

全国计算机等级 考试二级教程



教育部考试中心

——MySQL 数据库程序设计 (2017年版)

高等教育出版社



全国计算机等级考试二级教程

——MySQL 数据库程序设计

(2017 年版)

Quanguo Jisuanji Dengji Kaoshi Erji Jiaocheng
——MySQL Shujuku Chengxu Sheji

教育部考试中心

主编 黄 靖

参编 袁 枚 覃爱民 桂 浩

高等教育出版社·北京

内容简介

本书根据教育部考试中心最新颁布的《全国计算机等级考试二级MySQL数据库程序设计考试大纲(2015年版)》编写而成,内容紧扣考试大纲,取舍得当,是一本系统的考试教材。

全书共分十四章,内容包括:数据库技术的基本概念与方法、MySQL编程语言、数据定义、数据查询、数据更新、索引、视图、触发器、事件、存储过程与存储函数、访问控制与安全管理、备份与恢复、MySQL数据库的应用编程和开发实例等。书中各章后配有一定的思考与练习题,包括选择题、填空题、编程题和简答题,附录部分配有各章思考与练习题的参考答案。

图书在版编目(CIP)数据

全国计算机等级考试二级教程:2017年版.MySQL数据库程序设计 / 教育部考试中心编. --北京:高等教育出版社,2016.10
ISBN 978-7-04-046558-7

I. ①全… II. ①教… III. ①电子计算机-水平考试-教材②关系数据库系统-程序设计-水平考试-教材
IV. ①TP3

中国版本图书馆 CIP 数据核字(2016)第 235619 号

策划编辑 何新权 责任编辑 何新权 封面设计 张志奇 版式设计 童 丹
责任校对 高 歌 责任印制 韩 刚

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街 4 号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	河北新华第一印刷有限责任公司		http://www.hepmall.com
开 本	787mm×1092mm 1/16		http://www.hepmall.cn
印 张	15.75	版 次	2016 年 10 月第 1 版
字 数	380 千字	印 次	2016 年 10 月第 1 次印刷
购书热线	010-58581118	定 价	33.00 元
咨询电话	400-810-0598		

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 46558-00

积极发展全国计算机等级考试 为培养计算机应用专门人才、促进信息 产业发展作出贡献

(序)

中国科协副主席 中国系统仿真学会理事长
第五届全国计算机等级考试委员会主任委员
赵沁平

当今,人类正在步入一个以智力资源的占有和配置,知识生产、分配和使用为最重要因素的知识经济时代,也就是小平同志提出的“科学技术是第一生产力”的时代。世界各国的竞争已成为以经济为基础、以科技(特别是高科技)为先导的综合国力的竞争。在高科技中,信息科学技术是知识高度密集、学科高度综合、具有科学与技术融合特征的学科。它直接渗透到经济、文化和社会的各个领域,迅速改变着人们的工作、生活和社会的结构,是当代发展知识经济的支柱之一。

在信息科学技术中,计算机硬件及通信设施是载体,计算机软件是核心。软件是人类知识的固化,是知识经济的基本表征,软件已成为信息时代的新型“物理设施”。人类抽象的经验、知识正逐步由软件予以精确地体现。在信息时代,软件是信息化的核心,国民经济和国防建设、社会发展、人民生活都离不开软件,软件无处不在。软件产业是增长快速的朝阳产业,是具有高附加值、高投入高产出、无污染、低能耗的绿色产业。软件产业的发展将推动知识经济的进程,促进从注重量的增长向注重质的提高方向发展。软件产业是关系到国家经济安全和文化安全,体现国家综合实力,决定 21 世纪国际竞争地位的战略产业。

为了适应知识经济发展的需要,大力促进信息产业的发展,需要在全民中普及计算机的基本知识,培养一批又一批能熟练运用计算机和软件技术的各行各业的应用型人才。

1994 年,国家教委(现教育部)推出了全国计算机等级考试,这是一种专门评价应试人员对计算机软硬件实际掌握能力的考试。它不限制报考人员的学历和年龄,从而为培养各行业计算机应用人才开辟了一条广阔的道路。

1994 年是推出全国计算机等级考试的第一年,当年参加考试的有 1 万余人,2012 年报考人数已达 549 万人。截至 2013 年年底,全国计算机等级考试共开考 38 次,考生人数累计达 5 422 万人,有 2 067 万人获得了各级计算机等级证书。

事实说明,鼓励社会各阶层人士通过各种途径掌握计算机应用技术,并通过等级考试对他们的能力予以科学、公正、权威性的认证,是一种比较好的、有效的计算机应用人才培养途径,符合我国的具体国情。等级考试同时也为用人单位录用和考核人员提供了一种测评手段。从有关公司对等级考试所作的社会抽样调查结果看,不论是管理人员还是应试人员,对该项考试的内容和

形式都给予了充分肯定。

计算机技术日新月异。全国计算机等级考试大纲顺应技术发展和社会需求的变化,从 2010 年开始对新版考试大纲进行调研和修订,在考试体系、考试内容、考试形式等方面都做了较大调整,希望等级考试更能反映当前计算机技术的应用实际,使培养计算机应用人才的工作更健康地向前发展。

全国计算机等级考试取得了良好的效果,这有赖于各有关单位专家在等级考试的大纲编写、试题设计、阅卷评分及效果分析等多项工作中付出的大量心血和辛勤劳动,他们为这项工作的开展作出了重要的贡献。我们在此向他们表示衷心的感谢!

我们相信,在 21 世纪知识经济和加快发展信息产业的形势下,在教育部考试中心的精心组织领导下,在全国各有关专家的大力配合下,全国计算机等级考试一定会以“激励引导成才,科学评价用才,服务社会选材”为目标,服务考生和社会,为我国培养计算机应用专门人才的事业作出更大的贡献。

前 言

随着我国计算机应用的进一步普及和深入,人们已经达成了一个共识:计算机知识是当代人类文化的重要组成部分;计算机应用能力是跨世纪人才不可缺少的素质。因此,许多单位把计算机知识和应用能力作为考核、录用工作人员的重要条件;许多人也在努力证实自身在这方面的实力。人们都在寻找一个统一、客观、公正的衡量标准,教育部考试中心组织的“全国计算机等级考试”自1994年举办以来,应试人数逐年递增,深受社会各界欢迎。

随着计算机应用的发展,等级考试的内容也在不断更新。根据教育部考试中心最新颁布的《全国计算机等级考试二级 MySQL 数据库程序设计考试大纲》,我们编写了这本教程。本书紧扣考试大纲,内容取舍得当,是一本系统的考试教材。

全书共分十四章,内容包括:数据库技术的基本概念与方法、MySQL 编程语言、数据定义、数据查询、数据更新、索引、视图、触发器、事件、存储过程与存储函数、访问控制与安全管理、备份与恢复、MySQL 数据库的应用编程和开发实例等。本书力求在体系结构上安排合理、重点突出、难点分散,便于读者由浅入深逐步掌握;在语言叙述上注重概念清晰、逻辑性强、通俗易懂、便于自学。本书在讲解各理论知识的同时,列举了相应的例题,这些例题均在 MySQL 5.5 的环境下运行通过。根据等级考试要求,考试分为笔试和上机考试两部分。书中各章后配有一定的思考与练习题,包括选择题、填空题、编程题和简答题。其中,编程题与各章中的例题相似,以便读者上机练习;简答题则分别对应于各章的基本知识要点。为便于读者自我检查,附录部分配有各章思考与练习题的参考答案。

本书由教育部考试中心组织编写,并由武汉理工大学黄靖老师主编,第二章、第十一章至第十四章及附录由黄靖编写,第一章和第六章由北京联合大学袁枚老师编写,第三章至第五章由首都经贸大学覃爱民老师编写,第七章至第九章由武汉大学桂浩老师编写。

华中科技大学卢炎生教授对本书进行了全面审阅。在本书的编写和出版过程中,教育部考试中心和高等教育出版社给予了大力支持,在此一并表示衷心感谢。

由于水平有限,错误在所难免,敬请广大读者批评指正,以便我们修订改正。

编者

目 录

第一章 数据库技术的基本概念与方法	1	3.1.4 删除数据库	33
1.1 数据库基础知识	1	3.2 定义表	33
1.1.1 数据库相关的基本概念	1	3.2.1 MySQL 常用的数据类型	34
1.1.2 数据库系统的特点	2	3.2.2 创建表	35
1.1.3 数据库系统的结构	4	3.2.3 查看表	39
1.1.4 数据模型	5	3.2.4 修改表	41
1.2 关系数据库	8	3.2.5 重命名表	45
1.2.1 基本概念	8	3.2.6 删除表	45
1.2.2 基本性质	10	3.3 数据的完整性约束	46
1.3 数据库设计基础	10	3.3.1 定义实体完整性	46
1.3.1 数据库设计的步骤	10	3.3.2 定义参照完整性	50
1.3.2 关系数据库设计的方法	11	3.3.3 用户定义的完整性	53
1.4 MySQL 概述	13	3.3.4 更新完整性约束	55
1.4.1 MySQL 系统特性	13	思考与练习	57
1.4.2 MySQL 服务器的安装与配置	14	第四章 数据查询	59
1.4.3 MySQL 服务器的启动与关闭	15	4.1 SELECT 语句	59
1.4.4 MySQL 客户端管理工具	15	4.2 单表查询	62
思考与练习	17	4.2.1 选择字段	62
第二章 MySQL 编程语言	19	4.2.2 选择指定记录	65
2.1 结构化查询语言 SQL	19	4.2.3 对查询结果排序	74
2.2 MySQL 语言组成	19	4.2.4 限制查询结果的数量	76
2.3 MySQL 函数	22	4.3 分组聚合查询	77
2.3.1 聚合函数	22	4.3.1 使用聚合函数查询	77
2.3.2 数学函数	24	4.3.2 分组聚合查询	78
2.3.3 字符串函数	25	4.4 连接查询	81
2.3.4 日期和时间函数	26	4.4.1 交叉连接	82
2.3.5 其他函数	27	4.4.2 内连接	82
思考与练习	29	4.4.3 外连接	86
第三章 数据定义	30	4.5 子查询	88
3.1 定义数据库	30	4.5.1 带 IN 关键字的子查询	88
3.1.1 创建数据库	30	4.5.2 带比较运算符的子查询	90
3.1.2 选择与查看数据库	31	4.5.3 带 EXISTS 关键字的子查询	92
3.1.3 修改数据库	32	4.6 联合查询(UNION)	94

思考与练习	97	7.7 查询视图数据	128
第五章 数据更新	100	7.8 对视图的进一步说明	128
5.1 插入数据	100	思考与练习	129
5.1.1 插入完整的数据记录	100	第八章 触发器	130
5.1.2 为表的指定字段插入数据	101	8.1 触发器	130
5.1.3 同时插入多条数据记录	102	8.2 创建触发器	130
5.1.4 插入查询结果	103	8.3 删除触发器	132
5.1.5 使用 REPLACE 语句插入表 数据	104	8.4 使用触发器	132
5.2 修改数据记录	105	8.5 对触发器的进一步说明	134
5.2.1 修改特定数据记录	105	思考与练习	134
5.2.2 修改所有数据记录	105	第九章 事件	136
5.2.3 带子查询的修改	106	9.1 事件	136
5.3 删除数据记录	107	9.2 创建事件	136
5.3.1 删除特定数据记录	107	9.3 修改事件	138
5.3.2 带子查询的删除	108	9.4 删除事件	138
5.3.3 删除所有数据记录	108	思考与练习	139
思考与练习	109	第十章 存储过程与存储函数	140
第六章 索引	111	10.1 存储过程	140
6.1 索引概述	111	10.1.1 创建存储过程	141
6.2 查看数据表上所建立的索引	113	10.1.2 存储过程体	143
6.3 创建索引	114	10.1.3 调用存储过程	148
6.3.1 使用 CREATE TABLE 语句创建 索引	114	10.1.4 删除存储过程	148
6.3.2 使用 CREATE INDEX 语句创建 索引	117	10.2 存储函数	149
6.3.3 使用 ALTER TABLE 语句创建 索引	118	10.2.1 创建存储函数	149
6.4 删除索引	118	10.2.2 调用存储函数	150
6.5 对索引的进一步说明	119	10.2.3 删除存储函数	151
思考与练习	120	思考与练习	151
第七章 视图	121	第十一章 访问控制与安全管理	152
7.1 视图概述	121	11.1 用户账号管理	152
7.2 创建视图	122	11.1.1 创建用户账号	152
7.3 删除视图	124	11.1.2 删除用户	154
7.4 修改视图定义	125	11.1.3 修改用户账号	154
7.5 查看视图定义	126	11.1.4 修改用户口令	155
7.6 更新视图数据	126	11.2 账户权限管理	156
		11.2.1 权限的授予	156
		11.2.2 权限的转移与限制	159
		11.2.3 权限的撤销	160
		思考与练习	161

第十二章 备份与恢复	162	13.3.1 编程步骤	176
12.1 数据库备份与恢复的概念	162	13.3.2 建立与 MySQL 数据库服务器的 连接	176
12.2 MySQL 数据库备份与恢复的 方法	163	13.3.3 选择数据库	179
12.2.1 使用 SQL 语句备份和恢复表 数据	163	13.3.4 执行数据库操作	180
12.2.2 使用 MySQL 客户端实用程序 备份和恢复数据	166	13.3.5 关闭与数据库服务器的连接	188
12.2.3 使用 MySQL 图形界面工具备份 和恢复数据	169	思考与练习	189
12.2.4 直接复制	170	第十四章 开发实例	191
12.3 二进制日志文件的使用	171	14.1 需求描述	191
12.3.1 开启日志文件	171	14.2 系统分析与设计	191
12.3.2 使用 mysqlbinlog 实用工具处理 日志	172	14.3 数据库设计与实现	192
思考与练习	173	14.4 应用系统的编程与实现	195
第十三章 MySQL 数据库的应用编程 ..	174	思考与练习	205
13.1 PHP 概述	174	附录 1 考试指导	206
13.2 PHP 编程基础	174	附录 2 全国计算机等级考试二级 MySQL 数据库程序设计 考试大纲(2015 年版)	216
13.3 使用 PHP 进行 MySQL 数据库 应用编程	176	附录 3 MySQL 的安装与配置	219
		附录 4 思考与练习参考答案	227
		参考文献	241

第一章 数据库技术的基本概念与方法

数据库技术诞生于 20 世纪 60 年代末期。经过几十年的发展,数据库相关的理论研究、应用技术都有了非常大的发展。数据库技术是信息系统的一项核心技术,已经成为现代计算机系统的重要组成部分。

1.1 数据库基础知识

数据库技术是计算机科学的重要分支,它能有效地帮助一个组织或企业科学、有效地管理各类信息资源。如今,作为信息系统基础与核心的数据库技术得到了广泛的应用,越来越多的应用领域都采用数据库技术进行数据的存储与处理。数据库的建设规模、数据库信息量的规模及使用频度已成为衡量一个企业、组织乃至一个国家信息化程度的重要标志。

1.1.1 数据库相关的基本概念

数据、数据库、数据库管理系统、数据库系统是与数据库技术最为密切相关的基本概念。

1. 数据

通常这样定义数据:描述事物的符号记录称为数据(Data)。数据有多种表现形式,可以是包括数字、字母、文字、特殊字符组成的文本数据,也可以是图形、图像、动画、影像、声音、语言等多媒体数据。例如,日常生活、工作中使用的客户档案记录、商品销售记录等都是数据。各种形式的数据经过数字化处理后可存入计算机,以便于进一步加工、处理、使用。

在现实世界中,人们可直接用中文、英文等自然语言描述客观事物、交流信息。在计算机中,为了存储和处理客观事物,需要抽象出所关注的事物特征。例如,在客户档案中,人们关注客户的姓名、性别、年龄、籍贯、所在城市、联系电话等特征,那么由这些具体的特征值所构成的一组数据就构成一条记录。例如,(张三,男,25,北京,上海,10012345678)就表示客户张三的信息。

需要注意的是,仅有数据记录往往不能完全表达其内容的含义,需要经过解释。例如,对于上面的客户记录,了解其含义的人会得到这样的信息:张三是男性,今年 25 岁,北京人,目前居住上海,他的联系电话是 10012345678。而不了解数据含义的人,则难以直接从北京、上海两个地名理解所表达的意思。所以,数据以及关于该数据的解释是密切相关的。数据的解释是对数据含义的说明,也称为数据的语义。数据与其语义密不可分,没有语义的数据是没有意义和不完整的。

2. 数据库

所谓数据库(DataBase, DB),是长期储存在计算机内的、有组织的、可共享的数据集合。过去,针对一个具体应用项目,人们收集并抽取出所需的大量数据,保存起来以供进一步加工处理及使用,这些数据往往以文件的形式存放在文件柜里。如今,随着信息技术的迅猛发展,数据量

急剧增加,人们借助计算机和数据库技术科学地保存和管理大量复杂的数据,以便能方便、快捷、充分地利用这些宝贵的信息资源。例如,把客户的档案记录、客户订购的商品信息、商品库存等数据有序地组织并存放在计算机内,构造客户订单的数据库,能够为企业的经营活动提供高效、准确的业务数据支持。

数据库中的数据按一定的数据模型组织、描述和存储,具有较小的冗余度、较高的数据独立性,系统易于扩展,并可以被多个用户共享。

3. 数据库管理系统

数据库管理系统(DataBase Management System,DBMS)是位于操作系统与用户之间的一层数据管理软件,是数据库系统的核心。DBMS按照一定的数据模型科学地组织和存储数据,能够高效地获取数据,提供安全性和完整性等统一控制机制,有效地管理、维护数据,是数据库系统的核心。DBMS的主要功能包括数据定义、数据操纵、数据库建立和维护、数据库运行管理等。

(1) 数据定义

DBMS提供数据定义语言(Data Definition Language,DDL)。用户通过DDL可以对数据库中的数据对象进行定义。

(2) 数据操纵

DBMS提供数据操纵语言(Data Manipulation Language,DML)。使用DML,能够操纵数据,实现对数据库的基本操作,例如数据的查询、插入、删除和修改等。

(3) 数据库的建立和维护功能

数据库的建立和维护功能主要包括数据库初始数据的输入、转换、数据库的转储、恢复、数据库的重组功能 and 性能监视、分析等。这些功能通常是由一些实用程序来完成的。

(4) 数据库的运行管理

数据库的建立、运用和维护等工作由DBMS统一管理、统一控制,以保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

(5) 提供方便、有效存取数据库信息的接口和工具

编程人员可通过程序开发工具与数据库接口编写数据库应用程序。数据库管理员(DataBase Administrator,DBA)可通过相应的软件工具对数据库进行管理。

4. 数据库系统

数据库系统(DataBase System,DBS)是指引入数据库技术的计算机系统。一个完整的数据库系统不仅包含数据库,还包含支持数据库的硬件、数据库管理系统及相关软件、数据库管理员和用户。在不引起混淆的情况下,常常将数据库系统简称为数据库。

1.1.2 数据库系统的特点

数据库系统主要具有以下几个方面的特点。

1. 数据结构化

在数据库系统中,数据不再针对某一应用,而是面向全局应用,具有整体的结构化。这里所说的“结构”是指数据的组织方式。不仅数据是结构化的,而且存取数据的方式也很灵活,可以存取数据库中的某一个数据项、一组数据项、一个记录或一组记录,并且数据库中的结构化数据由DBMS统一管理。DBMS既管理数据的物理结构,也管理数据的逻辑结构;既考虑数据本身,

也考虑数据之间的联系。

2. 数据冗余度小

数据库系统从整体和全局上看待和描述数据,数据不仅面向某个应用,而且面向全局应用,从而可大大减少数据冗余,节约存储空间,避免因数据的重复存储和不同拷贝而造成数据之间的不一致性。例如,学生学籍管理、学校图书管理、校园卡管理等系统都要使用学生的信息,即学生信息被这些系统共享,由数据库管理系统统一进行管理。这种数据共享节约了存储空间,避免数据之间的不相容性与不一致。

3. 数据共享性好

同样,由于数据库系统是从整体和全局上看待和描述数据,使得数据不仅面向某个应用,而且面向整个系统,因此数据可以被多个用户和多个应用共享使用。数据共享可以大大减少数据冗余,节约存储空间,还能够避免由于数据冗余造成的同一数据重复存储而导致修改时的困难和可能造成数据的不一致。

4. 数据独立性高

数据独立性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数据库中的数据是相互独立的;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。也就是说,数据的逻辑结构改变了,用户程序也可以不变。数据独立性是数据库的一种特征和优点,它有利于在数据库结构改变时能保持应用程序尽可能地不改变或少改变,这样可减少应用人员的开发工作量。

5. 数据库保护

数据库管理系统具有对数据的统一管理和控制功能,主要包括数据的安全性、完整性、并发控制与故障恢复等,即数据库保护。

(1) 数据的安全性

数据的安全性(Security)是指保护数据,以防止不合法的使用而造成数据的泄密和破坏,使每个用户只能按规定对某些数据以某些方式进行使用和处理,即保证只有赋予权限的用户才能访问数据库中的数据,防止对数据的非法使用。

(2) 数据的完整性

数据的完整性(Integrity)是对数据的正确性、有效性和相容性要求,即控制数据在一定的范围内有效或要求数据之间满足一定的关系,保证输入到数据库中的数据满足相应的约束条件,以确保数据有效、正确。例如,确保“性别”的取值只能是“男”或“女”。

(3) 并发控制

并发控制(Concurrency)是指当多个用户的并发进程同时存取、修改数据库时,可能会发生相互干扰而得到错误结果,并使得数据库的完整性遭到破坏,因而对多用户的并发操作加以控制和协调。例如,网上购买火车票的应用系统必须要确保不会由于多用户同时购买相同的车票而造成冲突、错误,即必须有并发控制的能力。

(4) 故障恢复

计算机产生的硬件故障、操作员的失误以及人为的破坏都会影响数据库中数据的正确性,甚至造成数据库部分或全部数据的丢失,DBMS 必须具有将数据库从错误状态恢复到某一已知的正确状态的功能,这就是数据库的故障恢复(Recovery)。

1.1.3 数据库系统的结构

在一个组织的数据库系统中,有各种不同类型的人员(或用户)要与数据库打交道。他们从不同的角度以各自的观点来看待数据库,且所处的立足点也不同,从而形成了数据库系统不同的视图结构。因此,考察数据库系统的结构可以有多种不同的层次或不同的视角。若从数据库用户视图的视角来看,数据库系统通常采用三级模式结构,这是数据库管理系统内部的系统结构;若从数据库管理系统的角度来看,数据库系统的结构分为集中式结构、分布式结构、客户/服务器结构和并行结构,这是数据库系统的外部体系结构;若从数据库系统应用的角度来看,目前数据库系统常见的结构有客户/服务器结构和浏览器/服务器结构,这是数据库系统整体的运行结构。这里,主要介绍客户/服务器结构与浏览器/服务器结构。

1. 客户/服务器结构

在数据库系统中,数据库的使用者(如 DBA、程序设计者)可以使用命令行客户端、图形化界面管理工具或应用程序等来连接数据库管理系统,并可以通过数据库管理系统查询和处理存储在底层数据库中的各种数据。数据库系统的这种工作模式采用的就是客户/服务器(Client/Server,C/S)结构。在这种结构中,命令行客户端、图形化界面管理工具或应用程序等称为“客户端”“前台”或“表示层”,主要完成与数据库使用者的交互任务;而数据库管理系统则称为“服务器”“后台”或“数据层”,主要负责数据管理。这种操作数据库的模式也称为客户/服务器(C/S)模式,图 1.1 给出了这种工作模式的一般处理流程。

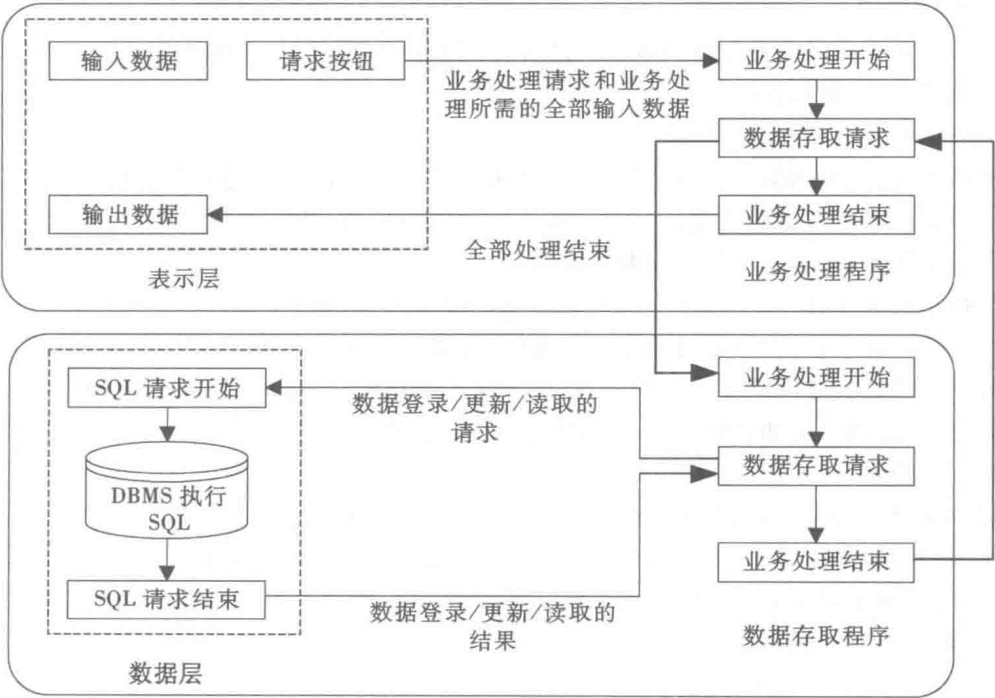


图 1.1 客户/服务器(C/S)模式的一般处理流程

在客户/服务器(C/S)模式中,客户端和服务端可以同时工作在同一台计算机上,这种工作方式称为“单机方式”;也可以“网络方式”运行,即服务器被安装和部署在网络中某一台机器上,而客户端被安装和部署在网络中不同的一台或多台主机上。客户端应用程序的开发,目前常用的工具主要有 Visual C++、.NET 框架、Delphi、Visual Basic 等。

2. 浏览器/服务器结构

浏览器/服务器(Browser/Server,B/S)结构是一种基于 Web 应用的客户/服务器结构,也称为三层客户/服务器结构。在数据库系统中,它将与数据库管理系统交互的客户端进一步细分为“表示层”和“处理层”。其中,“表示层”是数据库使用者的操作和展示界面,通常是用于上网的各种浏览器,由此减轻数据库系统中客户端的工作负担;而“处理层”也称为“中间层”,则主要负责处理数据库使用者的具体应用逻辑,它与后台的数据库管理系统共同组成功能更加丰富的“胖服务器”。数据库系统的这种工作模式被称为浏览器/服务器(B/S)模式,图 1.2 给出了这种工作模式的一般处理流程。

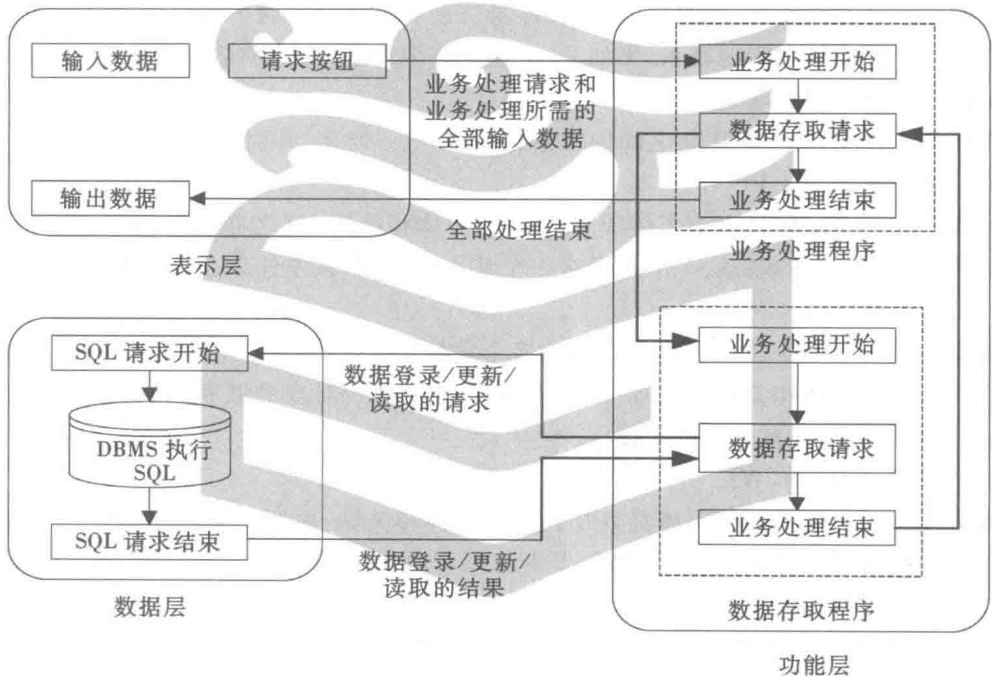


图 1.2 浏览器/服务器(B/S)模式的一般处理流程

目前,开发基于浏览器/服务器结构的数据库应用系统,主要使用的开发语言有 PHP、Java、Peal、C#等。

1.1.4 数据模型

数据库中的数据是有一定结构的,这种结构用数据模型(Data Model)表示。根据不同的应用目的,数据模型可以分为概念模型、逻辑模型和物理模型。

1. 概念模型

概念模型 (Conceptual Model) 用来描述现实世界的事物, 与具体的计算机系统无关。现实世界是存在于人脑之外的客观世界。在设计数据库时, 用概念模型来抽象、表示现实世界的各种事物及其联系。最典型的概念模型是实体联系 (Entity-Relationship, E-R) 模型。实体联系模型用“实体-联系”图表示, 简称 E-R 图。

客观存在并可相互区别的事物称为实体 (Entity)。实体可以是实际的事物, 也可以是抽象的概念, 如商品、学生、部门、课程、比赛等都可以作为实体。

实体的某种特性称为实体的属性 (Attribute)。一个实体可以由多个属性描述。例如, 学生具有学号、姓名、性别、出生日期等特性, 也就是说学生实体具有学号、姓名、性别、出生日期等属性。每个学生是一个实体, 所有学生构成一个实体集。

在现实世界中, 事物内部的特性及各事物之间是有关系的。这些关系称为实体内部的联系及实体之间的联系 (Relationship)。实体内部的联系通常是指实体各属性之间的联系, 例如确定了身份证号, 就一定能知道与之对应的姓名, 即身份证号与姓名这两个属性之间有联系。实体之间的联系是指不同实体之间的联系, 例如, 一个班有许多的学生, 一个学生只属于一个班级, 学生与班级这两个实体之间有联系。

设有两个实体集 A 、 B , 实体集之间的联系有一对一、一对多、多对多三种类型。

(1) 一对一联系 (1:1)

实体集 A 中一个实体最多与实体集 B 中一个实体相关联, 反之亦然。例如, 系与系主任两个实体的联系是一对一的联系, 一个系只有一个系主任, 一个系主任只在一个系任职。

(2) 一对多联系 (1:N)

实体集 A 中的一个实体与实体集 B 中的多个实体相关联, 但实体集 B 中的一个实体至多与实体集 A 中的一个实体相关联。例如, 班级与学生之间是一对多的联系, 即每个班级包含多个学生, 但是每个学生只能属于一个班级。

(3) 多对多联系 (M:N)

实体集 A 中的一个实体与实体集 B 中的多个实体相关联, 而实体集 B 中的一个实体也可以与实体集 A 中的多个实体相关联。例如, 学生与课程两个实体之间是多对多的联系, 即一个学生可以选修多门课程, 而每门课程可以有多个学生选修。

通常, 使用 E-R 图 (即实体-关系图) 来描述现实世界的概念模型, 即描述实体、实体的属性、实体之间的联系。例如, 图 1.3 是一个描述学生、课程、班级等实体之间联系的 E-R 图, 其中表示实体、联系、属性的图形含义如下:

- 实体: 用矩形表示, 矩形框内写明实体的名称。
- 属性: 用椭圆形表示, 并用无向边将其与相应的实体连接起来。
- 联系: 用菱形表示, 菱形框内写明联系的名称, 并用无向边分别与有关实体连接起来, 同时在无向边旁标上联系的类型 (1:1、1:N 或 M:N)。如果一个联系具有属性, 则这些属性也要用无向边与该联系连接起来。例如图 1.3 中连接在“选课”联系上的“成绩”就是联系的属性。

说明: 由于学生、课程等实体的属性较多, 图 1.3 适当简化, 部分属性没有标出。

2. 逻辑模型

逻辑模型 (Logical Model) 是具体的 DBMS 所支持的数据模型。任何 DBMS 都基于某种逻辑

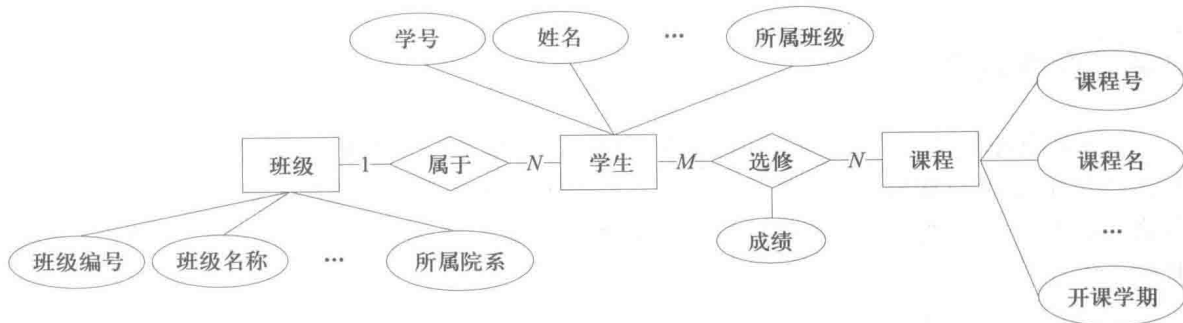


图 1.3 E-R 图示例

数据模型。主要的逻辑数据模型有层次模型(Hierarchical Model)、网状模型(Network Model)、关系模型(Relational Model)、面向对象模型(Object-Oriented Model)等。本书主要介绍关系数据库的概念、技术与方法,这里只简单介绍其他几类逻辑数据模型的基本概念。

(1) 层次模型

层次模型是数据库系统最早使用的一种数据模型,它的数据结构是一棵“有向树”,树的每个结点对应一个记录集,也就是现实世界的实体集。层次模型的特点是:有且仅有一个结点没有父结点,它称作根结点;其他结点有且仅有一个父结点。我们所熟悉的组织机构就是典型的层次结构。但现实世界实体之间的联系有很多种,层次模型难以表达实体之间比较复杂的联系。

(2) 网状模型

网状模型以网状结构表示实体与实体之间的联系。网状模型是层次模型的扩展,允许结点有多于一个父结点,并可以有一个以上的结点没有父结点。现实世界中实体集之间的联系很复杂,网状模型可以方便地表示实体间各种类型的联系,既可以表示从属的联系,也可以表示数据间的交叉联系,但结构复杂,实现的算法难以规范化。

(3) 关系模型

关系模型是用二维表结构来表示实体及实体间联系的模型,并以二维表格的形式组织数据库中的数据。目前流行的商用数据库多是基于关系模型的。支持关系模型的数据库管理系统称为关系数据库管理系统,例如 MySQL 就是一个关系数据库管理系统。有关关系模型的叙述见 1.2 节。

(4) 面向对象模型

尽管关系模型简单灵活,但是对于现实世界中一些复杂的数据结构,很难用关系模型描述。面向对象方法与数据库相结合所构成的数据模型称为面向对象模型。面向对象模型既是概念模型又是逻辑模型。面向对象数据模型用面向对象的观点来描述现实世界实体的逻辑组织、对象间的联系,其表达能力丰富,具有对象可复用、维护方便等优点,是正在发展的数据模型,也是数据库的发展方向之一。

3. 物理模型

物理模型用于描述数据在存储介质上的组织结构。每一种逻辑数据模型在实现时都有与其相对应的物理数据模型。物理数据模型不但由 DBMS 的设计决定,而且与操作系统、计算机硬件密切相关。物理数据结构一般都向用户屏蔽,用户不必了解其细节。

1.2 关系数据库

关系数据库是目前应用最广泛的数据库,它以关系模型作为逻辑数据模型,采用关系作为数据的组织方式,其数据库操作建立在关系代数的基础上,具有坚实的数学基础。关系数据库具有较高的数据独立性,当数据的存储结构发生变化时,不会影响应用程序,这样,能大大减少系统维护的工作量。

1.2.1 基本概念

关系的数据结构就是二维表。无论是实体,还是实体之间的联系,都用关系表示。从用户角度看,关系数据库以二维表格的形式组织数据,例如表 1.1 为一张学生信息登记表。

表 1.1 学生基本信息

学号	姓名	性别	出生日期	籍贯	民族	班号	身份证号
2013110101	张晓勇	男	1997-12-11	山西	汉	AC1301	xxx1
2013110103	王一敏	女	1996-03-25	河北	汉	AC1301	xxx2
2013110201	江山	女	1996-09-17	内蒙古	锡伯	AC1302	xxx3
.....							

这里以表 1.1 为示例,介绍关系模型中几个重要的基本概念。

1. 表

表(Table)也称为关系,由表名、构成表的各个列(如学号、姓名等)及若干行数据(各个学生的具体信息)组成。每个表有一个唯一的表名,表中每一行数据描述一个学生的基本信息。表的结构称为关系模式,例如表 1.1 的关系模式可以表达为:

学生基本信息(学号,姓名,性别,出生日期,籍贯,民族,班号,身份证号)

需要说明的是:以上关系模式的名称和字段名称均使用中文,但在实际的数据库应用系统中,一般不采用中文作为表名、字段名等。因为在编写数据库应用程序时,表名、字段名会作为变量名,因而使用中文标示不方便,而且更重要的是有些数据库管理系统不能很好地支持中文的表名和字段名。因此,本书中所有的数据库名、表名、字段名等均不使用中文标示,而是使用表 1.2 中的英文字段名。

表 1.2 学生信息表 tb_student 的结构定义

含义	字段名	数据类型	宽度
学号	studentNo	字符型	10
姓名	studentName	字符型	20
性别	sex	字符型	3
出生日期	birthday	日期型	
籍贯	native	字符型	20