

21世纪高等教育计算机规划教材

COMPUTER

数据库系统原理 及应用 (SQL Server 2012)

Principles and Applications of
Database Systems

袁丽娜 主编

王刚 罗琼 副主编

案例驱动教学

理论与实践结合

提升实际应用能力



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

数据库系统原理 及应用 (SQL Server 2012)

Principles and Applications
of Database Systems

■ 袁丽娜 主编

■ 王刚 罗琼 副主编



人民邮电出版社

北京

图书在版编目 (CIP) 数据

数据库系统原理及应用 : SQL Server 2012 / 袁丽娜主编. — 北京 : 人民邮电出版社, 2015.9
21世纪高等教育计算机规划教材
ISBN 978-7-115-25547-1

I. ①数… II. ①袁… 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(2015)第150978号

内 容 提 要

本书理论联系实际,系统地介绍了数据库系统的基本原理,并通过案例详细介绍了数据库技术的应用。本书内容循序渐进、条理清晰,并提供大量习题及案例。

全书共分两篇:第I篇为数据库理论篇,着重介绍数据库系统概述、关系数据库、关系数据库标准语言SQL、关系规范化理论、数据库设计和数据库安全保护等数据库技术的基本原理;第II篇为数据库实践指导篇,以SQL Server 2012作为开发环境,通过案例介绍了数据库技术的实践,包括数据库基础操作实践、数据库开发实践和数据库综合实践,数据库基础操作实践涉及数据库和数据表的基本操作、完整性约束的实现、数据的增删改查操作、索引、视图、数据库设计,数据库开发实践主要介绍了T-SQL编程、触发器、存储过程、数据库的备份与恢复等高级应用,数据库综合实践提供了综合案例。全书数据库理论知识和实验指导相互补充,理论和实际紧密结合。

本书适合作为高等院校计算机、软件工程及相关专业的教材,同时也可作为从事计算机软件相关工作的开发人员和研究人员的参考用书。

-
- ◆ 主 编 袁丽娜
 - 副 主 编 王 刚 罗 琼
 - 责任编辑 许金霞
 - 责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京中新伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19 2015年9月第1版
字数: 499千字 2015年9月北京第1次印刷
-

定价: 45.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前 言

随着计算机信息化的不断发展,数据库技术已经属于应用最为广泛的技术之一,已成为各类信息管理的核心技术和重要基础。而数据库课程是计算机、软件工程及相关专业的专业基础课程之一。本书详细介绍了数据库系统的基本原理及相关实践应用,使学生掌握数据库系统原理相关理论知识的同时,并熟练掌握一门主流数据库管理系统(SQL Server 2012)的应用技术,利用常用的开发工具进行数据库应用系统的设计与开发。

本书理论联系实际,主要包括两篇,数据库理论篇和数据库实践指导篇。

数据库理论篇主要介绍数据库系统原理基础理论知识,包括数据库系统概述、关系数据库、关系数据库标准语言 SQL、关系规范化理论、数据库设计和数据库安全保护等。数据库实践指导篇主要介绍在 SQL Server 2012 环境下,数据库实践的各项实验。实践指导篇又分为三大部分:数据库基础操作、数据库开发实践和数据库综合实践。数据库基础操作涉及数据库和表的基本操作、完整性约束的实现、数据的增删改查操作、索引、视图和数据库设计;数据库开发实践主要介绍了 T-SQL 编程、触发器、存储过程及数据库的备份与恢复等高级应用;数据库综合实践提供了综合案例。在学生学习数据库理论的同时,通过实践部分的讲解,为学生提供了数据库应用开发实践指导。

本书具有以下几个特点。

(1) 本书内容系统全面,同时包含数据库理论知识和实验指导,理论和实践内容相互融合、相互补充。

(2) 本书实践采用较新的 SQL Server 2012 作为开发环境,采用实际等案例作为实验驱动,以案例贯穿整个实验内容。实验中还特别提供了案例场景、最佳实践等实践经验,采用深入浅出、循序渐进的方式进行讲解,每个实验都具有典型性,且实践性强。实践篇最后还给出了一个综合案例,将所有知识进行融合,便于读者学习和实践,以达到理论联系实际、真正学以致用教学目的。

(3) 本书还提供了丰富的习题和练习,供读者进行举一反三的学习。

(4) 为方便读者学习和教师授课,本书还提供了 PowerPoint 电子课件,读者可以登录人民邮电出版社教学服务与资源网(www.ptpedu.com.cn)进行下载。

本书由袁丽娜担任主编,王刚、罗琼担任副主编。其中第 1 章和部分数据库基础操作实验由王刚编写,第 3 章、部分数据库基础操作实验和综合实验由罗琼编写,第 2 章、第 4 章、第 5 章、第 6 章和部分数据库基础操作实验、数据库开发实验由袁丽娜编写。本书在编写过程中还得到了很多领导及老师的帮助,在此表示衷心的感谢。

本书在编写过程中,由于时间仓促,编者水平有限,书中个别地方难免会有疏漏,敬请广大读者批评指正。

编 者

2015 年 5 月

目 录

第 I 篇 数据库理论篇

第 1 章 数据库系统概述 2	第 3 章 关系数据库标准语言 SQL38
1.1 数据、数据库的基本概念..... 2	3.1 SQL 概述..... 38
1.2 数据库技术的产生、发展..... 3	3.2 SQL Server 2012 简介..... 39
1.3 数据库系统的体系结构..... 4	3.2.1 SQL Server 的发展与版本..... 40
1.3.1 内部体系结构..... 4	3.2.2 SQL Server 2012 主要组件..... 41
1.3.2 外部体系结构..... 6	3.3 数据定义..... 42
1.4 数据库管理系统..... 7	3.3.1 创建及使用数据库..... 43
1.5 数据模型..... 8	3.3.2 创建及使用数据表..... 46
1.5.1 数据模型概念..... 8	3.4 数据操作..... 51
1.5.2 层次数据模型..... 8	3.4.1 数据查询..... 51
1.5.3 网状数据模型..... 9	3.4.2 数据更新..... 58
1.5.4 关系数据模型..... 10	3.5 视图..... 60
1.5.5 面向对象数据模型..... 11	3.5.1 创建视图..... 60
1.6 数据库技术展望..... 11	3.5.2 修改视图..... 61
习题 13	3.5.3 删除视图..... 61
第 2 章 关系数据库 17	3.5.4 查询视图..... 61
2.1 关系模型的概述..... 17	3.5.5 更新视图..... 61
2.2 关系模型的数据结构及定义..... 19	3.6 索引..... 62
2.2.1 关系的定义及相关概念..... 19	3.6.1 索引的概念及作用..... 62
2.2.2 关系的码..... 20	3.6.2 索引的分类..... 62
2.2.3 关系的性质..... 22	3.6.3 创建索引..... 63
2.2.4 关系模式和关系数据库..... 23	3.6.4 删除索引..... 64
2.3 关系的完整性..... 24	3.7 数据控制..... 64
2.3.1 实体完整性..... 24	习题 65
2.3.2 参照完整性..... 24	第 4 章 关系规范化理论 67
2.3.3 用户自定义完整性..... 25	4.1 规范化问题的提出..... 67
2.4 关系操作..... 25	4.2 规范化..... 69
2.4.1 关系代数..... 25	4.2.1 函数依赖..... 69
2.4.2 关系演算*..... 32	4.2.2 关系模式的范式..... 75
习题 35	

4.3 关系模式的分解.....	81	5.7.1 对数据库性能的监测、分析和改善.....	111
4.3.1 无损连接分解.....	81	5.7.2 数据库的转储和恢复.....	112
4.3.2 保持函数依赖的分解.....	84	5.7.3 维持数据库的安全性和完整性.....	112
习题 85		5.7.4 数据库的重组和重构.....	112
第 5 章 数据库设计.....	87	习题 112	
5.1 数据库设计概述.....	87	第 6 章 数据库安全保护.....	114
5.1.1 数据库设计的特点.....	88	6.1 数据库的安全性控制.....	114
5.1.2 数据库设计方法概述.....	88	6.1.1 数据库安全性概述.....	115
5.1.3 数据库设计的基本步骤.....	90	6.1.2 安全性控制的方法.....	116
5.2 需求分析.....	91	6.1.3 SQL Server 2012 的安全性策略.....	119
5.2.1 需求分析的任务.....	91	6.2 数据库完整性控制.....	123
5.2.2 需求分析的方法.....	92	6.2.1 数据库完整性的概述.....	123
5.3 概念结构设计.....	94	6.2.2 数据库完整性约束条件的分类.....	123
5.3.1 概念结构设计的重要性及特点.....	94	6.2.3 SQL Server 2012 的数据库完整性策略.....	124
5.3.2 概念结构设计的方法和步骤.....	94	6.3 事务和并发控制.....	125
5.3.3 使用 E-R 模型进行概念结构设计.....	96	6.3.1 事务的概念和特点.....	125
5.4 逻辑结构设计.....	105	6.3.2 并发控制概述.....	126
5.4.1 逻辑结构设计的任务.....	105	6.3.3 封锁和封锁协议.....	127
5.4.2 逻辑结构设计的方法和步骤.....	105	6.3.4 活锁和死锁.....	130
5.5 物理结构设计.....	108	6.3.5 SQL Server 2012 的并发控制.....	132
5.5.1 确定物理结构.....	108	6.4 数据库恢复.....	136
5.5.2 评价和优化物理结构.....	110	6.4.1 数据库恢复概述.....	136
5.6 数据库的实施.....	110	6.4.2 数据库恢复技术.....	137
5.6.1 实际数据库结构的建立及数据加载.....	110	6.4.3 数据库故障类型及其恢复策略.....	138
5.6.2 应用程序及数据库的调试和运行.....	111	6.4.4 SQL Server 2012 恢复策略.....	139
5.6.3 文档整理.....	111	习题 140	
5.7 数据库的运行和维护.....	111		

第 II 篇 数据库实践指导篇

第 1 部分 数据库基础操作实践..... 142

实验 1 安装及配置 SQL Server 2012.....	144
实验 1.1 SQL Server 2012 的安装.....	147
实验 1.2 SQL Server 2012 服务的启动与停止.....	153

实验 1.3 熟悉 SQL Server 2012

管理平台.....	154
-----------	-----

实验 2 数据库的基本操作..... 156

实验 2.1 创建数据库及架构.....	158
实验 2.2 修改数据库.....	163

实验 2.3 删除数据库.....	164	实验 10.1 T-SQL 程序设计.....	237
实验 3 表的基本操作.....	166	实验 10.2 使用游标.....	239
实验 3.1 创建别名数据类型.....	169	实验 11 触发器.....	241
实验 3.2 创建表.....	170	实验 11.1 创建触发器.....	242
实验 3.3 修改表结构.....	174	实验 11.2 修改触发器.....	246
实验 3.4 删除表结构.....	175	实验 11.3 删除触发器.....	248
实验 4 完整性约束.....	177	实验 12 存储过程.....	251
实验 5 数据查询.....	186	实验 12.1 创建存储过程.....	252
实验 5.1 单表查询.....	187	实验 12.2 修改存储过程.....	256
实验 5.2 多表查询.....	201	实验 12.3 删除存储过程.....	257
实验 6 数据增删改操作.....	213	实验 13 自定义函数.....	259
实验 6.1 插入数据.....	214	实验 13.1 创建自定义函数.....	260
实验 6.2 修改数据.....	215	实验 13.2 修改自定义函数.....	263
实验 6.3 删除数据.....	216	实验 13.3 删除自定义函数.....	264
实验 7 索引.....	218	实验 14 数据库备份与恢复.....	266
实验 7.1 创建索引.....	219	实验 14.1 数据库的备份.....	268
实验 7.2 优化索引.....	221	实验 14.2 数据库的还原.....	272
实验 7.3 删除索引.....	222	实验 14.3 数据库的分离和附加.....	274
实验 8 视图.....	225	第 3 部分 数据库综合实践.....	278
实验 8.1 视图创建及使用.....	226	实验 15 综合案例——学校管理	
实验 8.2 删除视图.....	228	信息系统的设计与开发.....	278
实验 9 数据库设计.....	231	参考文献.....	296
第 2 部分 数据库开发实践.....	235		
实验 10 T-SQL 编程.....	235		

第 1 章

数据库系统概述

【本章重点】

- 掌握数据、数据库、数据库系统的相关概念；
- 了解数据库技术的产生、发展过程；
- 掌握数据库系统的体系结构；
- 掌握数据模型；
- 了解数据库技术发展趋势。

21 世纪是信息技术飞速发展的时代。作为信息技术主要支柱之一的数据库技术在社会各个领域中有广泛的应用。数据库技术专门研究如何高效地组织和存储数据，如何快速获取和处理数据。对信息进行收集、组织、存储、加工、传播、管理和使用都是以数据库为基础。利用数据库可以为各种用户提供及时的、准确的、相关的信息，满足这些用户的各种不同的需要。因此，数据库课程不仅仅是计算机相关专业的必修课程，也成为很多非计算机专业的必修课程之一。

本章将介绍数据库系统的相关基本概念、数据库技术的产生、发展过程以及数据库系统的体系结构，然后全面分析数据模型，最后介绍关于数据库技术未来发展的趋势。

1.1 数据、数据库的基本概念

随着信息系统在各个行业的广泛使用，我们周围有着很多数据库应用的案例，比如图书馆信息管理系统、OA 系统、财务系统等。在科学计算、过程控制和数据处理这计算机的三大主要应用领域中数据处理所占比例差不多已达到 70%。信息已经成为这个时代最核心的资源，通过对信息的高效利用可以使社会资源得到最大限度的节约和合理运用，在这个过程中数据库起到了极其重要的作用，在学习数据库之前先了解数据库相关的概念非常有必要。

1. 数据

数据 (Data) 是数据库中存储的基本对象。由于早期的计算机系统主要用于科学计算，处理的数据基本都是整数、浮点等数值型数据，因此数据在人们头脑中的第一反应是数字，其实数字只是数据最简单的一种形式。在现代计算机系统中数据的种类非常丰富，比如文本、图像、音频、视频等都是数据。

数据的定义是用来记录信息的可识别的符号组合，是信息的具体表现形式。例如：一个学生的信息可以用一组数据“SS20140204124，张三，男，18，信息工程”来表示。数据必须和语义相结合才有实际意义，例如：上面这组数据根据其赋予的语义，表示的信息是一个姓名叫张三的

学生,学号为 SS20140204124,性别为男,年龄 18 岁,专业为信息工程。

数据的表现形式是多样的,可以用多种不同的数据形式表示同一个信息。例如:“2015 年股市将上涨 50%”“二零一五年股市将上涨 50%”“2015 年股市将上涨百分之五十”,这三种不同的数据表现形式所表达的信息是同一个,并无不同。

2. 数据库

数据库(DataBase, DB)严格的定义是:长期存储在计算机中的有组织的、可共享的大量数据的集合。数据库中的数据按一定的数据模型组织、描述和存储,具有较小的数据冗余、较高的数据独立性和易扩展性,可为多种用户共享。

数据库也可简单地理解为是在计算机中按照一定的格式存储数据的仓库。人们在使用计算机的过程中采集和生成了大量的数据,这些数据都需要保存在计算机中以便将来使用,而且随着计算机技术的不断快速发展和在人类各个领域的广泛应用,需要存储的数据量越来越大,简单的数据存储技术早已不能满足实际需求,这促使了性能强大的数据管理技术——数据库的诞生。

1.2 数据库技术的产生、发展

数据库技术的发展,已经成为先进信息技术的重要组成部分,是现代计算机信息系统和计算机应用系统的基础和核心。根据数据管理手段,数据库技术的发展,可以划分为三个阶段: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 40 年代至 50 年代中期,计算机外部设备只有磁带机、卡片机和纸带穿孔机,而没有直接存取的磁盘设备,也没有操作系统,只有汇编语言,计算机主要用于主要科学计算,数据处理采取批处理的方式,被称为人工管理数据阶段。

这个阶段的特点是数据不保存,数据面向某一应用程序,数据无共享、冗余度极大,数据不独立,完全依赖于程序,数据无结构,应用程序自己控制数据。

2. 文件系统阶段

从 20 世纪 50 年代中期到 69 年代中期,计算机不仅用于科学计算,同时也开始用以信息处理,硬件方面有了很大改进,出现了磁盘、磁鼓等直接存储设备。软件方面出现了高级语言和操作系统,且操作系统中出现了专门的数据管理软件。

这个阶段的数据是以文件形式长期保存的。由文件系统管理数据,文件形式多样化,程序与数据间有一定独立性。由专门的软件即文件系统进行数据管理,程序和数据间由软件提供的存取方法进行转换,数据存储发生变化不一定影响程序的运行。这个阶段仍存在明显缺陷,数据冗余度大,数据一致性差。

3. 数据库系统阶段

进入 20 世纪 60 年代,计算机软件、硬件技术得到了飞速发展。1969 年 IBM 公司研发的层次性信息管理系统(IMS 系统)、美国数据系统语言协会发布的数据库任务组关于网状数据库的报告以及 1970 年 IBM 公司的研究员 E.F.Codd 在发表的论文“大型共享数据库数据的关系模型”中提出的“关系模型”是数据库技术发展史上具有里程碑意义的重大事件,这些研究成果大大促进了数据库管理技术的发展和應用。

这个阶段的数据高度结构化。使用规范的数据模型表示数据结构,数据不再针对某一项应用,

而是面对系统整体,应用程序可通过 DBMS 访问数据库中所有数据。较小的数据冗余,共享性高,数据与应用程序相互独立,通过 DBMS 进行数据安全性和完整性控制。数据库管理系统 DBMS 可以有效地防止数据库中的数据被非法使用或修改。对于完整性控制,DBMS 提供了数据完整性定义方法和进行数据完整性检验的功能。

数据管理三个阶段的比较如表 1-1 所示。

表 1-1 数据管理的三个阶段的比较

	人工管理	文件系统	数据库系统
应用领域	科学计算	科学计算、管理	大规模管理
硬件需求	无直接存取存储设备	磁盘、磁鼓	大容量磁盘
软件需求	没有操作系统	文件系统	数据库管理系统
数据共享	无共享,冗余度极大	共享性差,冗余度大	共享性高,冗余度小
数据独立性	不独立,完全依赖于程序	独立性差	具有高度的物理独立性和逻辑独立性
数据结构化	无结构	记录内有结构,整体无结构	整体结构化,用数据模型描述
数据控制能力	应用程序自己控制	应用程序自己控制	由数据库管理系统提供数据安全、完整性、并发控制和恢复能力

1.3 数据库系统的体系结构

数据库系统 (DataBase System, DBS) 是实现有组织地、动态地存储大量关联数据、方便多用户访问的计算机硬件、软件和数据资源组成的系统,即它是采用数据库技术的计算机系统。数据库系统一般由数据库、数据库管理系统 (及其开发工具)、数据库应用系统、数据库管理员和用户构成。数据库系统常被简称为数据库。数据库系统的组成可以图 1-1 表示。

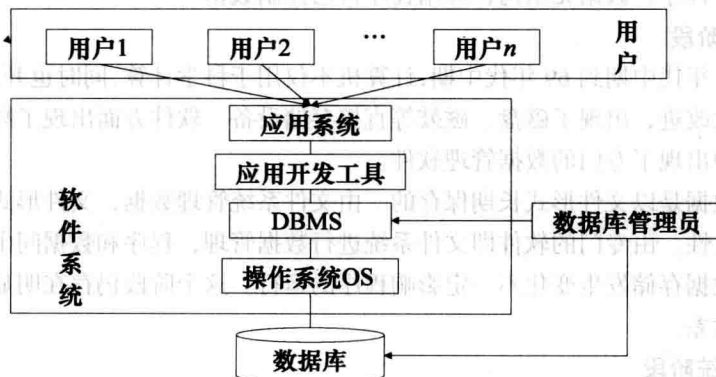


图 1-1 DBS 的组成

1.3.1 内部体系结构

数据库的结构可以从多个角度进行分析,从数据库管理系统的角度看,数据库系统通常采用三级模式结构,这被称为数据库系统的内部体系结构。

根据美国国家标准学会和标准计划与需求委员会提出的建议,将数据库系统的内部体系结构定义为三级模式和二级映像结构,如图 1-2 所示。数据库系统的三级模式之间的联系通过二级映像实现,实际的映像转换工作是由 DBMS 完成。

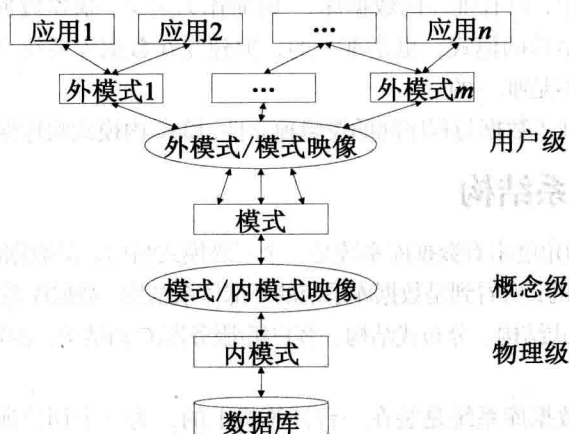


图 1-2 数据库系统的三级模式和二级映像结构

与数据抽象的层次相对应,数据库系统的三级模式分别是外模式、模式和内模式。一个数据库只有一个模式、一个内模式,但可以有多个外模式。数据库系统的三级模式不仅可以使数据具有独立性,而且还可以使数据达到共享,使同一数据能够满足更多用户的不同要求。

1. 外模式

外模式 (External Schema) 也称子模式或用户模式,是数据库在视图层上的数据库模式。它是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述,是数据库用户的数据视图,是与某一应用有关的数据的逻辑表示。

一个数据库可以有多个外模式,因为不同用户有不同的需求,以及拥有不同的访问权限,因此,对不同用户提供不同的外模式的描述,即每个用户只能看见和访问所对应的外模式中的数据。同一外模式可以为多个应用程序使用。例如,学生信息系统中,根据学生需要看到的信息组成一个数据库视图,包含学生的个人信息、所选课程等数据,而教师需要的信息组成另外一个数据库视图,包含教师的个人信息、课表等数据。

2. 模式

模式 (Schema) 也称逻辑模式,是在逻辑层描述数据库的设计。模式是数据库中全体数据的逻辑结构和特征的描述,通常称为数据模式,是所有用户的公共数据视图。模式实际上是数据库数据在逻辑层上的视图。

DBMS 一般提供数据模式描述语言(Data Description Language, DDL)来严格定义数据的逻辑结构、数据之间的联系以及与数据有关的安全性要求、完整性约束等。数据的逻辑结构包括数据记录的名称、组成的数据项的名称、类型、取值范围等。例如,学生信息系统中,有学生关系模式:学生(学号,姓名,性别,出生日期,班级,所在系);教师模式:教师(教师编号,姓名,性别,年龄,职称,所在系);课程模式:课程(课程编号,课程名称,学分,所在系)等。

3. 内模式

内模式 (Internal Schema) 也称存储模式。它是对数据库物理结构和存储方法的描述,是数据在存储介质上的实际保存方式,是数据库最低一级的逻辑描述。例如,数据的存储方式是顺序

存储、还是按照 B 树结构存储等。

一般由 DBMS 提供的内模式 DDL 来定义内模式。内模式对一般用户是透明的，通常不需要关心内模式的具体实现细节，但它的设计会直接影响到数据库的性能。

在一个数据库系统中，只有唯一的数据库，因而作为定义、描述数据库存储结构的内模式和定义、描述数据库逻辑结构的模式，也是唯一的，但建立在数据库系统之上的应用则是非常多样的，所以对应的外模式不是唯一的。

外模式/模式映像保证了数据与程序间的逻辑独立性，模式/内模式映像保证了数据的物理独立性。

1.3.2 外部体系结构

从数据库管理系统的角度来看数据库系统是一个三级模式结构，但数据库的这种模式结构对程序员和最终用户来说是透明的，所看到是数据库系统的外部体系结构。数据库系统的外部体系结构有 5 种结构：单用户结构、主从式结构、分布式结构、客户机/服务器(C/S)结构、浏览器/服务器 (B/S) 结构。

1. 单用户结构

单用户结构的整个数据库系统是装在一台计算机上的，为一个用户所独占，不同机器之间不能共享数据，数据冗余度大，是早期和最简单的数据库系统。例如，一个企业的各个部门都使用本部门的机器来管理本部门的数据，各个部门间的机器是相互独立的。由于不同部门之间不能共享数据，因此企业内部存在大量的冗余数据。

2. 主从式结构

主从式结构也称为集中式结构，是一个主机带多个终端用户结构的数据库系统。在这种结构中，包括应用程序、DBMS、数据，都集中存放在主机上，所有处理任务都由主机来完成。各个用户通过主机的终端可同时或并发地存取数据库，共享数据资源。主从式结构的优点是结构简单，易于管理、控制与维护，缺点是当终端用户数目增加到一定程度后，主机的任务会过分繁重，成为瓶颈，使系统性能下降。系统的可靠性依赖主机，当主机出现故障时，整个系统都不能使用。

3. 分布式结构

分布式数据库是数据库技术与网络技术相结合的产物。在实际应用中，一些大型企业和连锁店等经常是在物理位置上分布式存在的，单位中各个部门都维护着自身的数据，整个单位的信息被分解成了若干信息分块，分布式数据库正是针对这种情形建立起来的信息桥梁。

分布式数据库中的数据在逻辑上相互关联，是一个整体，但物理地分布在计算机网络的不同节点上。网络中的每个结点都可以独立处理本地数据库中的数据，执行局部应用，同时也可以通过网络通信系统执行全局应用。

分布式结构的优点是适应了地理上分散的公司、团体和组织对于数据库应用的需求，缺点是数据的分布存放给数据的处理、管理与维护带来困难。当用户需要经常访问远程数据时，系统效率会明显地受到网络传输的制约。

4. 客户机/服务器结构

客户机/服务器 (Client/Server, C/S) 结构也称为 C/S 结构。它将数据库系统看作由两个非常简单的部分组成，一个服务器(后端)和一组客户(前端)。服务器指 DBMS 本身，客户指在 DBMS 上运行的各种应用程序，包括用户编写的应用程序和内置的应用程序。

在 C/S 结构的数据库系统中，客户端具有一定的数据处理、数据表示和数据存储能力，服务器端完成数据库管理系统的核心功能。客户机和服务器两者都参与一个应用程序的处理，可以有效地降低网络通信量和服务器运算量，从而降低系统的通信开销，可以称之为一种特殊的协作式处理模式。在

该体系结构中,客户机向服务器发送请求,服务器响应客户机发出的请求并返回客户机所需要的结果。

C/S 结构的优点是充分利用两端硬件环境的优势,发挥客户端的处理能力,很多工作可以在客户端处理后再提交给服务器,可以有效降低系统的通信开销,缺点是只适用于局域网,客户端需要安装专用的客户端软件,升级维护不方便,并且对客户端的操作系统一般也会有一定限制。

C/S 形式的两层结构中客户机和服务器直接相连,服务器要消耗资源用于处理与客户机的通信。当大量客户机同时提交数据请求时,服务器很有可能无法及时响应数据请求,导致系统运行效率降低甚至崩溃,而且客户机应用程序的分发和协调难于处理。为此,三层结构的 C/S 模式应运而生。

5. 浏览器/服务器结构

浏览器/服务器 (Browser/Server, B/S) 结构也称为 B/S 结构,实质是一个三层结构的客户机/服务器体系。该结构是一种以 Web 技术为基础的新型数据库应用系统体系结构。它把传统 C/S 模式中的服务器分解为一个数据服务器和多个应用服务器(Web 服务器),统一客户端为浏览器。在 B/S 结构的数据库系统中,作为客户端的浏览器并非直接与数据库相连,而是通过应用服务器(Web 服务器)与数据库进行交互。这样减少了与数据库服务器的连接数量,而且应用服务器(Web 服务器)分担了业务规则、数据访问、合法校验等工作,减轻了数据库服务器的负担。

B/S 结构的优点,首先是简化了客户端,客户端只要安装通用的浏览器软件即可。因此,只要有一台能上网的计算机就可以在任何地方进行操作而不用安装专门的客户应用软件,节省客户机的硬盘空间与内存,实现客户端零维护。其次是简化了系统的开发和维护,使系统的扩展非常容易。系统的开发者无须再为不同级别的用户设计开发不同的应用程序,只需把所有的功能都实现在应用服务器(Web 服务器)上,并就不同的功能为各个级别的用户设置权限即可。B/S 结构的缺点,首先是应用服务器(Web 服务器)端处理了系统的绝大部分事务逻辑,从而造成应用服务器运行负荷较重。其次是客户端浏览器功能简单,许多功能不能实现或实现起来比较困难。

目前大多数应用软件系统都是采用 B/S 结构。

1.4 数据库管理系统

在大量的数据中如何能快速找到所需要的数据,并能对庞大的数据库进行日常维护,这就需要使用数据库管理系统(DataBase Management System, DBMS)。这个系统是一种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 DBMS 访问数据库中的数据,数据库管理员也通过 DBMS 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时或不同时刻去建立、修改和询问数据库。大部分 DBMS 提供数据定义语言 (Data Definition Language, DDL) 和数据操作语言 (Data Manipulation Language, DML), 数据控制语言 (Data Control Language, DCL) 供用户定义数据库的模式结构与权限约束,实现对数据的追加、删除等操作。

数据库管理系统是位于用户与操作系统之间的一层数据库管理软件,主要功能是为用户或应用程序提供访问数据库的方法,如图 1-3 所示。

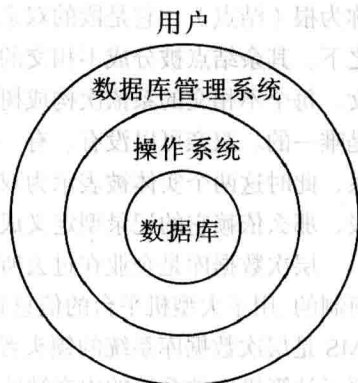


图 1-3 DBMS 在计算机系统的位置

1.5 数据模型

数据(Data)是描述事物的符号记录。模型(Model)是现实世界的抽象。数据模型(Data Model)是数据特征的抽象,是数据库管理的教学形式框架。数据库系统中用以提供信息表示和操作手段的形式构架。数据模型包括数据库数据的结构部分、数据库数据的操作部分和数据库数据的约束条件。

1.5.1 数据模型概念

随着数据库学科的发展,数据模型的概念也逐渐深入和完善。早期,一般把数据模型仅理解为数据结构。其后,在一些数据库系统中,则把数据模型归结为数据的逻辑结构、物理配置、存取路径和完整性约束条件四个方面。现代数据模型的概念,则认为数据结构只是数据模型的组成成分之一。数据的物理配置和存取路径是关于数据存储的概念,不属于数据模型的内容。此外,数据模型不仅应该提供数据表示的手段,还应该提供数据操作的类型和方法,因为数据库不是静态的而是动态的。因此,数据模型还包括数据操作部分。层次模型、网状模型和关系模型是三种重要的数据模型。这三种模型是按其数据结构而命名的。层次模型和网状模型采用格式化的结构。在这类结构中实体用记录型表示,而记录型抽象为图的顶点。记录型之间的联系抽象为顶点间的连接弧。整个数据结构与图相对应。对应于树形图的数据模型为层次模型;对应于网状图的数据模型为网状模型。关系模型为非格式化的结构,用单一的二维表的结构表示实体及实体之间的联系。满足一定条件的二维表,称为一个关系。

1.5.2 层次数据模型

层次数据模型表现为倒立的树,用户把层次数据库理解为段的层次。一个段(Segment)等价于一个文件系统的记录型。在层次数据模型中,文件或记录型之间的联系形成层次。换句话说,层次数据库把记录集合表示成倒立的树结构。层次数据模型如图 1-4 所示。树可以被定义成一组结点,即有一个特别指定的结点称为根(结点),它是段的双亲,其他段都直接在其之下。其余结点被分成不相交的系并作为上面段的子女。每个不相交的系依次构成树和根的子树。树的根是唯一的。双亲可以没有、有一个或者有多个子女。层次模型可以表示两个实体之间的一对多联系,此时这两个实体被表示为双亲和子女关系。树的结点表示记录型。如果把根记录型定义成 0 级,那么依赖它的记录型定义成 1 级,依赖 1 级的记录型称为 2 级,等等。

层次数据库是企业在过去所使用的最老的数据库模型之一。由 IBM 和北美洛克韦尔公司共同研制的、用于大型机平台的信息管理系统是第一个层次数据库。在 20 世纪 70 年代和 80 年代早期,IMS 是层次数据库系统的领头者。层次数据库模型是第一个体现数据库概念的商业化产品,它克服了计算机文件系统的内在缺陷。

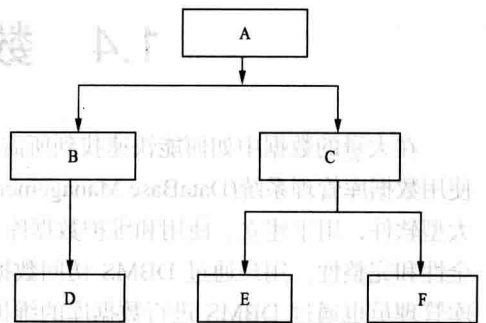


图 1-4 层次数据模型

1. 层次数据模型的优点

(1) 简单。由于数据库基于层次结构，所以各层之间的联系逻辑上（或概念上）简单并且层次数据库的设计也简单。

(2) 数据共享。因为所有数据都保存在公共数据库里，所以数据共享成为现实。

(3) 数据安全。层次模型是第一个由 DBMS 提供和强制数据安全的数据模型。

(4) 数据独立性。DBMS 提供了保持数据独立性的环境，这充分地降低了编程的难度，减少了对程序的维护工作量。

(5) 数据完整性。给定双亲/子女联系，在双亲段和它的子女段之间存在链接。由于子女段是自动地引用它的双亲，所以这种模式保证了数据完整性。

(6) 高效率。当数据库包含大量一对多（1:m）联系的数据并且用户在大量事务中所使用的数据的联系固定时，层次数据模型是非常高效率的。

(7) 可用的技术。由于已有许多大型计算机技术基础，因此经验丰富的编程人员可以加以有效利用。

(8) 可靠的商业应用程序。在主机环境内部存在大量可靠的商业应用程序。

2. 层次数据模型的缺点

(1) 如果改变了层次数据库的数据库结构，那么必须修改所有访问数据库的应用程序。这样，维护数据库和应用程序变得非常困难。

(2) 在现实世界中普通的多对多（n:m）联系在层次数据模型中都很难实现。

(3) 虽然层次数据库概念简单、容易设计而且没有数据依赖性问题，但实现起来特别复杂。DBMS 要求数据存储的物理级知识，数据库设计者必须要具备一定的物理数据存储特性的知识。数据库结构的任何变化，都要求所有访问数据库的应用程序随之改变。

(4) 层次数据模型数据库缺乏灵活性，新的联系或段的改变通常会带来非常复杂的系统管理任务。

(5) 为层次数据模型数据库编写应用程序非常费时和复杂。应用程序编程人员和终端用户必须准确地知道数据库中数据的物理描述以及如何编写访问数据的线性控制代码。只有很少或没有编程技术的普通用户通常是很难掌握这一知识的。

1.5.3 网状数据模型

网状模型是最早出现的网状 DBMS。网状模型中以记录为数据的存储单位。记录包含若干数据项。网状数据库的数据项可以是多值的和复合的数据。每个记录有一个唯一地标识它的内部标识符，它在一个记录存入数据库时由 DBMS 自动赋予。网状数据库是导航式数据库，用户在操作数据库时不但说明要做什么，还要说明怎么做。例如在查找语句中不但要说明查找的对象，而且要规定存取路径。世界上第一个网状数据库管理系统也是第一个 DBMS 是美国通用电气公司 Bachman 等人在 1964 年开发成功的 IDS（Integrated Data Store）。IDS 奠定了网状数据库的基础，并在当时得到了广泛的发行和应用。1971 年，美国数据系统委员会（CODASYL）中的数据库任务组（DBTG）提出了一个著名的 DBTG 报告，对网状数据模型和语言进行了定义，并在 1978 年和 1981 年又做了修改和补充。因此网状数据模型又称为 CODASYL 模型或 DBTG 模型。1984 年美国国家标准学会（ANSI）提出了一个网状定义语言（Network Definition Language, NDL）

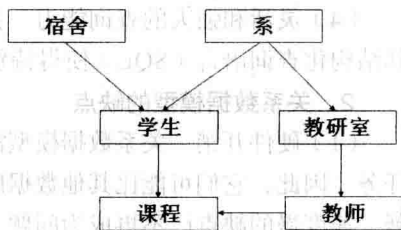


图 1-5 网状数据模型

的推荐标准。在 20 世纪 70 年代, 曾经出现过大量网状数据库的 DBMS 产品。比较著名的有 Cullinet 软件公司的 IDMS、Honeywell 公司的 IDSII、Univac 公司的 DMS1100 和 HP 公司的 IMAGE 等。网状数据库模型对于层次和非层次结构的事物都能比较自然地模拟, 在关系数据库出现之前网状 DBMS 要比层次 DBMS 用得普遍。在数据库发展史上, 网状数据库占有重要地位。图 1-5 所示为一个网状数据模型。

1.5.4 关系数据模型

网状数据库和层次数据库已经很好地解决了数据的集中和共享问题, 但是在数据独立性和抽象级别上仍有很大欠缺。用户在对这两种数据库进行存取时, 仍然需要明确数据的存储结构, 指出存取路径。而后来出现的关系数据库较好地解决了这些问题。关系数据库理论出现于 20 世纪 60 年代末到 70 年代初。1970 年, IBM 的研究员 E. F. Codd 博士发表《大型共享数据银行数据的关系模型》一文提出了关系模型的概念。后来 Codd 又陆续发表多篇文章, 奠定了关系数据库的基础。关系模型有严格的数学基础, 抽象级别比较高, 而且简单清晰, 便于理解和使用。但是当时也有人认为关系模型是理想化的数据模型, 用来实现 DBMS 是不现实的, 尤其担心关系数据库的性能难以接受, 更有人视其为当时正在进行中的网状数据库规范化工作的严重威胁。为了促进对问题的理解, 1974 年 ACM 牵头组织了一次研讨会, 会上开展了一场分别以 Codd 和 Bachman 为首的支持和反对关系数据库两派之间的辩论。这次著名的辩论推动了关系数据库的发展, 使其最终成为现代数据库产品的主流。

关系数据模型提供了关系操作的特点和功能要求, 但不给 DBMS 的语言给出具体的语法要求。对关系数据库的操作是高度非过程化的, 用户不需要指出特殊的存取路径, 路径的选择由 DBMS 的优化机制来完成。Codd 在 20 世纪 70 年代初期的论文论述了范式理论和衡量关系系统的 12 条标准, 用数学理论奠定了关系数据库的基础。Codd 博士也以其对关系数据库的卓越贡献获得了 1981 年 ACM 图灵奖。

关系数据模型是以集合论中的关系概念为基础发展起来的。关系模型中无论是实体还是实体间的联系均由单一的结构类型——关系来表示。在实际的关系数据库中的关系也称表。一个关系数据库就是由若干个表组成。

1. 关系数据模型的优点

(1) 简单。关系数据模型比层次和网状模型更简单, 设计人员不再受到实际物理数据存储细节的约束, 因而可以专注于数据库的逻辑视图。

(2) 结构独立性。与层次和网状模型不同的是, 关系数据模型不依赖于导航式的数据访问系统。数据库结构的变化不会影响数据访问。

(3) 易于设计、实现、维护和使用。关系模型提供结构独立性和数据独立性, 这使得数据库的设计、实现、维护和使用更容易。

(4) 灵活和强大的查询能力。关系数据库模型提供非常强大、灵活和易于使用的查询能力。其结构化查询语言 (SQL) 使得特别的查询成为现实。

2. 关系数据模型的缺点

(1) 硬件开销。关系数据模型需要更高效的计算硬件和数据存储设备来完成 RDBMS 安排的任务。因此, 它们可能比其他数据库更慢。但是, 随着计算机技术和更高效的操作系统的快速发展, 速度慢的缺点已不再成为问题。

(2) 容易设计出不合理或性能低下的数据库。关系数据库易于使用的特点, 容易导致未经训