库	294
5.8	ASP.NET 网站的高级开发技术	324
	本章参考文献	342
第6章	“数据库技术及应用”课程设计	343
6.1	NetMusicShop 项目开发总结	343
6.2	课程设计要求及报告书写规范	343
6.2.1	课程设计要求	344
6.2.2	课程设计报告书写规范	344
6.3	课程设计综合训练选题	346
6.3.1	电视节目查询系统	348
6.3.2	传媒博物馆网站	348
6.3.3	报社新闻文稿编审系统	349
6.3.4	广播电视设备管理系统	349
6.3.5	电视广告业务管理系统	349
	本章参考文献	349

第 1 章 绪 论

数据库技术是研究如何科学地组织和存储数据、高效地检索和管理数据的一门学科。在高度信息化的今天,大到政治军事、金融经济、文化教育,小到企业管理、办公自动化、休闲娱乐,我们的工作、学习和生活一刻也离不开对各种各样数据的管理。作为一种有效的数据管理手段,数据库技术的应用无处不在。

在这一章中,我们首先给出数据库应用系统体系结构的概念,并对几种典型结构进行分析和比较;之后将重点介绍 B/S 结构的数据库应用系统的设计和开发步骤;最后针对 B/S 结构数据库应用系统的开发需求,给出几种常用的解决方案。

1.1 数据库应用系统的体系结构

一个数据库系统(database system)通常由数据库、数据库管理系统(database management system,DBMS)和应用程序三大部分组成。这三部分在计算机上不同的配置方法,构成了不同的数据库系统体系结构,主要有单用户数据库系统、主从式数据库系统、分布式数据库系统、客户机/服务器两层结构数据库系统(简称 C/S)、浏览器/服务器多层结构数据库系统(简称 B/S)以及 C/S+B/S 混合结构的数据库系统等多种形式。

1.1.1 单用户数据库系统

单用户数据库系统是一种早期使用的、结构最为简单的数据库系统。在该体系结构中,整个数据库系统,包括数据库、数据库管理系统和应用程序全都集中装在一台计算机上,由一个用户独占。虽然它管理维护简单、安全性好,但却存在着不同计算机上的用户间不能共享数据的致命弊端。

1.1.2 主从式数据库系统

主从式(又称主机/终端式)数据库系统是对单用户数据库系统的改良,该结构仍然是将所有的资源集中安装在一台计算机上(我们将它称为主机),所不同的是它支持多用户访问,不同的数据库用户借助各自与主机相连的终端设备(主要是显示器)分时访问数据库。这种结构延续了单用户系统管理维护简单的优点,缺点更是显而易见:当共享数据的终端用户数目增加到一定程度后,主机会不堪重负,致使系统性能急剧下降;由于所有的数据处理任务都由主机承担,一旦主机发生故障,系统的所有用户都会受到影响。

上述两种结构本质上都属于集中式的数据库系统。为了克服集中式系统的不足,分布式结构的数据库系统应运而生。

1.1.3 分布式数据库系统

分布式结构的数据库系统是指数据库中的数据在逻辑上是一个整体,但物理分布在计

算机网络的不同节点上。网络中的每个节点都装有本地的数据库管理系统和应用程序,可以处理本地数据库中的数据,执行局部应用。另外也可以同时存取和处理多个异地数据库中的数据,执行全局应用。分布式数据库系统能够很好地满足地理上分散的企业、公司、团体和组织对数据库应用的需求,固然是一种实现数据共享,保障数据一致性、可靠性和可用性的理想结构,但由于其对硬件设施要求很高,实现技术相对复杂,特别是数据的处理、管理和维护的代价十分高昂,除了实力雄厚的金融电信、航空铁路、广播电视等大型企业或政府部门,一般用户难以问津。

更适合中小企业和普通用户开发数据库应用的是客户机/服务器两层结构数据库系统,以及浏览器/服务器多层结构数据库系统或者它们的混合结构。

1.1.4 客户机/服务器两层结构数据库系统

主从式数据库系统中的主机和分布式数据库系统中的每个节点机都安装了完整的应用程序、数据库管理系统和全部或局部数据,是一个通用计算机,既执行数据库管理系统的数据管理功能又执行应用程序的业务处理功能。随着工作站功能的增强和广泛使用,人们开始把 DBMS 的数据管理功能和应用程序的业务处理功能分开,即对网络中某个(些)节点上的计算机安装数据库管理系统软件和数据库,专门用于执行 DBMS 功能,称为数据库服务器,简称服务器(Server);而对其他节点上的计算机仅安装 DBMS 的外围应用开发工具和应用程序,支持用户的业务处理功能,称为客户机(Client)。这就是所谓的客户机/服务器(C/S)两层结构的数据库系统,通常一台服务器要为多个客户端节点服务,如图 1-1-1 所示。

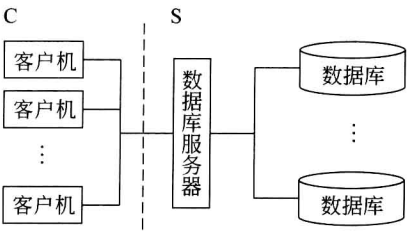


图 1-1-1 C/S 两层结构的数据库系统

C/S 的工作过程是:客户机运行应用程序,通过中间件(如:ODBC、ADO.NET 等)或其他方式连接数据库服务器,并以 SQL 命令的方式将服务请求(查询/更新数据)发送到服务器;数据库服务器运行数据库管理系统软件(如 SQL Server、Oracle、DB2 等)验证用户身份、权限,如通过验证则执行 SQL 命令,对数据库中的相关数据进行处理,然后将处理结果通过网络回传给客户机应用程序,由应用程序作进一步加工(如有返回数据的话)。若没有通过验证或 SQL 命令有误,则返回出错信息。

数据库应用系统客户端的主要功能包括:

- (1) 为用户提供交互式界面,完成数据的输入。
- (2) 格式化数据请求(拼装 SQL 命令)。
- (3) 连接数据库,发出访问请求。
- (4) 显示从服务器收到的数据,实现本地应用(如数据分析、打印等)。

数据库应用系统服务器端的主要功能包括:

- (1) 接收来自客户机的服务请求,并进行安全性控制。
- (2) 进行数据处理(选取、更新数据库中数据)。
- (3) 返回处理结果。

C/S 结构数据库系统的优点在于充分利用了客户端的处理能力及其自治性,把对数据

实施进一步加工的应用逻辑从数据库服务器上分离出来,减轻了服务器的负载,从而扩大了服务器的数据共享规模和事务处理能力,加快了对客户端的响应速度。但这种分离也导致了另一个问题:由于将应用逻辑放在了客户端,对于一些复杂的应用,要求客户机具有完成这些任务的强大功能。特别是当系统规模较大时,需要配备多台高性能的客户机,并为它们重复安装相同的应用程序,这样不仅浪费了系统资源,而且日后系统的升级维护成本也很高。另外,这样的两层结构也存在安全隐患:由于大部分业务逻辑以代码的形式分散安装在用户的客户机上,企业的业务机密很容易被泄露。此外,每台客户机都可以对服务器上的数据进行直接操作,也容易产生安全性漏洞。

三层及多层结构的数据库系统较好地解决了以上问题。

1.1.5 三层结构数据库系统

三层结构的数据库系统将数据处理过程进一步细化为界面层(又叫表示层)、业务逻辑层(又叫应用逻辑层或功能层)和数据层三部分。界面层主要提供用户访问数据库的操作界面;业务逻辑层负责业务逻辑的实现,同时也是连接界面层和数据层的桥梁,它响应界面层的用户请求,从数据层抓取数据,执行业务处理,并将必要的数据传送给界面层展示给用户;数据层完成数据的访问、存储、查询优化、事务管理、数据完整性、安全性控制和故障恢复等任务。三层结构的数据库系统如图 1-1-2 所示。

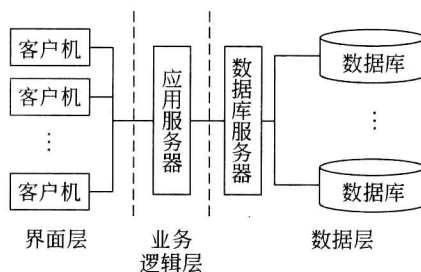


图 1-1-2 三层结构的数据库系统

三层结构的数据库系统在两层结构的基础上增加了业务逻辑层,客户机不需要很高的配置,就可以通过简单友好的界面层共享应用服务器上的应用程序;更新应用程序也只需要在一两台应用服务器上进行即可,从而降低了信息系统开发和维护的成本。此外,该结构下业务逻辑代码被放在了应用服务器上,客户机与数据库服务器不直接相连,而是通过中间层进行统一调度,这些都大大提高了系统的安全性。

三层结构中的每一层都可以根据实际需要继续细分而成为多层结构。如可以将数据层再细分为数据访问层和数据库层,以下“网上多媒体音乐商店”的设计采用的就是这样的结构,在第 2 章和第 5 章的实验 5.6 中将进行详细介绍。

1.1.6 浏览器/服务器多层结构数据库系统

随着互联网技术的发展,出现了浏览器/服务器(Browser/Server,B/S)的多层结构数据库系统。浏览器(亦即 Web 浏览器)是一种运行在客户机上的软件,通过它可以找到并浏览 Internet 上各种 Web 服务器上的网页。常用的浏览器有 Internet Explorer、Mozilla Firefox 和 Netscape Navigator 等,浏览器软件的版本越高,支持的网页效果也就越多。B/S 结构中的服务器由 Web 站点和多种服务器资源组成。Web 站点是由众多网页组成的 Web 页集合,由 Web 服务器进行管理,其他服务器资源可包括应用服务器、邮件服务器和数据库服务器等。因此 B/S 是一个多层结构的数据库系统,如图 1-1-3 所示。

Web 服务器又称 WWW 服务器或 HTTP 服务器。常用的 Web 服务器软件包括:

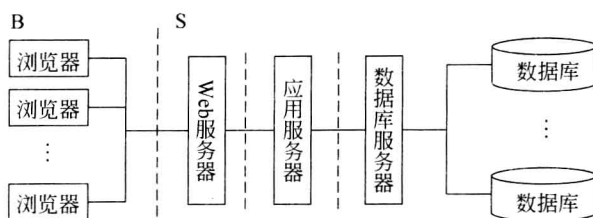


图 1-1-3 B/S 多层结构的数据库系统

Microsoft Internet Information Server (IIS)、Netscape Enterprise Server、Sun ONE Web Server 和 Apache HTTP Server 等,其主要功能是响应来自 Web 浏览器的请求并为其提供 Web 页面。

B/S 结构下实施一次数据库访问的工作流程如下:

(1) 用户通过浏览器向 Web 服务器提出访问请求,如查询某歌手最新出版的专辑信息。

(2) Web 服务器接受用户的请求,由于该请求是一个涉及数据库访问的动态页面,于是将其传递给应用服务器。

说明:所谓动态网页,是指该网页文件不仅具有 HTML 标记,而且含有需要应用服务器或具有应用程序扩展功能的 Web 服务器执行的程序代码。之所以称其为“动态”,是因为此种网页的部分甚至全部内容会随着应用程序处理结果的不同而不同。

对于用户请求的静态页面,即保存在 Web 站点中,以 .htm 或 .html 为扩展名的内容固定不变的页面,Web 服务器将直接用 HTTP 协议将该页面回送给浏览器。

(3) 应用服务器执行动态页中的程序,通过专门的程序接口与数据库相连,并以 SQL 命令的方式将服务请求发送到数据库服务器。

(4) 数据库服务器验证用户身份、权限,如通过验证则执行 SQL 命令,对数据库中的相关数据进行处理,如在数据库相关表中找出该歌手最新出版的全部专辑。

(5) 数据库服务器将查询结果返回给应用服务器。

(6) 应用服务器将查询到的数据插入到动态页中,然后传送给 Web 服务器。

(7) Web 服务器将加工后的标准格式的 HTML 页发送回请求该页面的浏览器。

B/S 结构是 C/S 结构的继承和发展,除了具有 C/S 结构的许多优点外,它还可使客户机更加“苗条”(thin),几乎不需要专门的客户端软件和额外的培训及维护。因为客户端任何经过授权的计算机只要安装了通用的浏览器软件就可以访问 Web 服务器,进而通过应用服务器、数据库服务器访问数据库,而风格统一的浏览器界面早已为广大用户所熟知,并且浏览器软件可以在网上免费下载和升级。这些优势使得 B/S 结构成为 Internet 和 Intranet 环境下开发数据库应用的主流技术,也为本书的应用实例所使用。

当然,B/S 结构也有其不足之处,如 HTTP 协议的低可靠性有可能造成应用故障,特别是对于管理者来说,采用浏览器方式进行系统的维护既不安全也不方便。因此,在实际应用中可将 C/S 与 B/S 结构结合起来使用,如:一些需要用 Web 处理的、为大多数访问者提供信息服务的功能界面(如注册、信息发布、信息查询等)采用 B/S 结构,后台只需少数人员使用的功能界面(如数据库的管理和维护、数据的分析和统计、报表制作等)采用 C/S 结构。

另外需要说明的是,B/S 结构中的所谓“多层”既可以是物理意义上的多层,也可以是逻辑意义上的多层。在本书的实验中,我们所构建的多层 B/S 结构实际上是一个逻辑结构,即所有的服务器软件都被放在了同一台计算机上。

1.2 数据库应用系统的开发步骤

一个 B/S 体系结构的数据库应用系统的开发,要完成后台服务器上数据库的建立与维护以及面向前台浏览器用户的 Web 应用程序开发两大部分的任务。对于前者要求建立起尽量满足一致性、完整性、安全性要求的数据库,而对于后者则要求应用程序具备功能完备、操作简便、界面友好等特点。

作为一种软件系统,数据库应用系统的开发遵循软件工程的规范,即通常要经过系统需求分析、系统设计、系统实现和系统运行与维护几个阶段。

1.2.1 系统需求分析

所谓系统需求分析,就是通过对现实世界要处理的事物进行详细的调研并与工作人员进行充分的沟通,来深入了解原系统的业务流程,准确把握用户的各种需求,从而确定所要开发的数据库应用系统的功能要求、性能要求、运行环境要求和将来可能的扩充要求等。

据本章参考文献[7]给出的统计数据,软件中的错误大约有 15%是软件需求分析的错误,更有高达 50%的软件开发失败缘于软件需求的不合理。这一惊人的现实使人们认识到,只有正确把握必须“做什么”,才谈得上解决“如何做”的问题。

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、处理要求和安全性与完整性要求。需求分析的最终结果是“软件需求分析说明书”,它包含了所开发系统完整的数据定义和处理说明,可以用数据流图和数据字典来表达。

1. 数据流图

数据流图(data flow diagram,DFD)是一种用于表达系统数据和数据处理间关系的逻辑模型工具。由图 1-2-1 可见,数据流图的基本元素有 4 种:“数据源点/数据终点”代表在系统以外的事物或人,它表达了该系统数据的外部来源或要到达的目的地,用矩形表示;“数据处理”代表数据的加工过程,用圆角正方形或圆形表示;“数据存储”可以是数据库基本表或其他形式的数据组织,用三边框形或两条平行线表示;“数据流”代表数据在系统中流动的方向,用带有说明文字的箭头表示。数据流来自数据源点、数据处理或数据存储等元素,可以指向数据终点、数据处理或数据存储等元素,它将系统中的各元素有机地连接在一起。

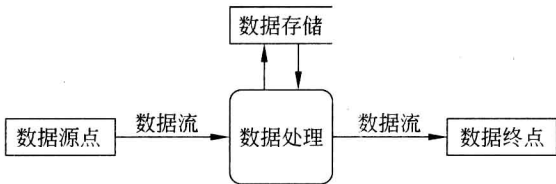


图 1-2-1 数据流图

2. 数据字典

由于数据流图只是从宏观上给出了所开发系统数据与处理间的关系,更细化的描述要借助于数据字典。数据字典(data dictionary,DD)是对数据流图的补充和解释,主要描述数据项、数据流和数据存储的详细内容以及加工逻辑。数据项是组成数据流和数据存储的最小单位。加工逻辑可以用文字表达,也可借助结构化语言、判定表或判定树等描述工具。关于描述工具的使用,可参考本章参考文献[7]。

数据字典通常包括:数据项、数据结构、数据流、数据存储和处理过程五个部分。其中数据项是数据的最小组成单位,若干个数据项可以组成一个数据结构,数据字典通过对数据项和数据结构的定义来描述数据流、数据存储的逻辑内容。

1.2.2 系统设计

通过需求分析,系统必须“做什么”已经清楚了,接下来的系统设计要解决系统“怎么做”的问题。系统设计包括数据库结构设计和软件结构设计两方面的内容。数据库结构设计是指根据系统需求分析的结果,进行数据库模式及子模式等设计;软件结构设计则要解决用户如何操作数据库中数据的问题,又称数据库的行为设计。软件的结构决定了实现系统功能的程序是由哪些模块组成及这些模块相互间的调用关系。

面向数据库服务器的数据库设计和面向浏览器的 Web 应用程序设计,尽管二者在系统实现时所采用的开发语言(前者采用 SQL 数据库语言,后者采用 C# 等面向对象语言)和处理方式(前者是非过程化的处理,后者是过程化的处理)截然不同,但却有着密不可分的联系,在进行系统设计时一定要综合加以考虑。

1. 数据库结构设计

数据库结构设计简称数据库设计,又分为概念结构设计、逻辑结构设计和物理结构设计三个阶段。

1) 概念结构设计

在概念结构设计阶段,现实世界要处理的事物所涉及的各种对象被抽象为实体(Entity),对象之间的关联被抽象为联系(Relationship),使用实体-联系图(E-R图)的概念模型来表达这种抽象。

在 E-R 图中,用矩形表示实体。在矩形框内写明实体名,通常实体带有属性(Attributes,即原对象的特征),并有一个或一组称为码(Key)、能够唯一标识实体的特殊属性。用椭圆形来表示实体的属性,并用无向边将它们与实体相连。用实体名及其属性名集合来抽象和刻画同类实体。同类实体的集合称为实体集。用菱形表示实体集之间的联系,在菱形框内写明联系名,并用无向边连接实体集,同时,在无向边旁标明联系的类型。

两个实体集之间的联系有三种类型,即若实体集 X 中的每一个实体至多与实体集 Y 中的一个实体相对应(相联系),且反之亦然,则称实体集 X 与实体集 Y 间具有一对一的联系,记为 1:1。若实体集 X 中的每一个实体可以与实体集 Y 中的多个实体相对应,而实体集 Y 中的每一个实体至多与实体集 X 中的一个实体相对应,则称实体集 X 与实体集 Y 间具有一对多的联系,记为 1:n。由此不难得出实体集 X 与实体集 Y 间多对多联系(m:n)的定义应该是:实体集 X 中的每一个实体可以与实体集 Y 中的多个实体相对应,且反之亦然。实体集之间的三种联系如图 1-2-2 所示。

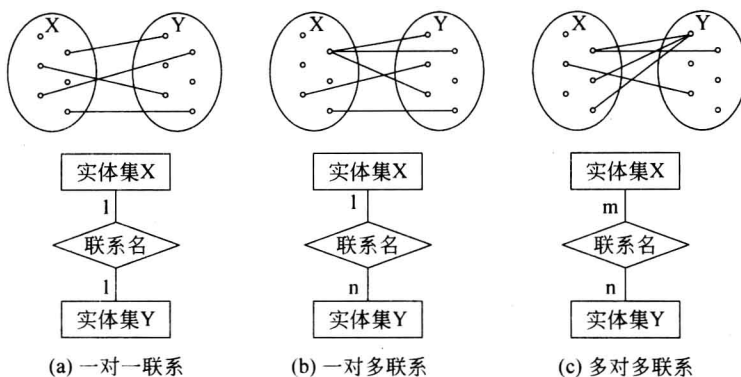


图 1-2-2 实体集的三种联系类型

需要注意的是,当描述实体集的属性比较多时,为了避免 E-R 图过于凌乱,一般不建议直接把属性画在实体集旁边,而应采用在 E-R 图下加文字说明的方式。除了实体集,联系也经常会有与多个实体集相关的属性,但它们的个数一般较少,可以直接画在联系的旁边。

另外,在进行 E-R 图设计时,一定要充分利用好前期系统需求分析时所获得的成果,它们对抽象出 E-R 图中的实体和联系十分有利。如数据字典中的数据存储、数据结构往往可直接转换为实体集,构成它们的数据项自然就成为实体集的属性。

2) 逻辑结构设计

逻辑结构设计就是要把在概念结构设计阶段获得的概念模型(即 E-R 图)转化为关系数据模型的基本表。这一转化应遵循以下原则:

(1) 一个实体集转换为一个关系模式。关系的属性就是实体集的属性,关系的码就是实体型集的码。

(2) 一个 $m:n$ 联系转换为一个关系模式。关系的属性是与该联系相连的各实体集的码以及联系本身的属性,关系的码是与该联系相连的各实体集码的组合。

(3) 一个 $1:n$ 联系可以转换为一个独立的关系模式,也可以与 n 端实体集对应的关系模式合并。如果转换为一个独立的关系模式,则关系的属性是与该联系相连的各实体集的码以及联系本身的属性,关系的码为 n 端实体集的码;如果与 n 端实体集对应的关系模式合并,则只要将 1 端实体集的码和联系本身的属性加入 n 端关系中即可,合并后关系的码保持不变。因后者可以减少数据库中的关系的个数,一般情况下更倾向于采用这种方法。

(4) 一个 $1:1$ 联系可以转换为一个独立的关系模式,也可以与任意一端实体集对应的关系模式合并。如果转换为一个独立的关系模式,则关系的属性是与该联系相连的各实体集的码以及联系本身的属性,关系的码可以从与该联系相连的各实体集的码中任选其一。换句话说,各实体集的码均是该关系的候选码;若与某一端实体集对应的关系模式合并,则只需将另外一端实体集的码和联系本身的属性加入即可,合并后关系的码保持不变。

说明:在不影响语义的前提下,本书后文将实体集简写为实体。

(5) 具有相同码的关系模式可以合并。

依据以上转换原则获得的关系模式(即基本表)是否是一个没有异常弊病的“好”表,可以用关系数据库理论的范式标准进行衡量和优化。一般要求基本表最好能够达到第三范式

(3NF),具体做法可参考本章参考文献[3]。另外,从数据处理的方便性、灵活性、高效性和安全性的角度考虑,是否需要以基本表为基础创建子模式(即用户视图)或组织存储过程,将视系统功能设计的需要来决定。

3) 物理结构设计

逻辑设计给出了将 E-R 图转换为关系模式的一般准则,并未涉及具体的关系数据库管理系统(RDBMS)软件。物理设计的任务就是要为逻辑数据模型选取一个最适合应用环境的物理存储结构,即确定数据库的存储空间分配、空间增长策略、数据文件与事务日志文件的存放路径等。它们的设置与所选取的 RDBMS 软件有直接关系。另外,还应以适当的方式对性能提供保证,如对于数据量较大,且主要用于查询、更新较少的数据表在查询关键字项上创建必要的索引等。

2. 软件结构设计

软件结构设计又分为概要设计和详细设计两个阶段。

1) 概要设计

概要设计是从需求分析的工作结果出发,采用自顶向下、逐步求精的方法,把一个复杂问题分解和细化为由许多模块组成的层次结构。每个模块完成一个子功能,各模块既相对独立又相互关联,成为一个有机整体,共同完成系统的全部功能。概要设计也可以称为功能设计。

如果说在数据库概念结构设计阶段,抽象实体和确定实体之间关系时数据字典起了很大作用的话,则在软件概要设计阶段,数据流图会对我们进行功能模块划分及确定模块与模块、模块与基本表间的调用关系带来很大的方便。通常,数据流图中的一个或多个数据处理可以自然地映射成功能模块,由数据流引导的模块间往往构成调用关系,而连接数据处理的数据存储恰好反映出模块与数据库基本表的关系。

功能模块图是表达软件结构的常用工具。如图 1-2-3 所示,功能模块图中的矩形表示模块,连线表示模块间的调用关系。

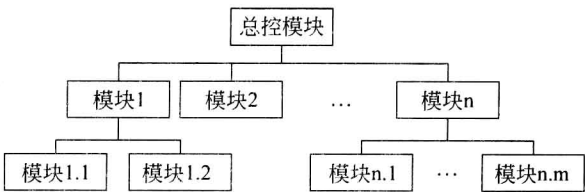


图 1-2-3 功能模块图

2) 详细设计

软件详细设计阶段的关键任务是确定如何具体地实现所要求的目标,也就是要设计出完成指定任务的程序“蓝图”,可以用程序流程图、N-S 图和 PAD 图等来表达(详见本章参考文献[7])。在之后的系统实现阶段,程序员将依此编写出实际的应用程序代码。设计人机界面是详细设计的另一项重要内容。一个“友好的”人机界面应该至少满足可用性、灵活性、健壮性和安全性的要求。为了提高可用性,对输入项应尽可能多地通过选择方式或及时有效的提示,来减轻用户记忆的负担和避免用户因不了解正确的输入要求而导致的失败。对已经更改的内容,在正式提交前,用户应可以选择是否放弃而让数据恢复原状。即便是已经