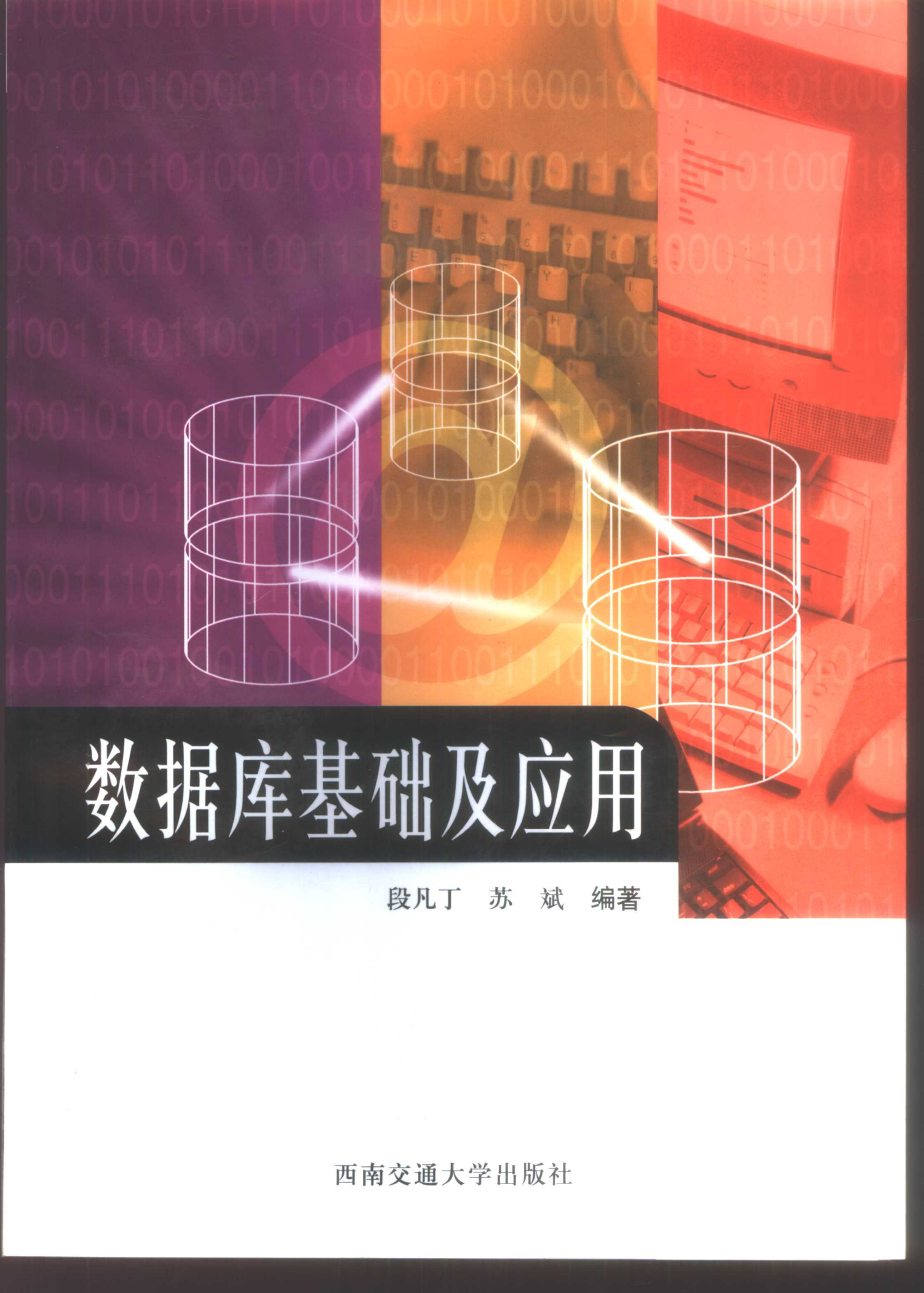


The background of the cover is a collage. On the left, a dark purple vertical band contains binary code (0s and 1s) in a lighter shade. The central and right portions feature a warm-toned image of a computer keyboard and a monitor displaying a bar chart. Overlaid on this are three wireframe cylinders representing databases. Two cylinders on the left are connected by a glowing purple beam, and a third cylinder on the right is connected to them by a glowing yellow beam.

数据库基础及应用

段凡丁 苏 斌 编著

西南交通大学出版社

数据库基础及应用

段凡丁 苏 斌 编著

西南交通大学出版社

· 成 都 ·

内 容 简 介

本教材分为四大部分,第一部分介绍了数据库的基本概念、基本理论知识以及常用数据库管理系统;第二部分介绍了目前流行的结构化查询语言—SQL 语言;第三部分介绍了两种数据库管理系统 Access 和 SQL Server;第四部分介绍了常见的数据库开发工具软件以及开发实例。总之,本教材在简要介绍基础理论的基础上,结合了最新的数据技术,着重于数据库开发的实用性和开发过程,使读者通过学习和上机操作,达到利用数据库技术开发管理信息系统的目的。

图书在版编目 (C I P) 数据

数据库基础及应用 / 段凡丁, 苏斌编著. —成都: 西南交通大学出版社, 2001.9 (2003.1 重印)
ISBN 7-81057-592-9

I. 数... II. ①段... ②苏... III. 数据库系统
IV. TP311.13

中国版本图书馆 CIP 数据核字 (2001) 第 065840 号

数据库基础及应用

段凡丁 苏 斌 编著

*

责任编辑 王 旻

封面设计 肖 勤

西南交通大学出版社出版发行

(成都二环路北一段 111 号 邮政编码: 610031 发行科电话: 87600564)

<http://press.swjtu.edu.cn>

E-mail: cbsxx@swjtu.edu.cn

四川森林印务有限责任公司印刷

*

开本: 787 mm × 1092 mm 1/16 印张: 12.625

字数: 232 千字 印数: 7001—10000 册

2001 年 9 月第 1 版 2003 年 1 月第 3 次印刷

ISBN 7-81057-592-9/TP·263

定价: 17.00 元

前 言

数据库技术的应用到目前为止,大致经历了几个发展阶段。从 dBASE III、Foxbase+、FoxPro 到 Access,再到 SQL Server、Sybase、Oracle 大型网络数据库以及面向对象的开发工具的应用,每个阶段都有它自身的应用特点和应用层。本教材相对于普通数据库方面的教材来说,内容组织比较丰富,力图比较全面地介绍数据库经历的发展以及相应的应用平台软件。本教材分为四大部分,第一部分介绍了数据库的基本概念、数据库的基本理论知识以及常用数据库管理系统,包括了数据库的产生与特点、数据库系统结构、数据库语言、信息结构的三个世界和数据模型等知识点。常用数据库管理系统,包括 Foxbase+软件平台、FoxPro 软件平台、Access 软件平台、SQL Server 软件平台。第二部分介绍了目前流行的结构化查询语言—SQL 语言,这部分内容有别于以往学生用的数据库方面的教材。第三部分介绍了两种数据库管理系统 Access 和 SQL Server 及具体的操作使用技术等,比较细致地介绍了 Access 的安装、使用和 SQL 语言的应用。第四部分介绍了常见的数据库开发工具软件以及开发实例,如 PowerBuilder、Visual Basic、Delphi 等,开发实例以 PowerBuilder 开发工具为主。

本教材内容丰富新颖,综合性较强。在简要介绍基础理论的基础上,结合了最新的数据库技术,着重于数据库开发的实用性和开发过程,使读者通过学习和上机操作,达到利用数据库技术开发管理信息系统的目的。通过例子具体帮助读者掌握相关的知识和技能。本教材主要作为高等院校成人教育(本科或专科)的教材,也可以作为高等院校本科学生的学习参考书以及自学参考书。

本教材第一、三、五、七章由段凡丁编写;第二章由苏斌编写;第六章由魏红宁编写;第四章由李俊峰编写。苏斌高级工程师仔细审阅了本教材并提出了宝贵意见。

由于本教材的改革幅度较大,加之计算机软件发展更是日新月异,限于作者水平,不妥之处在所难免,敬请读者批评指正。

作 者

2001 年 6 月于西南交通大学

1 数据库基本概念

- 1.1 数据库系统的基本概念
 - 1.1.1 数据库的产生与特点
 - 1.1.2 数据库语言
- 1.2 信息结构的三个世界
 - 1.2.1 信息、数据和数据处理
 - 1.2.2 信息结构的三个世界
 - 1.2.3 实体之间的各种联系
- 1.3 数据模型
 - 1.3.1 关系模型 (Relational Model)
 - 1.3.2 层次模型 (Hierarchical Model)
 - 1.3.3 网状模型 (Network Model)
- 1.4 关系系统及有关概念
 - 1.4.1 关系模型的三个完整性
 - 1.4.2 关系代数
 - 1.4.3 关系系统
 - 1.4.4 关系规范化
 - 1.4.5 关系数据语言概述
- 1.5 数据库系统的结构
 - 1.5.1 数据库系统的构成
 - 1.5.2 数据库系统的模式结构

习 题

2 常用数据库管理系统介绍

- 2.1 dBASE III、Foxbase+ 软件平台
 - 2.1.1 dBASE III、Foxbase+ 的特点
 - 2.1.2 数据类型
 - 2.1.3 常量和变量
 - 2.1.4 表达式
 - 2.1.5 命令分类
 - 2.1.6 函 数
- 2.2 FoxPro 软件平台
 - 2.2.1 FoxPro 的用户界面

23	2.2.2 FoxPro 操作方式
24	2.2.3 FoxPro 的帮助系统
24	2.2.4 命令类型和文件类型
25	2.3 Access 软件平台
25	2.3.1 Access 数据库系统的特点
26	2.3.2 Access 2000 简介
32	2.4 SQL Server 软件平台
32	习 题
33	3 SQL 语言基础
33	3.1 SQL 语言概述
33	3.1.1 强大的数据处理功能
33	3.1.2 一体化的特点
33	3.1.3 两种使用方式
34	3.1.4 非过程化的语言
34	3.1.5 语言简洁, 易学易用
34	3.2 SQL 语言介绍
34	3.2.1 数据定义
36	3.2.2 SQL 语言的表达式
37	3.2.3 数据操纵
38	3.2.4 数据查询
44	3.2.5 视图的建立和查询
46	习 题
47	4 Access 2000 数据库管理系统
47	4.1 Access 2000 数据库的一般使用
47	4.1.1 安装 Access 2000
50	4.1.2 启动并打开 Access 数据库
52	4.1.3 Access 2000 数据库窗口
54	4.1.4 保存并退出 Access 2000 数据库
55	4.1.5 获得帮助
57	4.2 Access 2000 数据库的构成
57	4.2.1 表
57	4.2.2 查 询
57	4.2.3 窗 体
58	4.2.4 报 表
58	4.2.5 数据访问页
58	4.2.6 宏
58	4.2.7 模 块

58	4.3 使用查询
60	4.3.1 使用向导创建查询
62	4.3.2 使用视图创建查询
65	4.3.3 建立查询规则
68	4.4 设计 Access 窗体
68	4.4.1 什么是窗体
68	4.4.2 自动创建窗体
70	4.4.3 使用向导创建窗体
74	4.4.4 修改、设计窗体
80	4.5 创建报表
80	4.5.1 自动创建报表
81	4.5.2 使用向导创建报表
85	4.6 创建 Web 页
85	4.6.1 使用向导创建 Web 页
88	4.6.2 设计、修改 Web 页
89	4.6.3 创建 ASP 网页
90	4.7 宏和模块的应用
90	4.7.1 什么是宏
91	4.7.2 创建宏
93	4.7.3 使用宏
94	4.7.4 什么是模块
94	4.7.5 如何编写模块
95	4.7.6 VBA 代码基本语法
97	4.8 表之间的关系
97	4.8.1 什么是表之间的关系
97	4.8.2 怎么建立表的关系
99	习 题
100	5 SQL Server 数据库管理系统
100	5.1 SQL Server 的主要特点
102	5.2 SQL Server 系统的组成
102	5.2.1 SQL Server 的结构
102	5.2.2 SQL Server 的组成
109	5.3 SQL Server 语言简介
115	5.4 数据库对象
119	习 题
120	6 面向对象的开发工具
121	6.1 PowerBuilder 开发工具

121	6.1.1 PowerBuilder 6.0 的主要功能特点
123	6.1.2 PowerBuilder 6.0 的安装
124	6.1.3 PowerBuilder 6.0 可视化集成环境
125	6.1.4 PowerBuilder 6.0 应用程序开发过程简介
130	6.2 Visual Basic 开发工具
130	6.2.1 Visual Basic 6.0 的主要功能特点
131	6.2.2 Visual Basic 6.0 的运行与安装
131	6.2.3 Visual Basic 的可视化集成开发环境
133	6.2.4 Visual Basic 程序开发的一般过程简介
135	6.3 Delphi 开发工具
135	6.3.1 Delphi 5.0 的新功能特点
138	6.3.2 Delphi 5.0 的安装
138	6.3.3 Delphi 5.0 可视化集成编程环境简介
141	6.3.4 Delphi 程序开发的一般步骤
143	6.4 Visual C++ 开发工具
143	6.4.1 Visual C++ 6.0 的功能特点
144	6.4.2 Visual C++ 6.0 的软、硬件环境
144	6.4.3 Visual C++ 6.0 的安装过程
144	6.4.4 Visual C++ 6.0 的集成开发环境
146	6.4.5 Visual C++ 开发应用程序的一般步骤
150	习 题
151	7 一个软件开发实例
151	7.1 软件功能及特点(需求分析)
153	7.2 数据库系统的表及配置文件
174	7.3 程序界面设计
178	7.4 面向对象程序设计实例
191	习 题
193	参考文献

1 数据库基本概念

最初人们研制计算机的目的,是为了帮助人们解决大型复杂的科学计算和工程计算方面的问题,因而就有了计算机(Computer)这一名称的来源。随着计算机及其应用的普及,计算机不仅仅是作为“计算”的机器,而且还可以作为“数据存储和处理”的机器,当然还有其它用途,如作为“图形图像处理”的机器,作为“人工智能”的机器,作为“网络信息传递”的机器等。只是由于传统和习惯的原因,人们一直叫这种具有一定智能的机器为“计算机”或“电脑”。实际上,计算机具有各种各样的用途。到目前为止,计算机的“数据存储和处理”的作用越来越大,由于数据无时不在我们周围存在,那么“数据存储和处理”的应用需求就在我们生活和工作的任何行业 and 任何领域。

1.1 数据库系统的基本概念

1.1.1 数据库的产生与特点

20 世纪 50 年代末出现了文件管理系统,它将数据存储在一个独立的数据文件中,实现了“按文件名访问,按记录来存取”的管理技术,减轻了程序员的数据管理的劳动。但随着社会生产力的发展,信息量急剧增长,人们必须借助一种能大量存储数据、高速处理数据、及时做出结果得到信息的机器,自然只有计算机能满足这种要求。大容量的存储设备——磁盘可为存储大量数据提供物质基础,CPU 具有逻辑判断即处理数据的能力,计算机操作系统软件和文件系统又为数据管理提供了技术基础。但文件系统还存在问题,不能完全适应数据处理的需要。数据库系统(Data Base System)是一种软件,它是在文件系统的基础上发展起来的一种理想的数据库管理技术,它克服了文件系统数据冗余度大、数据 and 应用程序过分的相互依赖、对数据缺乏统一的管理和控制、对数据的正确性、安全性、保密性缺乏有效的统一控制等不足。数据库系统能使数据在统一控制下为尽可能多的应用服务,实现数据的共享,同时使应用程序和数据尽可能相互独立,当数据修改时不要求应用程序作较大的改变,而且当应用程序修改时,也不要求数据结构作较大的改变。

归结起来,数据库系统有以下特点:

1. 数据共享性

数据共享是数据库系统的主要目的,为应用而建立的数据库中的数据不仅可以为同一企

事业或机构内的各个部门的应用所共享，也可为不同单位、地域的用户所共享。

2. 数据独立性

数据库系统中的数据结构和应用程序相互具有独立性，使得一方的修改对另一方的影响不大，便于开发和维护。

3. 数据的结构化

数据库系统的数据具有结构化，即数据库系统中的数据文件是有联系的，并且在整体上服从一定的结构形式。

4. 数据冗余度小

所谓冗余度，即重复性。文件系统中数据是根据应用程序需要而专门建立的，每个应用程序拥有自己的专用数据，因此数据冗余度大。数据库系统能实现应用程序数据共享，不必要的重复将消除，可节省存储空间，减少数据的不一致性。

5. 数据处理效率高

在数据库管理系统 DBMS (Data Base Management System) 的支持下，数据的插入、删除、更新、检索、查找、排序、统计等处理都能通过 DBMS 按一种公用的和可控制的方法进行，对数据处理的程序设计要比用高级语言程序设计要直观、简便得多，因为 DBMS 是面向设计处理的系统软件，本身具有较完善的数据处理的功能。

6. 数据的安全性和数据的完整性好

数据库系统具有用户和用户密码机制，使得只有合法用户才能对允许他访问的那一部分数据进行规定的操作；数据库系统还可以定义数据的完整性检查和并发控制来保证数据的完整性和一致性；数据库系统还采用了故障恢复机制，当随机性的断电、硬件故障等情况发生时，系统能尽快恢复到出错以前的状态。

1.1.2 数据库语言

数据库语言通常包括两大部分：数据库描述语言 (DDL) 和数据库操纵语言 (DML)。数据库描述语言用于定义和描述数据的各种特性，定义数据库的各级模式。根据所定义的不同级别的模式又分为：子模式数据描述语言 (子模式 DDL)；模式描述语言 (模式 DDL)；物理数据描述语言 (物理 DDL) 三种。数据库操纵语言也是用户和数据库系统的主要接口之一。它是由对数据库中的数据进行操作的语句组成，一般完成按给定条件检索数据、向数据库中添数据、删除数据库中的数据以及更新数据库中的数据。

以数据库的标准查询语言 SQL 为例，其 DDL 语言一般设置有 create table/index、alter table、drop table/index 等语句，可分别供用户建立、修改或删除数据库的二维表结构，或者定义或删除数据表的索引。又例如 SQL 中的 DML 语言一般设置有 select ... from ... 语

句, update 语句和 insert 语句等, 用来完成用户对数据库中的数据进行查询、更新、添加等操作。

早期的数据库语言与普通程序设计高级语言不同, 它缺少程序控制语句等部分, 只有嵌入到其它高级语言如 COBOL 语言中使用, 这些高级语言成为其宿主语言, 这样的使用方式显然束缚了数据库语言的功能和应用。因而也必然导致数据库语言一定会发展成为一种具有控制语句和数据操纵语句以及较丰富的、完整的、独立的、库函数的程序设计语言。目前普遍使用的数据库管理系统都具有上述功能。

1.2 信息结构的三个世界

1.2.1 信息、数据和数据处理

在计算机数据处理系统中, 我们经常在说数据、信息以及数据处理这样的名词, 特别是对数据和信息, 往往是不加区别的。其实, 它们的概念是不同的。

数据是一种物理符号序列, 是用来记录事物情况和状态的文字、符号、图像和声音的组合, 它不仅包括表示量值概念的数值数据, 即通常的“1、2、3、4、5、...”等, 也包括表示“陈述”意义的数据, 即非数值数据, 如字符串“student”等。

信息是经过加工并对人类社会的生活、工作、学习、生产及经营活动产生影响的有意义的数据, 即信息也是一种数据, 不过是经过加工的、有用的数据。

信息和数据是两个相互依存、相互联系但又相互区别的概念。信息是数据的有意义的表示, 数据是信息的具体表示形式。零散的、杂乱无章的数据不构成信息, 因为人们无法从中获得信息。

数据处理是对各种类型的数据进行收集、存储、分类、计算、加工、检索和传输的过程。应该说, 经过数据处理的数据, 一般就成为了信息, 通常数据处理也称为信息处理。

1.2.2 信息结构的三个世界

在信息系统中, 信息从客观事物出发将数据存入数据库, 通过处理、加工、决策, 又回到客观事物。信息的这一循环, 经历了三个世界, 即现实世界、信息世界和数据世界。

现实世界是存在于人们头脑之外的客观世界, 这些客观世界的事物反映到人的脑子中, 人的脑子对它有个认识过程, 经过分析(选择、命名、分类等)之后, 进入信息世界, 也就是说信息世界是现实世界在人们头脑中的反映。信息再进一步加工、编码、进入数据世界, 数据世界是信息世界的具体表达。

在信息世界中常用到下列术语:

(1) 实体(Entity)。客观事物在信息世界中称为实体。实体是彼此可以明确识别的对象。实体可以分成“对象”与“属性”两大类, 比如说人、学校、工厂等是对象。而表示“人”这一对象的属性可以有姓名、性别、出生年月、民族、籍贯等诸方面的特征。属性是客观事物中性质的抽象描述。

(2) 实体模型 (Entity Model)。反映实体之间联系的称为实体模型。如教学情况的实体模型反映学生学习情况、教员教学情况, 我们关心的任何事物均可建立一个实体模型。

(3) 属性 (Attribute)。属性表明实体在某一方面的特征。

(4) 属性值。区别属性特征的值。例如学号可有 19990001、19990002、19990003、…等, 都是学号这一属性的值。

(5) 域 (Domain)。属性所取值的范围称为域。如性别的域是男、女。

在数据世界中常用到下列术语:

(1) 记录 (Record)。相应于每一实体的数据。

(2) 字段 (数据项) (Field)。相应于信息世界的属性。

(3) 数据模型 (Data Model)。相应于信息世界中的实体模型, 它是实体模型的数据化, 是数据库设计的关键。

(4) 关键字 (Key)。能惟一标识记录的一个或多个字段称为关键字。例如在学生实体集中, 学号可以惟一地标识每个学生的记录, 所以学号可以成为关键字。一般不选用姓名为关键字, 因为目前重名的学生比较多, 姓名就不能惟一标识每一个学生的记录。在数据库中常常用关键字来进行分类和索引, 这样可方便用户对文件中的记录进行访问。

(5) 文件 (File)。现实世界中的事物并不是孤立的, 我们常把同类事物组合在一起, 称为事物类, 相应地在信息世界中就有一个实体集与之对应, 在数据世界中的对应概念就是文件。记录的类型与各记录值的集合就构成一个数据文件。

数据项、记录、文件都是数据的重要单位, 称为逻辑数据。当把它们存储到计算机内部的存储介质上时, 就称为物理数据。

1.2.3 实体之间的各种联系

现实世界中的事物是彼此关联的, 任何一个实体都不是孤立存在的。因此描述实体的数据也是相互联系的。联系有两种, 一种是实体内部的联系, 反映在数据上是记录内部即字段间的联系; 一种是实体与实体之间的联系, 反映在数据上就是记录之间的联系。实体间的联系虽然比较复杂, 但经过抽象化以后, 可以把它们归纳成一对一联系、一对多联系和多对多联系。

1. 一对一联系 (1:1)

若对于实体集 A 中的每一个实体, 实体集 B 中至多有一个实体与之联系, 反之亦然, 则称实体集 A 与实体集 B 具有一对一联系, 记为 1:1。

例如, 一个学校的学生班级只有一位班长, 同时这位班长只能在这个班级中任职, 这样, 班级实体集和班长实体集是 1:1 的关系。

2. 一对多联系 (1:n)

若对于实体集 A 中的每一个实体, 实体集 B 中有 n 个实体 ($n \geq 0$) 与之联系, 反之, 对于实体集 B 中的每一个实体, 实体集 A 中至多只有一个实体与之联系, 则称实体集 A 与实体集 B 有一对多的联系, 记为 1:n。

例如，一个学校的学生班级有若干学生，而每个学生只在一个班上课，则学生班级和学生之间构成一对多的联系。

3. 多对多联系 ($m:n$)

若对于实体集 A 中的每一个实体，实体集 B 中有 n 个实体 ($n \geq 0$) 与之联系；反过来，对于实体集 B 中的每一个实体，实体集 A 中有 m 个实体 ($m \geq 0$) 与之联系，则称实体集 A 与实体集 B 具有多对多联系，记为 $m:n$ 。

例如，学校开运动会，一个体育项目有多个运动员参加，并且一个运动员可以参加多个体育项目，则体育项目和运动员之间构成多对多的联系。

两个实体型之间的三种联系如图 1-1 所示。

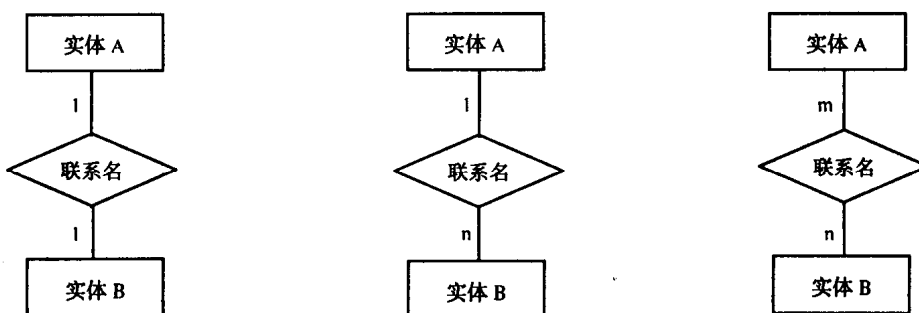


图 1-1 两个实体之间的联系

1.3 数据模型

数据模型的任务是指出数据间的联系，主要是实体间的联系。数据模型的设计方法决定了数据库的设计方法。当前较流行的设计方法有关系方法、层次方法和网络方法三种。对应的数据模型为关系模型、层次模型和网状模型。

1.3.1 关系模型 (Relational Model)

关系模型是以数学理论为基础构造的数据模型，它把每一个实体集合看成是一张二维表，即关系表。例如，表 1-1 便是一个关系，可命名为图书关系。

表 1-1 图书目录

书 号	书 名	作 者	出版社	出版日期
1	操作系统	吴 军	华东出版社	01/03/1992
2	计算机语言	周大同	科学出版社	02/25/1993
3	计算机原理	王德明	华中出版社	07/05/1996
4	数据库基础	魏 超	华北出版社	05/05/1995
5	数据方法	宋 光	华南出版社	06/02/1994

二维表中横向的一行为一个元组，相当于一记录，第一行是各字段“型”的集合，构成一个框架，即为记录的“型”。关系模型的最大优点是简单，一个关系就是一个数据表格，且它可由仅仅一行和一列组成。在关系模型中常用的术语有：

- (1) 关系。是用于定义一个数据表格的，有 m 行和 n 列。
- (2) 域。每个列称为一个域，其中含有一个属性的一切值。
- (3) 元组。每个行称为一个元组。

(4) 关键字。元组由它的关键字来标识。一个元组可能有多个潜在的关键字，其每一个都能惟一地标识该元组。所有这些潜在的关键字称为候选关键字，选中的其中之一来标识该元组的这个关键字称为主关键字。

关系模型的基本组成是关系，我们不仅需要找出表内各个字段的关系，而且经常需要找出表与表之间的关系。如表 1-2 所示。

表 1-2a 学生（关系）

学 号	姓 名	年 龄	性 别
2001001	张 文	20	男
2001002	周 琴	19	女
2001003	李 强	19	男

表 1-2b 课程（关系）

课程号	课程名	学时数
C101	高等数学	200
C102	英 语	200
J101	计算机导论	150

表 1-2c 选课（关系）

学 号	课程号	成 绩
2001001	C101	90
2001001	C102	85
2001002	C101	80
2001002	J101	95
2001003	J101	80

表 1-2 给出的三张表，即学生表、课程表和选课表，它们分别为三个实体集合，学生和课程通过选课这一事实相联系，例如选课中的第二行把学生表中的第一行和课程表中的第二行联系起来了。可见在关系模型中，文件不仅存放实体本身的数据，还能表示实体之间的关系。所以关系模型是通过表格数据而不是通过指针来表示和连接两实体间的联系的，关系模型的结构比较简单，易于接受，同时又有很强的表示能力。

目前大多数数据库管理系统都是关系模型的数据库管理系统。

1.3.2 层次模型（Hierarchical Model）

层次模型的结构是树型结构，树的节点是实体，树的枝是联系。在树型结构中，有一个惟一的节点向上没有联系，这个节点称为根；有若干节点向下没有任何联系，称为叶子；其余节点向上只有一个联系，而向下可以有多个联系。层次模型如图 1-2 所示。

对于层次模型定义的数据库只能依照层次路径存取数据。层次模型的基本联系至少是一

对多的联系，例如，学校对系、班级对学生等联系都属于一对多的联系。

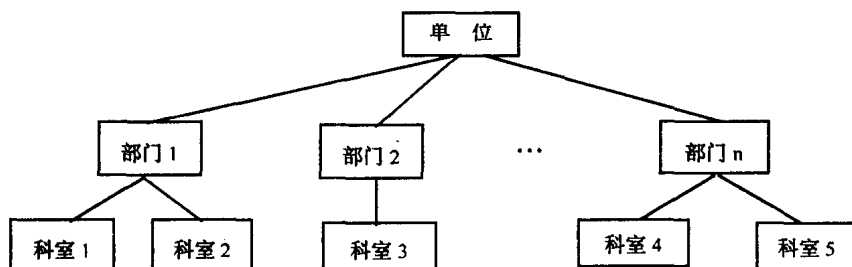


图 1-2 层次模型

IMS (Information Management System) 是 IBM 公司研制的最早的大型数据库系统，它是层次模型的典型代表。

1.3.3 网状模型 (Network Model)

数据的网络模型是以记录类型为节点的网络结构，网络模型的特点是一个子节点可以有两个或多个父节点，可以有一个以上的节点无父节点，在两个节点之间可以有两种或多种联系。图 1-3 表示了一个网状模型的结构。

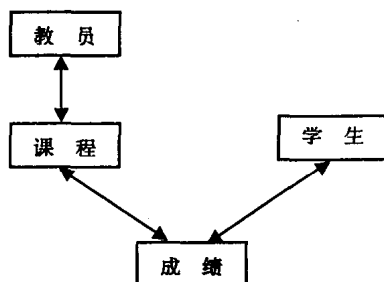


图 1-3 网状模型

DBTG (Data Base Task Group) 是美国 CODASYL (Conference on Data System Language) 下的一个组织，它的 DBTG 系统是网状模型的典型代表。

数据库模型主要描述实体和实体间的联系，关系模型中，统一地用一种结构来描述实体和实体间的联系，这正是关系数据库简单、易学、易用的原因，也是关系数据库检索效率较低的原因。层次模型和网状模型都比关系模型复杂，实际应用也不如关系数据库多。

1.4 关系系统及有关概念

1.4.1 关系模型的三个完整性

1. 实体完整性

它是指基本关系 R 的主属性不能取空值。关系数据库中有各种关系，如基本关系、查询

关系、视图表等。基本表是实际存在的表，它是实际存储数据的逻辑表示；查询表是查询的结果所对应的表；视图表是由基本表导出的表，是虚表，不对应实际存储的数据。实体完整性是针对基本关系的。对于实体完整性有如下说明：

- (1) 一个基本关系通常是现实世界的一个实体集。如学生关系对应学生的集合。
- (2) 现实世界中实体是可区分的，即它们具有某种惟一性标识。
- (3) 关系模型中由主码作惟一性标识。
- (4) 主码不能取空值，因为主码取空值说明存在某个不可标识的实体，而这和第(2)点相矛盾。

2. 参照完整性

若基本关系 R 中含有与另一个基本关系 S 的主码 K_s 相对应的属性组 F (F 称为 R 的外部码)，则对于 R 中的每个元组，其在 F 上的取值为下列两种情况之一：

- (1) 取空值。
- (2) 等于 S 中某个元组的主码值。

例如，设有教工关系 EMP (职工号、职工姓名、所在部门号)，部门关系 DEPT (所在部门号、部门名称)，EMP 的主码为“职工号”，EMP 含有与基本关系 DEPT 的主码“所在部门号”(DNO) 相对应的属性组，因此，DNO 是它的外部码。EMP 中的每个元组在 DNO 上的值有两种可能：一是取空值，这说明该职工尚未分配到某个部门。二是不取空值，则 DNO 的值必须是 DEPT 中的某个元组中的 DNO 值，表示此职工已分配到一个已存在的部门中。换句话说，被参照的关系 DEPT 中一定存在一个元组，它的主码值等于该关系 EMP 中的外部码值，这就是参照完整性。

3. 用户定义完整性

它是指某一具体应用所涉及的数据必须满足该数据库的约束条件，这些约束条件由应用环境决定。如工作时间应晚于出生时间，但早于系统时间。又如月份数应为 1 至 12 之间的整数。年龄不可能为负数等。

实体完整性和参照完整性是关系模型必须满足的完整性约束条件，由关系系统自动支持。

1.4.2 关系代数

关系代数的运算可以分为两类：

1. 传统的集合运算

这种运算是把关系看成是元组的集合，以元组作为集合中的元素来进行计算，其运算是从“水平”的角度，既从行的角度来进行的。

设关系 R 和关系 S 具有相同目的 n ，且 R 中的第 i 个属性和 S 中的第 i 个属性来自同一个域，则说这两个关系是相容的，可以定义三种运算如下：

- (1) 并运算。关系 R 和关系 S 的并运算记为 $R \cup S$ ，结果仍为 n 目关系，由属于 R 或属

于 S 的元组组成。

(2) 差运算。关系 R 和关系 S 的差运算记为 $R-S$ ，结果仍为 n 目关系，由属于 R 但不属于 S 的元组组成。

(3) 交运算。关系 R 和关系 S 的交运算记为 $R \cap S$ ，结果仍为 n 目关系，由既属于 R 又属于 S 的元组组成。

2. 专门的关系运算

这种运算是为数据库的应用而引进的特殊运算，它主要从列的角度即属性的角度来进行的，但有时也会对行有影响。

(1) 选择运算。从关系 R 中选择满足给定条件的诸元组。

(2) 投影运算。关系 R 上的投影是从 R 中选择若干属性列组成新的关系。

(3) 连接运算。从两个关系的笛卡尔积中选出属性间满足一定条件的元组。

1.4.3 关系系统

关系系统的定义应包含两方面内容：一是关系数据库。即从用户观点看，数据库是由表构成的，并且只有表这种结构。二是支持选择、投影和自然连接运算。对这些运算不必要求定义任何物理存取路径。

一个系统仅支持关系数据库而没有选择、投影和连接运算功能的，不能称为关系系统。因为不支持这三种关系运算的系统，用户使用仍不方便，不能实现提高用户生产率这一关系系统的主要目标。

一个系统虽然支持这三种关系运算，但要求定义物理存取路径，如要求用户建立索引才能按索引字段检索记录，也不能成为关系系统。这是因为依赖物理存取路径来实现关系运算就降低了或丧失了数据的物理独立性。不依赖物理存取路径来实现关系运算就要求关系系统自动地选择存取路径。为此，系统要进行查询优化，以获得较好的性能，这正是关系系统实施的关键技术。

上面给出了关系系统的定义，仅仅是关系系统的最小要求。当前许多产品如 SQL Server、Sybase、Oracle、Informix 等都是关系系统，它们都不同程度地超过了定义的要求。根据 E. F. Codd 的思想，可以把关系系统分为四类：

(1) 表式系统。这类系统仅支持关系（即表）数据结构，不支持集合级的操作。表式系统不能算关系系统。

(2) (最小) 关系系统。这类关系系统，仅支持关系数据结构选择、投影和自然连接这几种操作。许多微机关系系统就属于这一类。

(3) 关系上完备的系统。这类系统支持关系数据结构和所有的关系代数操作（功能上与关系代数等价）。SQL Server、Sybase、Oracle、Informix 等许多系统都属于这一类。

(4) 全关系系统。全关系系统应该完全支持关系模型的所有特征。E. F. Codd 具体地给出了全关系型的关系系统应遵循的十二条基本原则，从实际意义上看，这十二条准则可以作为评价或购买关系型数据库产品的标准。但是，到目前为止尚没有一个系统真正属于全关系系统。