高校计算机教学系列教材

数据库系统基础教程

姚春龙　丁春欣　姜翠霞　编

北京航空航天大学出版社

http://www.buaapress.com.cn

需要全本请访问 www.ertongbook.com

内容提要

本书系统地介绍了数据库系统的基本概念、原理及其开发与设计技术。全书以 SQL Server2000为背景，讲述包括数据库系统的构成、关系数据库、SQL 语言、关系数据理论、数据库的安全和恢复、数据库设计、查询处理和优化、开发客户机/ 服务器应用所需的数据库访问标准接口等概念和技术。结合当前数据库技术的发展，又讲述了分布式数据库、主动数据库、数据挖掘等知识。同时，本书注重基础知识的描述，具有深入浅出、便于自学的特点。可作为高等院校计算机专业的本、专科生的数据库课程教材或参考书，也可以供从事数据库教学与科研工作的教师和科技工作者参考。

图书在版编目(CIP)数据

数据库系统基础教程/ 姚春龙编 . —北京:北京航空航天大学出版社,2003 .3

ISBN 7 - 81077 - 258 - 9

Ⅰ .数… Ⅱ .姚… Ⅲ .数据库系统—高等学校—教材 Ⅳ . TP311 .13

中国版本图书馆 CIP 数据核字(2002)第 104903 号

数据库系统基础教程

姚春龙 丁春欣 姜翠霞 编

责任编辑 许传安

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100083) 发行部电话:82317024 传真:82328026

http:/ / www .buaapress .com .cn

E-mail: bhpress@263 .net

北京市松源印刷有限公司印装 各地书店经销

*

开本:787×1 092 1/ 16 印张:16 字数:410 千字

2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷 印数:5000 册

ISBN 7 - 81077 - 258 - 9 定价:24 .00 元

总前言

科教兴国,教育先行,在全国上下已形成共识。在教育改革过程中,出现了多渠道、多形式、多层次办学的局面。同时,政府逐年加大教育的投入力度。教育发展了,才能有效地提高全民族的文化、科学素质,使我们中华民族屹立于世界民族之林。

计算机科学与技术的发展日新月异,其应用领域迅速扩展,几乎无处不在。社会发展的需求,促使计算机教育生气蓬勃。从普通高校的系统性教学,到远距离的电视、网上教学;从全面讲述,到不同应用领域的、星罗棋布的培训班;从公办的到民办的;从纸介教材到电子教材等等,可以说计算机教学异彩纷呈。要进行教学,就必须有教材。

面对我们这么大的国家和教学形势,在保证国家教学基本要求的前提下,应当提倡教材多样化,才能满足各教学单位的需求,使他们形成各自的办学风格和特色。为此,我们组织北京工业大学、北京航空航天大学、北京理工大学、南开大学、天津工业大学等高校的有丰富教学经验的教师编写了计算机教学的系列教材,将陆续与师生见面。

系列教材包括以下各项。

(一) 基础理论:离散数学。

(二) 技术基础:电路基础与模拟电子技术;数字逻辑基础;计算机组成与体系结构;计算机语言(拼盘、选择使用),包括 C + + 程序设计基础、Visual Basic 程序设计基础、Matlab 程序设计基础、Java 程序设计基础、Delphi 语言基础、汇编语言基础等;数据结构;计算机操作系统基础;计算方法基础;微机与接口技术;数据库技术基础等。

(三) 应用基础:计算机控制技术;网络技术;软件工程;多媒体技术等。

(四) 技术基础扩展:编译原理与编译构造;知识工程——网络计算机环境下的知识处理。

(五) 应用基础扩展:计算机辅助设计;单片机实用基础;图像处理基础;传感器与测试技术;计算机外设与接口技术。

本系列教材主要是针对计算机教学编写的,供普通高校、社会民办大学、高等职业学校、业余大学等计算机本和专科选用。其中一部分教材也适合非计算机专业本科教学使用。在这部分教材的内容简介或前言中对使用范围均作了说明。

本系列教材在编写时,注重以下几点:(1) 面对计算机科学与技术动态发展的现实,在内容上应具有前瞻性;(2) 面对学以致用,既有系统的基础知识,又有应用价值的实用性;(3) 具有科学性、严谨性。另外,力求排版紧凑,使有限的版面具有最大的信息量,以使读者得到实惠。

能否实现这些愿望,只有师生在教学实践中评价。我们期望得到师生的批评和指正。

高校计算机教学系列教材编委会

前　言

自 20 世纪 60 年代末数据库系统产生以来,数据库技术得到了飞速的发展,已经成为信息管理的重要工具。随着计算机技术,特别是网络技术的飞速发展,数据库的应用领域不断扩大。例如,地理信息系统(GIS)、联机分析与处理系统(OLAP)、管理信息系统(MIS)、企业自愿计划(ERP)、客户关系管理(CRM)、数据仓库和数据挖掘等系统都是以数据库技术作为主要支撑的。随着数据库的推广和使用,计算机应用已深入到工业、农业、金融、商业、军事、科研等各个领域。

面向 21 世纪,信息技术已经成为社会生产的重要组成部分。作为信息资源管理和利用的基础,数据库技术是计算机软件领域的一个重要的分支。因此,数据库课程已经成为计算机专业的一门重要课程。

本书较为系统地介绍了数据库的基本概念、原理、方法和应用技术,特别注重了数据库的开发和设计方面的内容。全书共分十章。

第 1 章介绍了数据库系统的基本概念,包括数据管理及其发展、数据模型、数据库结构与组成,并简要介绍了层次、网状、关系以及面向对象等数据模型。第 2 章介绍关系模型,包括关系的定义、关系代数和关系演算等。第 3 章介绍关系数据标准语言 SQL,包括 SQL 的定义、操作和控制功能的基本语法和使用,另外还介绍了视图和嵌入式 SQL 技术。第 4 章介绍关系数据理论,主要包括数据依赖的定义、各种范式的定义及规范化方法。另外,作为数据库设计的理论基础,特别列举了大量的例题和分析,帮助读者理解关系数据理论的基本概念和应用。第 5 章介绍数据库设计的全过程,包括需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库实施以及数据库的运行和维护,并详细介绍 E－R 数据模型和如何利用 E－R 模型进行数据库设计的方法。第 6 章介绍数据库管理,主要包括数据库的安全保护和完整性控制。第 7 章介绍数据库的恢复技术,包括事务的概念、性质和并发性控制;并详细介绍了数据库的备份与恢复策略。第 8 章介绍关系系统的查询优化技术,包括代数优化和物理优化等技术。第 9 章介绍数据访问技术,包括对 ODBC、JDBC 和 OLE DB 三种标准数据访问接口原理和使用的介绍。第 10 章介绍了数据库技术的一些新技术核心应用,包括分布式数据库系统、主动数据库系统和数据挖掘等。

本书理论与实际系统相结合,相关章节结合了最新的 SQL Server2000 数据库系统进行介绍。作为教材适合于计算机相关专业的本科和专科学生使用,书中加“ * ”的部分可作为专科生的选修内容。为了便于学习和理解,每章都配有适量的习题。除适合教学外,本书还适合从事数据库设计与开发的工程技术人员和计算机爱好者学习有关数据库基础知识。

本书由姚春龙主编,负责内容的取材、组织和审定。全书由姚春龙、丁春欣和姜翠霞编写。

限于水平,书中难免出现欠妥之处,欢迎广大读者批评指正。

编　者

2002 年 10 月

目　　录

第 1 章　数据库系统概述

1.1　数据管理的进展 …… 1

1.1.1　数据管理的三个阶段 …… 1

1.1.2　数据库技术的发展 …… 6

1.2　数据模型 …… 7

1.2.1　数据模型的组成 …… 8

1.2.2　层次数据模型 …… 9

1.2.3　网状数据模型 …… 12

1.2.4　关系数据模型 …… 15

1.2.5　面向对象数据模型 …… 18

1.3　数据库系统 …… 20

1.3.1　数据库系统的组成 …… 20

1.3.2　数据库系统的结构 …… 21

习　　题 …… 25

第 2 章　关系数据库

2.1　关系模型 …… 27

2.1.1　基本概念 …… 27

2.1.2　关系形式化定义 …… 28

2.1.3　关系的完整性 …… 31

2.2　关系代数 …… 33

2.2.1　传统的集合运算 …… 33

2.2.2　专门的关系运算 …… 35

2.2.3　关系代数查询实例 …… 40

2.3　关系演算 …… 41

2.3.1　元组关系演算语言 ALPHA …… 41

2.3.2　域关系演算语言 QBE …… 45

习　　题 …… 50

第 3 章　关系数据库标准语言 SQL

3.1　SQL 语言概况 …… 52

3.2　SQL 语言的数据定义功能 …… 53

3.2.1　基本表的定义 …… 54

3.2.2　基本表的修改与删除 …… 57

3.2.3　索引的建立和撤消 …… 58

3.3　SQL 语言数据操纵功能 …… 59

3.3.1　SQL 数据查询 …… 59

3.3.2　SQL 数据插入 …… 74

3.3.3　SQL 数据删除 …… 75

需要全本请访问 www.ertongbook.com

3.3.4　SQL 数据修改 …… 76
3.4　视　图 …… 77
3.4.1　视图的定义和删除 …… 77
3.4.2　视图的查询 …… 78
3.4.3　视图的更新 …… 78
3.4.4　视图的优点 …… 79
3.5　SQL 语言的数据控制功能 …… 80
3.5.1　权限与角色 …… 80
3.5.2　权限与角色授予和收回 …… 81
3.6　嵌入式 SQL …… 82
3.6.1　嵌入式 SQL 概述 …… 83
3.6.2　嵌入式 SQL 数据库访问过程 …… 84
3.6.3　不使用游标的嵌入式 SQL 语句 …… 85
3.6.4　使用游标的嵌入式 SQL 语句 …… 87
3.6.5　动态 SQL 语句 …… 88
习　　题 …… 90
第 4 章　关系数据理论
4.1　问题的提出 …… 91
4.2　规范化 …… 94
4.2.1　函数依赖 …… 94
4.2.2　码 …… 95
4.2.3　范　式 …… 96
4.2.4　2NF …… 96
4.2.5　3NF …… 98
4.2.6　BCNF …… 98
4.2.7　多值依赖 …… 99
4.2.8　4NF …… 102
4.2.9　规范化小结 …… 103
4.3　* 数据依赖的公理系统 …… 103
4.4　例题及分析 …… 107
习　　题 …… 116
第 5 章　数据库设计
5.1　数据库设计概述 …… 119
5.1.1　数据库设计问题 …… 119
5.1.2　数据库设计方法和步骤 …… 119
5.2　实体-联系模型 …… 121
5.3　E-R 模型向关系模型的转换 …… 126
5.4　E-R 模型设计实例 …… 131
5.4.1　概念结构设计(设计 E-R 模型) …… 131
5.4.2　逻辑结构设计 …… 133

5.4.3 数据库的实施 …… 136
5.4.4 在数据库设计中应注意的问题 …… 138
习 题 …… 139
第6章 数据库管理
6.1 引 言 …… 141
6.2 数据库的完整性 …… 141
6.2.1 数据库完整性分类 …… 142
6.2.2 完整性约束条件与DBMS的完整性控制 …… 143
6.2.3 SQL Server的数据完整性 …… 144
6.2.4 存储过程 …… 148
6.2.5 触发器 …… 149
6.3 数据库的安全性 …… 155
6.3.1 定义视图 …… 155
6.3.2 访问控制 …… 156
6.3.3 数据加密 …… 158
6.3.4 数据库审计 …… 158
6.3.5 SQL Server的安全性机制 …… 158
习 题 …… 172
第7章 数据库的恢复
7.1 事 务 …… 173
7.1.1 事务的概念 …… 173
7.1.2 事务的操作 …… 173
7.1.3 事务的状态 …… 174
7.1.4 事务的性质 …… 175
7.2 SQL的事务管理 …… 176
7.3 数据库恢复及故障 …… 177
7.4 数据库恢复的原理 …… 178
7.4.1 数据转储 …… 178
7.4.2 登记日志文件 …… 179
7.5 数据库恢复技术 …… 179
7.6 SQL Server的备份策略与方案 …… 182
7.7 并发控制 …… 183
7.7.1 并发控制引起的问题 …… 183
7.7.2 封 锁 …… 186
7.7.3 封锁协议 …… 186
7.7.4 活锁与死锁 …… 187
7.7.5 并发调度的可串行性 …… 189
7.7.6 协 议 …… 190
7.7.7 封锁粒度 …… 190

7.7.8　插入和删除操作 …… 191
习　题 …… 193
第 8 章　关系系统的查询优化
8.1　引　言 …… 194
8.2　代数优化 …… 196
8.3　物理优化 …… 200
8.4　代价估算优化 …… 202
8.5　SQL Server 的查询优化 …… 204
习　题 …… 206
第 9 章　数据库访问技术
9.1　ODBC …… 208
9.1.1　ODBC 介绍 …… 208
9.1.2　数据源 …… 211
9.1.3　ODBC API 使用基础 …… 217
9.2　* JDBC …… 220
9.2.1　JDBC 介绍 …… 220
9.2.2　JDBC 的结构 …… 223
9.2.3　JDBC 接口概貌 …… 224
9.3　ADO 简介 …… 225
9.3.1　OLE DB 概述 …… 225
9.3.2　ADO 概述 …… 226
习　题 …… 231
第 10 章　数据库新技术与新应用
10.1　分布式数据库系统 …… 233
10.1.1　分布式数据库的定义及特点 …… 233
10.1.2　分布式数据库系统的环境 …… 234
10.1.3　分布式数据库系统的体系结构 …… 235
10.1.4　分布式数据库系统研究现状及应用前景 …… 238
10.2　主动数据库系统 …… 239
10.2.1　主动数据库的产生及描述 …… 239
10.2.2　主动数据库管理系统 …… 240
10.2.3　主动数据库系统的实现 …… 241
10.2.4　主动数据库的研究进展 …… 242
10.3　数据挖掘技术 …… 243
10.3.1　数据挖掘的概念及挑战 …… 243
10.3.2　数据挖掘的分析方法 …… 244
10.3.3　数据挖掘系统的体系结构及运行过程 …… 245
10.3.4　数据挖掘的应用 …… 246
习　题 …… 247

参考文献 …… 248

第 1 章
数据库系统概述

1.1 数据管理的进展

当今人类社会正步入信息化社会,人们在政治、经济、军事、科学研究、文化教育等各个领域都将产生大量的信息。这些信息需要加工、交流和应用。随着计算机技术的迅速发展,计算机进入了数据处理领域。计算机应用从科学计算进入数据处理,使计算机从少数科学家手中的珍品成为广大科技人员和管理人员广泛使用的有力工具。

数据处理主要包括对各种形式的数据进行收集、存储、加工和传播等一系列活动。从事这些活动的计算机应用一般被称作数据密集型应用(Data Intensive Application)。这类应用具有以下特点。

(1) 数据量大。应用所涉及的数据量无法完全在内存中存放,一般需要存放在辅助存储器中。

(2) 数据需要长期保留。应用程序结束时,数据并不消失,仍然保留在计算机系统中,以便于被继续使用和处理。

(3) 数据被共享。在系统中,数据需要被多个应用程序所共享,可以被多用户同时使用。其共享范围可以是一个部门、单位,甚至更大的范围。

数据密集型应用是最大的计算机应用领域,管理信息系统(MIS)、企业资源计划(ERP)以及客户关系管理(CRM)等应用都涉及这一应用领域。这类应用面临的共同问题是怎样科学、有效地管理这种大量的、持久的共享数据。自从 20 世纪 50 年代末以来,数据管理一直是计算机科学技术研究领域中的研究热点。

1.1.1 数据管理的三个阶段

数据的管理主要完成对数据的分类、组织、编码、存储、检索和维护等功能,这是数据处理的中心问题。几十年来,数据管理经历了三个发展阶段: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 年代中期以前,数据管理处于人工管理阶段。这一阶段计算机主要用于科学计算,只有少数的科学家才能使用计算机。在这一时期,计算机没有直接存储设备,也没有操作系统和数据管理软件,数据处理方式只有批处理方式。这一阶段的特点如下。

(1) 数据不保存。由于计算机主要用于科学计算,一般不需要长期保留数据。在计算某一问题时将数据输入,用完后就撤走这些数据,不仅对用户数据如此,甚至是系统软件有时也如此处理。

(2) 应用程序员管理数据。程序员不仅要规定数据的逻辑结构,而且还要在应用程序中

设计物理结构,包括规定数据的存储结构、存取方法、输入输出方式等。程序员要花费大量的精力去进行数据的物理布置,在很大的程度上加大了程序员的负担。

(3) 数据不共享。如图 1－1 所示,数据是面向应用的,即一组数据对应一个应用程序。对于不同的应用程序,程序员根据应用程序分别建立数据的逻辑和物理结构,即使两个应用程序涉及到某些相同的数据,也必须各自定义。这致使数据无法被多个应用程序所共享,并造成应用程序之间存在着大量的重复数据。

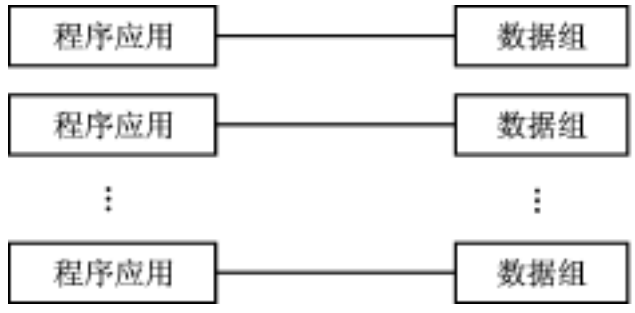

图 1－1　人工管理阶段数据与程序之间的关系

(4) 数据不具有独立性。由于没有数据管理软件对数据进行管理,程序员必须规定数据的存储结构。当数据的存储结构改变时,相应的存取数据的子程序也必须改变,其数据与程序不具有独立性。

2．文件系统阶段

20 世纪 50 年代后期至 60 年代中期,数据管理处于文件系统阶段。这一时期,计算机被大量用于数据处理,出现了磁盘、磁鼓等随机存取设备,操作系统中有专门的数据管理软件,即文件系统。这一阶段数据管理有如下几个特点。

(1) 数据可以长期保存。数据可以以文件的形式长期保存在外存储器上,并通过文件系统进行存取。各种应用程序需要经常对文件进行查询、修改、插入和删除等操作,以适应数据处理的需要。

(2) 由软件(即文件系统)管理数据,程序和数据有了一定的独立性。程序与数据之间通过文件系统提供的存取方法进行转换。操作系统提供打开、关闭、读、写等文件操作命令,程序可以运用这些操作完成对文件的查询、修改等处理。文件的逻辑结构和存储结构由系统进行转换,数据与程序具有了一定的独立性。这样程序员可以专心设计数据的逻辑结构和算法,不必过多地考虑物理细节。

(3) 文件系统与人工管理相比有了很大的进步,但这种方法仍然存在着许多缺点,主要表现如下。

编写应用程序的生产率不高

应用程序的设计者必须对所使用的文件的逻辑及物理结构有清楚的了解,才能编写相应的管理程序。由于文件系统只提供了读、写等几个低级的文件操作命令,因此对文件的查询、修改、插入和删除等操作必须在应用程序内编写相应的程序代码来解决。另外,文件系统还可能只是应用程序在功能上的重复。例如,多个应用程序需要对同一个文件进行相同的查询,由于查询操作是在程序中完成的,因此每个应用程序中都必须编写相同的查询过程。这在很大程度上降低了编程的效率。

数据冗余度(Redundancy)大

文件系统中的文件是面向特定的应用部门而设计的。当不同的应用程序所访问的数据部

分相同时,也必须各自定义所需的文件,这就带来了严重的数据冗余问题。例如,某学校的人事、财务和卫生保健部门都要处理职工的有关信息,但是各部门所处理的信息的内容不尽相同。

人事部门要处理的信息包括:部门、工作证号、姓名、年龄、籍贯、学历、职称、基本工资。为此,人事部门的应用程序员必须建立一个人事信息文件 PF。该文件的记录包括上述的各项数据。

财务部门要处理的信息包括:部门、工作证号、姓名、基本工资、奖金、扣款、实发工资。因此,财务部门必须定义一个工资信息文件 SF。该文件的每一记录包括上述的六项数据。

类似地,卫生保健部门应当建立一个保健信息文件 HF。该文件的每一个记录应当包含下列的数据项:部门、工作证号、姓名、性别、年龄、婚姻状况、身高、体重、健康状况。

这样,三个部门所使用的三个文件都被保存在计算机系统中。显然,PF,SF 和 HF 三个文件所保存的数据中,部门、工作证号、姓名等多个数据项是重复存储的,存在着严重的冗余。

数据冗余不仅浪费存储空间,而且会给数据带来潜在的不一致性(Inconsistency)。由于数据存在多个副本,当发生数据更新时,就很有可能发生某些副本被修改而另一些副本被遗漏的现象。例如,当某个职工的工作部门有所变动时,PF,SF 和 HF 三个文件中对应于该职工记录的部门数据项都必须同时进行修改,否则,将会造成数据的不一致性,影响数据的正确性和可靠性。

数据与程序缺乏独立性

文件系统中的文件是为某一特定的应用服务的。当应用的环境和需求等发生变化时,常常需要修改文件的结构,例如向文件的记录中增加一些字段或重新定义某些字段的宽度等。文件结构的每一次修改都会导致应用程序的修改,以修改文件结构的定义。此外,应用程序的改变,例如所使用的编程语言发生变化,也会影响到文件的数据结构的改变。数据与程序之间缺乏独立性。另外,文件方式不能提供在不同数据文件之间建立联系的能力,因此说文件之间是孤立的,无法反映现实世界事物之间的内在联系。

图 1－2 显示了文件系统阶段数据与程序之间的关系。在图中,虚线表示应用程序所访问的文件。

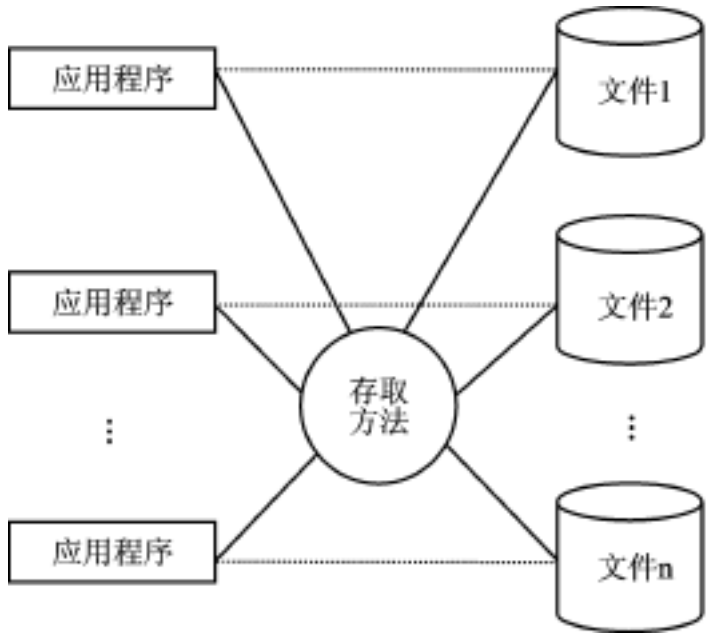

图 1－2　文件系统阶段数据与程序

需要全本请访问 www.ertongbook.com

不支持并发访问(Concurrent Access)

在现代计算机系统中,为了有效地利用资源,一般允许多个程序并发地运行。虽然一个文件可以被不同的应用程序所使用,但是在同一时刻,一个文件只能被一个应用程序所使用。当一个应用程序打开一个文件时,该文件就被此应用程序所独占,直到此程序关闭该文件。该文件不能被其它的应用程序所打开,也就是说,一个文件不能同时被多个应用程序所使用。如果禁止并发访问,将在很大程度上降低数据访问的效率,并且浪费大量的数据资源。而在文件系统中解决的办法是为每一个应用程序建立一个文件的副本。但是这样会带来极大的数据冗余,容易造成数据的不一致。

数据的安全性(Security)**不够灵活**

在现代操作系统中,对文件的访问提供了一定程度的安全性。例如,可以设置文件的访问权限,以限制某些用户的使用。但是,在文件系统中对文件中数据内容还不能实现安全访问策略。例如,对于某个用户,只是在某个文件的某一部分数据的访问上对其加以限制。这在文件系统中是不能实现的。

针对文件系统的上述缺点,人们开始研究新的数据管理技术,于是产生了以统一管理和共享数据为特征的数据库系统(Database System)。

3 . 数据库系统阶段

自从 20 世纪 60 年代后期,数据管理进入了数据库系统阶段。这一时期,计算机用于管理的规模更加庞大,数据共享的要求越来越强。随着计算机技术的飞速发展,计算机具有了庞大容量的存储设备和高速的信息处理能力,为保存和加工大量的信息提供了有利的硬件环境。在这样的背景下,出现了数据库技术。与文件系统相比,数据库技术提供了对数据的更高级、更有效的管理。数据库技术的数据管理方式具有以下特点。

(1) 数据一体化、结构化。传统文件系统中的各个文件之间彼此是毫无联系的,要想实现应用程序对它们的交叉访问是十分困难的。而在数据库中,数据是按照某种数据模型组织起来的,不仅文件内部数据彼此相关,而且文件之间在结构上也有机地联系在一起。描述数据时不仅描述数据本身,而且还描述数据之间的联系。例如对于学生和课程,可以定义一个选课联系来描述学生与课程之间的关系。这样通过对数据的结构化,使整个数据库浑然一体,数据库的这种特征称作一体化。

不同于文件系统,数据库是为多个应用目的服务的,数据库的结构不再面向特定的应用,而是面向全组织的复杂的数据结构。也就是说,数据库是面向整个系统或组织的,多数情况下,系统或组织的某个应用只涉及到整个数据库的一部分。因而,数据易于维护和扩充,应用程序也可以有灵活的方式。

(2) 数据共享。数据共享的意义是多种应用、多种语言互相覆盖地共享数据集合。在数据库中,数据不再分属于各个应用程序,而是集中存放在数据库中。对于某个组织而言,除了有安全和保密等限制以外,数据库中的数据被整个组织所共享。也就是说,该组织每个下属部门的应用可以共享这些数据,大大提高了数据的使用价值。

访问数据库的不同的应用程序可能使用不同的语言,因此通常数据库系统配置多种语言接口。

(3) 数据冗余度小。由于数据是一体化的,数据的冗余度大大减小,除了一些必要的副本,例如为了保持联系信息而重复存储的一些数据项,存储数据的冗余度保持在尽可能小的程

度。这既节约了存储空间又可在很大程度上避免数据的不一致性。

(4) 数据与程序的独立性较高。如图 1－3 所示,应用程序必须通过数据库管理系统 DBMS(Database Management System)访问数据库。数据库系统提供影响功能来保证应用程序对数据结构和存取方法有较高的独立性。所说的数据与程序的独立性是指,当数据的逻辑结构或物理结构发生变化时,通过映像部分的改变,保证应用程序不用改变。通常,数据库系统提供两方面的数据与程序的独立性。

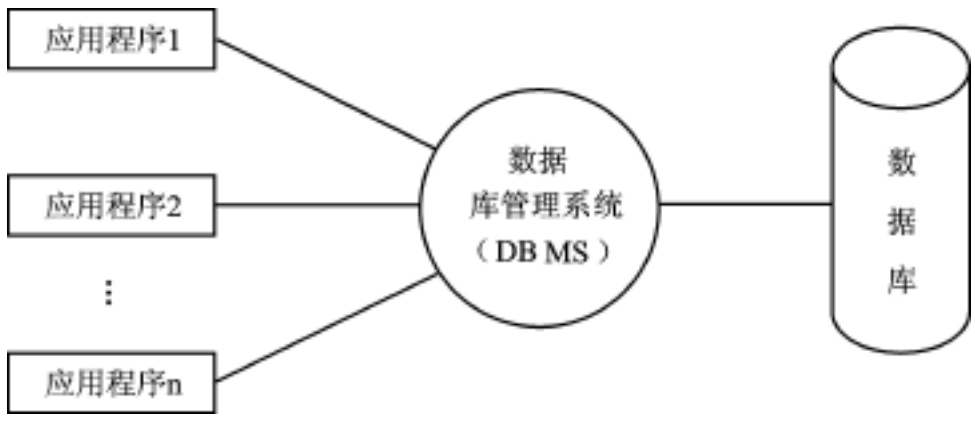

图 1－3　数据库系统阶段数据与程序之间的关系

物理独立性

当数据的存储结构(或物理结构)改变时,通过数据的存储结构与逻辑结构之间的映像或转换功能,使得数据的逻辑结构可以不变,从而应用程序也可以不变。这就是数据与程序的物理独立性。

逻辑独立性

数据库对整个数据的全体有一个整体的逻辑结构,而数据库系统中的某一类应用所使用的数据是全体数据的子集,并且不同的应用对同一数据的使用要求和观点也有不同。因此一个应用往往是根据描述所使用的数据子集的局部逻辑结构而编写的。数据库系统通常提供局部逻辑结构的说明以及整体逻辑结构与某类应用所涉及的局部逻辑结构之间的映像或转换功能。这使得当整体逻辑结构改变时,通过提供的映像或转换功能,保持应用程序所涉及的局部逻辑结构不变,从而应用程序也可以不变。这就是数据与程序的逻辑独立性。

(5) 对数据实行集中统一控制。数据库系统提供统一的数据定义、插入、删除、检索以及更新等操作。另外,由于数据库是系统的共享资源,各种用户可以同时使用数据库,因此说用户对数据的访问是并发的。这要求数据库系统必须提供以下三个方面的控制功能。

数据的安全性(Security)控制

数据的安全性是数据库系统的重要功能之一。为了防止不合法的使用所造成的数据的破坏或泄密,系统必须提供安全保密措施。例如,可以采用用户账号和密码等措施,只有当用户的账号合法,并且密码正确时,才允许用户进入数据库系统。另外,数据库系统还必须提供机制限制用户对某些数据的存取权限,以提供灵活的安全控制功能。

数据的完整(Integrity)性控制

数据的完整性是指数据库中的数据必须正确、有效和相容。数据库提供必要的功能,保证输入或更新的数据符合系统和用户的要求。例如,在职工数据库中,职工的年龄必须大于 0,小于等于 60,职工的性别必须是男或女,职工必须在一个存在的部门工作等。数据库系统必

须提供必要的定义和检查机制,以保证数据符合上述的要求。此外,当计算机的软、硬件发生故障或某个对数据库中数据的操作产生错误时,很容易造成数据的破坏和产生不一致现象。系统必须提供应急和恢复机制,以便把数据库恢复到某个正确状态。

并发性(Currency)控制

当多个用户同时存取、修改数据库中的某个数据时,由于互相干扰可能会得到错误的结果,使得数据库出现不一致现象。因此,数据库系统必须提供并发控制机制,提供策略对多用户的并发操作进行协调和控制。

上面的五个特征是数据库的主要特征。综上所述,数据库是面向特定组织的多种应用而存储起来的一个具有较高的数据与程序的独立性和最小的数据冗余的,相互关联的综合性的共享数据集合。

数据库系统的出现,使得人们对信息的处理方式发生改变。信息处理从围绕加工数据的程序为中心转变到围绕共享数据库来进行。数据库系统对数据集中的管理,有利于应用程序的开发和维护,提高了数据的利用率。目前,绝大多数数据密集型应用所使用的都是数据库系统。

1.1.2　数据库技术的发展

随着计算机科学的不断发展,数据库技术逐渐地壮大起来,大致上经历了三个发展时期。

1.20 世纪 60 年代的萌芽期

第三代电子计算机硬件又进行了一次飞跃,中小规模集成电路已经作为计算机组成的主要器件。此时已经有了磁盘、磁鼓等直接存储设备,并且文件系统已经十分成熟。计算机应用已经扩展到信息处理方面。数据库的概念也开始形成。在 20 世纪 60 年代末已经出现了商品化的数据库系统。在这一时期较有影响的工作如下。

(1) 1964 年美国通用电器公司 Bachman 等人开发成功了世界上第一个数据库管理系统 IDS(Integrated Data Store)。IDS 可以为多个 COBOL 应用程序共享数据库,奠定了网状数据库(Network Database)的基础,并得到了广泛的应用。

(2) 1968 年网状数据库系统 TOTAL 开始出现并应用。

(3) 1969 年 IBM 公司推出了由 Mcgee 等人开发的商品化的层次(Hierarchical)数据库管理系统 IMS(Information Management System)。

2.20 世纪 70 年代的发展期

这一时期出现了许多的商品化的数据库系统。这些系统大多是基于网状和层次的。由于商品化的数据库系统的出现和使用,数据库技术日益深入到了人们生产、生活的各个领域,使得数据库技术成为信息管理的基本技术。在这一阶段,关系数据库的基础理论逐渐充实,并开始出现了较完善的关系数据库系统。这一时期主要的工作如下。

(1) 美国数据系统语言协商会 CODASL(Conference On Data System Language)下属的数据库任务组 DBTG(Database Task Group)对数据库方法进行了深入的研究和讨论,并于 20 世纪 60 年代末,70 年代初提出了一系列的报告,称作 DBTG 报告,澄清了许多概念,建立了若干权威性的观点。DBTG 所提议的方法是基于网状结构的。许多的网状数据库都是基于 DBTG 模型的。CODASL 的工作极大地推动了数据库的发展。

(2) 1970 年 IBM 公司 San Jose 研究所的 E.F.Codd 发表了题为《大型共享数据库数据

的关系模型》的著名论文,开创了数据库的关系方法和关系规范化理论研究,为关系数据库技术奠定了理论基础。鉴于 E . F . Codd 所做的贡献,于 1981 年他获得了 ACM 图灵奖。

(3) 试验性的关系数据库系统开始研制。20 世纪 70 年代中期,IBM 公司 San Jose 研究所研制了 SYSTEM R 关系型数据库管理系统,美国加州大学伯克莱分校实现了 INGRES 关系数据库管理系统。这两个系统大约在 1977 年前后投入运行。这些实验系统在关系数据库管理系统的实施技术和性能方面做了大量的工作。

(4) 1978 年美国标准化组织发表了关于数据库系统结构的最终报告,即 ANSI/ X3/ SPARC 建议,规定了数据库系统的总体结构和特征。

(5) 1979 年美国 ORACLE 推出了第一个商品化的关系数据库管理系统 ORACLE V2 .0 版。

3 . 20 世纪 80 年代的成熟期

这一时期大量的商品化的关系数据库管理系统问世并被广泛的应用。例如,IBM 公司相继推出了 SQL/ DS 和 DB2 等商品化的关系数据库管理系统,INGRES 也被商品化。关系数据库技术已经非常成熟。数据库被应用到人类生活的各个领域。20 世纪 80 年代,关系数据库成为发展的主流,几乎所有新推出的 DBMS 都是关系型的,例如较有影响的商品化的系统 Sybase 和 Informix 等。

随着微机的出现和计算机网络的广泛应用,分布式数据库系统成为研究重点,并走向应用。例如,1986 年出现了分布式数据库管理系统 INGRES/ STAR 和 SQL* STAR,其中 SQL* STAR 是 ORACLE 公司推出的开放型分布数关系数据库系统。应当说,20 世纪 80 年代是关系数据库的全盛年代。

经过了近 40 年的发展,数据库技术仍然是当今十分活跃的研究领域。随着计算机的广泛应用,出现了许多新的应用和新的要求。人们开始发现关系数据库的不足和限制,开展了面向新的应用的数据库技术的研究。数据库技术域网络通信技术、面向对象技术、并行计算技术、多媒体技术、人工智能等技术的相互渗透和相互结合,出现了如 Web 数据库技术、面向对象数据库(Object-Oriented Database)、并行数据库(Parallel Database)、多媒体数据库(Multimedia Database)和知识库(Knowledge Database)等新的数据库技术,并且面向特定的应用领域,人们展开了时态数据库(Temporal Database)、工程数据库(Project Database)、主动数据库(Active Database)、空间数据库(Spatial Database)等技术的研究。可以说数据库技术已经进入了后关系数据库的时代。

1 .2　数据模型

数据的结构化、一体化是数据库与文件系统区别的重要特征。尽管数据的组织仍然以文件为基础,但数据库中的全部数据文件被按照某种数据模型组织在一起,形成了一个整体的数据库。它不仅反映数据本身的内容,而且反映数据之间的联系。数据库中是通过数据模型(Data Model)来抽象现实世界的。数据模型可看作是一种形式化描述数据、数据之间的联系及有关语义约束规则的方法。目前广泛使用的数据模型大致上有两类。一种被称作概念数据模型或简称概念模型。概念模型独立于任何计算机系统,用来描述某个部门或组织信息结构,完全不涉及到信息在计算机系统中如何表示。另一种就是我们所说的数据模型。数据模型实

质上是向用户提供的一组规则。这些规则用以规定计算机系统中数据结构的组织和相应允许进行的操作。

本节我们将介绍数据模型的组成要素及几种常见的数据模型。下面首先介绍数据模型的组成。

1.2.1　数据模型的组成

一般来说,数据模型是用来描述数据的一组概念和定义。它应当描述数据的静态和动态两方面的特性。数据的静态特性包括数据的基本结构和完整性约束;数据的动态特性是定义在数据上的操作。下面来分别介绍数据模型的三个组成部分。

1.基本数据结构

这是数据模型的最基本成分。它规定如何把基本数据项组织成大的数据单位,并表达数据之间的联系。基本的数据结构实质上是组成数据库的对象类型的集合。在这些对象类型中,一类描述与数据的类型、内容和性质有关的对象;另一类描述与数据之间的联系有关的对象。这种基本的数据结构必须具有较强的表示能力,以使其能够表达数据之间的各种复杂的语义相关关系。

2.数据操作

数据操作是指对数据库中的各种对象和实例允许执行的数据操作或推导规则。数据库中主要有检索(查询)和更新(包括插入、删除、修改)两大类操作。任何数据模型都必须定义这些操作的确切含义、操作规则以及实现操作的数据操作语言(DML)等。注意,对于同样的数据结构,所允许的操作不同可以形成不同的数据模型。例如,对于同样的线性存储结构,如果采用先进先出策略的操作,则形成一个队列模型;如果采用后进先出策略的操作,则形成一个栈模型。

3.完整性约束规则

完整性规则是给定的数据模型中的数据及其联系所满足的制约和依存规则,用以明确或隐含地定义正确的数据库状态和状态的变化,以保证数据的正确、有效和相容。

任何数据模型本身隐含着该模型所必须满足的、基本的完整性约束条件,保证数据库中的数据符合该数据模型的要求。此外,数据库系统必须提供完整性约束规则的定义机制,以满足某个应用对所涉及的数据的特定的语义要求。

可以看出,数据模型实际上给出了在计算机系统上描述现实世界信息结构和信息变化的抽象方法。这些方法是通过数据库管理系统(DBMS)来实现的。通常数据库系统是根据所支持的数据模型来命名的,例如支持关系模型的数据库系统被称为关系数据库系统。

当前主要的数据模型分为三类:非关系模型、关系模型和面向对象模型。其中非关系模型包括层次和网状两种数据模型。目前一些新的数据库技术的研究,例如时态(Temporal)、实时(Real Time)、主动(Active)等数据库技术,大多都是基于关系和面向对象模型的。其中,关系数据库以其具有系统的数学理论基础和成熟的技术,取得了巨大的成功。目前,大多数的商品化的系统都是基于关系模型的,因此关系数据库技术是需要重点了解的。在下面的几节中,将介绍这几种常见的数据模型。我们只对这几种数据模型的基本结构加以描述,使读者对这几种模型有一个概要的了解。关系数据库方法将在本书的下一章进行详细的介绍。

1.2.2　层次数据模型

层次模型是较早用于数据库技术的一种数据模型。最有代表性的层次数据库系统当属 IBM 公司的 IMS。顾名思义,所谓的层次模型,就是按照层次结构的形式来组织数据的数据模型。在现实世界中,存在着许多按层次组织起来的事物,例如一所学校包括若干个系,每个系有若干个专业,每个专业有若干名教师等。层次模型就是能够表示这种事物的一种数据模型。

1.层次模型的基本结构

在层次模型中,用记录来描述某个事物或事物间关系。记录是存储数据的单位,由若干个命名字段所组成。记录有型和值之分。所说的记录型是指所描述的事物所具有的共同特征的描述,也就是记录的模式;而记录值是该记录型的一个实例。该实例描述了具体的一个事物特征。例如,如图 1-4 所示,一个系记录型 DEPT 由四个字段组成。它们是:系号、系名、系主任、地点。图 1-5 是该记录型一个实例。

DEPT

系号	系名	系主任	地点

图 1-4　系记录型

D02	计算机系	王　平	信息馆

图 1-5　系记录型的一个实例

层次数据模型中最基本的数据关系是两个记录型之间的双亲-子女联系(Parent-Child Relationship),称作 PCR 型。它代表两个记录型之间的一对多(记为 1∶N)的联系,当然也包括一对一(记为 1∶1)的联系。这种双亲-子女联系表明,子女记录型的多个实例可以与双亲记录型的单个实例相对应;反过来,每个子女记录型的实例只能有一个双亲记录型的实例与之对应。例如,图 1-6 是系记录型与专业记录型间的一个 PCR 型;图 1-7 是该 PCR 型的一个实例。

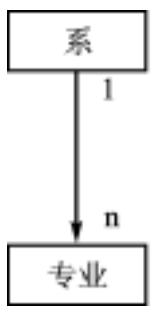

图 1-6　PCR 型

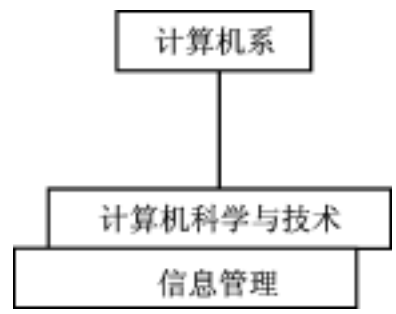

图 1-7　PCR 型的实例

层次数据模型把整个数据库的结构表示成一个有序树的集合。这些有序树的每一个节点是一个记录型,而且任一个节点与它的一个子节点间是一个 PCR 型。层次模型中每棵树必须满足下列条件:

(1) 有且仅有一个节点无双亲,该节点称为根节点;

(2) 除根节点外,其它节点有,且仅有一个双亲。

图 1-8 给出了一个采用层次模型的数据库模式的例子。

这个层次结构体系表达的语义是:在一个学校中,每个系(DEPT)有若干个专业(SP)和教研室(TG),每个教研室有若干名教师,每个专业有若干课程;而每名教师只能属于一个教研室,每门课程只能属于一个专业,每个教研室和专业只能属于一个系。图 1-9 是该数据库模

需要全本请访问 www.ertongbook.com