

吴炜煜 编著

工程数据 管理系统



清华大学出版社

(京)新登字 158 号

内 容 简 介

本书主要是为非计算机专业而写的,全书内容分为七章,在扼要介绍数据库技术基本原理后,对关系数据库系统特征、基本命令、程序设计和 SQL 语言系统地进行讲解,重点阐述工程数据库原理和专业语义表达、面向对象数据库系统主要技术关键问题、图象图形数据库设计理论和方法。旨在帮助读者学习应用数据库技术新成果,掌握工程数据管理和设计复杂业务的数据库管理系统应用技术。

本书简明扼要、深入浅出、原理清晰、方法具体。全书体系合理、技术内容更新及时。

凡具有计算机基础和学过一门程序设计语言者都可学习本书基本内容,这是一本学习新一代数据库技术基础及应用的综合教材,适合工科大学本科生、研究生和高级管理人员,也可供培训和自学参考。

JS153/07

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: 工程数据库管理系统

作 者: 吴炜煜

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

印刷者: 通县大中印刷厂

发行者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张: 21 字数: 550 千字

版 次: 1996 年 9 月第 1 版 1996 年 9 月第 1 次印刷

书 号: ISBN 7-302-02177-5/TP·1039

印 数: 0001—5000

定 价: 22.00 元

前 言

数据库技术是在计算机的主要应用领域从科学计算发展到社会事务数据处理时应运而生的。现在数据库理论及其商业化推广应用,都已取得了令人瞩目的成就,数据库技术已成为计算机科学技术的主要领域之一。

数据库技术发展将近 30 年,大体经历三次跃升。在 60 年代计算机操作系统问世后,人们致力于数据文件组织、管理、检索和更新等方面研究,为建立统一的数据组织的操作环境,产生了数据模型概念。以层次模型和网状模型为典型代表,奠定了第一代数据库技术基础;随后出现的关系数据模型,具有较高的数据独立性和较严密的数学基础,关系理论系统化和规范化的设计模式,是数据库技术走向成熟的标志,第二代数据库技术以关系数据库为代表,而且商业化生产的关系型数据库系统软件全面覆盖各种档次机型、各方面用户;随着计算机科学技术的进一步发展,计算机应用领域的不断扩大,数据库技术与通信网络、人工智能、图象图形技术、多媒体技术、工程应用技术等相结合,形成新的技术目标,出现第三次跃升,如知识库、面向对象数据库、图象图形数据库、多媒体数据库等,已成为人们研究的热点课题,社会在呼唤第三代数据库的到来。

数据库技术中的层次数据模型、网状数据模型、关系数据模型被称为传统数据库模型,它们都是以规范关系为基础,在处理结构化数据(如文字、数值信息媒体)上业已获得巨大成功。然而很多应用领域中,如 CAD/CAM, CIMS, CASE, OIS 中;存在大量非结构化数据,如图象、声音、甚至动态视频数据,存在大量复杂对象的模拟、操作和推理,数据对象本身内涵和其间联系规则多样性,传统的数据模型不能满足人们描述现实世界的需要,有待于数据库理论的新发展。

数据库技术的研究主要有以下几个方面:

1. 高级数据模型,能够表达丰富语义并具有高效处理能力,构成复杂对象模拟、操作、推理体系。如现在的语义模型、演绎模型、面向对象模型,超媒体模型等,都是很有意义的尝试。
2. 多媒体信息的存储管理和存取方式,能够有效地存储和管理对象数目多、数据容量大的超媒体信息,并且有高效率的数据操纵机制。实时存取是目前追求的目标,这是数据库技术对多媒体和虚拟现实技术应用的支持要求,国际上许多专家正在为此进行多方面的有效探索。
3. 新型用户界面技术,由于计算机中增加声音、图像接口,数据库可以提供更加友好的人性化用户界面,如窗口编辑,文语转换,头盔、手套和传感器等。
4. 协同环境与并发控制技术,在实现计算机协同工作 CSCW (Computer Supported Cooperative Work)和群件环境下,在分布式信息系统中,对相应的数据库系统、特别是事务并发控制技术提出了新的要求。

面临计算机应用领域不断拓展,特别是工程领域复杂应用需求的挑战,数据库技术在理论和应用研究上,有一系列待突破的重大课题。我们都记得,70 年代是关系数据库的研究得到突飞猛进的发展时期,并且它迅速走向成熟,E. F. Codd 博士对此作出了杰出贡献,曾获计算机科学最高奖——ACM 图灵奖。现在,同样是一个需要英雄并会造就英雄的年代,在全球信息化社会建设和信息高速公路计划实施中,将推动和加速数据库新技术的发展。

在目前大学教育中,为培养面向 21 世纪人才,作为计算机基础教育,对非计算机专业的学生,也普遍开设了数据库技术及应用课程,求学者趋之若鹜,实乃社会信息化对人才知识结构的需求驱动。而现在出版的教材,大致有两类:一是系统讲解数据库系统原理;一是专门介绍如何使用某种数据库系统软件的操作方法。前者适于计算机软件专业教学,后者适于一般数据库操作员培训,但对于高级工程技术人员、社会和经济管理人员、非计算机专业大学生和研究生,他们既要求学习一定的数据库技术理论,掌握该技术进步的新成果,又要具有较高的应用设计能力和使用水平,具有解决复杂的专业技术与数据库技术的跨学科交叉问题的能力。因此,他们感到现行的一些教材或教学内容安排,不尽人意,这是不无道理的。

鉴于上述情况,我们根据在清华大学非计算机专业的数据库技术及工程应用的多年教学实践,在原讲义基础上进行整理和增补,编写了这本教材。本书分为七章:第一章扼要阐述传统数据库基本模型、理论及设计方法;第二章着重分析工程数据库基本概念和原理,特别是专业语义的描述问题,结合实例进行介绍;第三章则综合介绍关系数据库软件特点及其基本命令格式和功能,第四章进一步讲述批命令程序设计方法和技术,这两章选材一方面考虑了较大应用面的上机实践,另一方面考虑到 dBASE、dBASE for Windows 和 FoxBASE⁺、Fox Pro for Windows 等 xBASE 系统基本命令的相似性,旨在使读者能在实际操作和编程技能上达到举一反三之功效。对于关系数据库的通用 SQL 标准及其典范 ORACLE 嵌入 SQL,在第五章进行较为详细的介绍,这样通过三、四、五章内容,使读者既能较全面地了解关系数据库理论的软件实现,又具有对广泛使用的 RDBMS 软件的应用技能;第六章全面介绍面向对象数据库的基本概念、数据模型、数据库设计和主要技术关键问题;第七章介绍图象图形数据库系统设计理论、方法和发展趋势;这两章同时还提供第三代数据库技术发展方向的有关内容,帮助读者学习和应用数据库技术的新成果,了解和研究数据库技术发展趋势。关于配合本书的应用程序实例、各章习题及习题解答、上机实验指导书,将另行编辑出版。

诚然,数据库技术及应用内容丰富,要求在有限学时内全面覆盖其理论、软件及应用技术,确实有一定的困难。在使用本书时,对非计算机专业本科生而授课学时为 36 时(上机另用 20 课时),讲授内容以前四章为主,其它章节为选讲或指导阅读;而研究生则要求以第二、六、七章为主修内容。教学方法上把精讲要点、指导自学与系列上机实践相结合。当课程结束时,学生面对自己所编制成的一套数据库管理程序,自觉“颇有成就感”,可谓基本达到教学目标。

在本教材编写中,尽可能反映数据库新技术的发展、工程应用的深入,但限于学识之不足,错误和不妥之处,敬请不吝赐教。

吴炜煜

1995 年 4 月于清华园

目 录

前言	(VI)
第一章 数据库技术基础	(1)
1.1 数据管理的基本概念	(1)
1.1.1 数据、信息与数据处理	(1)
1.1.2 信息的三个世界	(2)
1.1.3 信息数字化编码原则	(3)
1.2 数据库系统特点和构成	(4)
1.2.1 数据库系统特点	(4)
1.2.2 数据库系统模式结构	(5)
1.2.3 三种流行数据模型及数据库	(6)
1.2.4 数据库管理系统	(9)
1.3 关系数据库理论与规范	(10)
1.3.1 关系代数方法	(10)
1.3.2 数据依赖关系	(13)
1.3.3 规范化模式	(15)
1.4 数据库设计与实现	(17)
1.4.1 数据库设计工作步骤	(17)
1.4.2 数据库的实现与维护	(21)
第二章 工程数据库系统设计	(23)
2.1 工程数据库技术的发展	(23)
2.2 工程数据库系统分析	(24)
2.2.1 工程数据特点	(24)
2.2.2 工程数据库管理系统的功能需求	(25)
2.3 EDBMS 的数据模型	(27)
2.4 EDBMS 的专业信息处理设计	(29)
2.4.1 专业信息与系统功能结构	(29)
2.4.2 专业信息的综合与分解	(31)
2.4.3 模糊信息与专业语义	(34)
2.4.4 专业信息交换的标准化	(36)
2.4.5 工程数据与知识库	(37)
第三章 xBASE 系统基本命令	(41)
3.1 dBASE 数据库系统概述	(41)
3.1.1 dBASE 系统产品的发展	(41)
3.1.2 dBASE 系统功能特点	(41)
3.1.3 dBASE 技术规格	(42)

3.2	dBASE 系统基础	(43)
3.2.1	系统数据表示方法	(43)
3.2.2	运算符与表达式	(44)
3.2.3	命令构成与使用规则	(45)
3.2.4	系统设置与运行	(46)
3.2.5	dBASE 文件类型	(49)
3.2.6	dBASE 函数集	(50)
3.2.7	SET 命令集	(54)
3.3	数据库文件的建立与显示	(60)
3.3.1	建立库结构	(60)
3.3.2	库数据录入	(61)
3.3.3	数据库文件的开启和关闭	(62)
3.3.4	数据库的显示	(62)
3.3.5	数据库结构修改	(64)
3.4	数据库记录的修改与更新	(64)
3.4.1	数据库记录定位	(64)
3.4.2	全屏幕编辑方式	(65)
3.4.3	数据库记录修改	(66)
3.4.4	数据库记录删除	(68)
3.5	数据库的检索与统计	(70)
3.5.1	数据库记录排序	(70)
3.5.2	数据库查询检索	(71)
3.5.3	字段和记录筛选	(73)
3.5.4	数据统计命令	(74)
3.6	多重数据库操作	(76)
3.6.1	访问其它工作区中的数据库	(76)
3.6.2	建立与保存数据库文件间的关联	(76)
3.6.3	数据库文件间的横向连接	(77)
3.6.4	数据库文件间的数据交换	(78)
3.6.5	数据库文件间的更新	(79)
* 3.7	关系数据库系统软件简介	(80)
3.7.1	FoxPro 数据库系统	(80)
3.7.2	Oracle 数据库系统	(83)
3.7.3	dBASE5.0 for Windows 新型数据库系统	(90)
第四章	xBASE 命令语言程序设计	(92)
4.1	命令程序设计基础	(92)
4.1.1	程序的建立和运行	(92)
4.1.2	程序设计的步骤	(93)
4.1.3	编程辅助命令	(95)
4.2	程序控制结构与过程调用	(98)

4.2.1	选择结构程序设计	(98)
4.2.2	循环结构程序设计	(99)
4.2.3	过程调用结构程序设计	(102)
4.3	屏幕画面格式设计	(105)
4.3.1	屏幕设计命令	(105)
4.3.2	格式文件的建立与使用	(113)
4.3.3	窗口的定义和应用	(113)
4.3.4	屏幕菜单设计和使用	(115)
4.4	程序的中间数据处理设计	(119)
4.4.1	内存变量操作命令	(119)
4.4.2	变量数组操作方法	(121)
4.4.3	数据库与变量数组的数据转换	(122)
4.4.4	中间数据处理与转换的目标和技巧	(124)
4.5	外部通信程序设计	(126)
4.5.1	与外部文件交换数据	(126)
4.5.2	与高级语言的程序通信	(130)
4.5.3	与外部设备通信程序设计	(132)
4.5.4	直接存取数据库程序设计	(135)
4.6	dBASE 中的 SQL 应用	(138)
4.6.1	SQL 与 dBASE 互用方式	(138)
4.6.2	SQL 的数据库、表、窗口	(142)
4.6.3	SQL 命令格式与功能	(144)
第五章	SQL 语言标准与应用软件	(148)
5.1	SQL 语言标准的发展	(148)
5.2	SQL 语言特点和基本概念	(149)
5.3	SQL 语言的语法结构及公用元素	(153)
5.3.1	SQL 函数	(153)
5.3.2	SQL 语言的谓词	(155)
5.3.3	SQL 表达式	(157)
5.4	数据库模式定义与操纵语言	(158)
5.4.1	数据库模式定义	(158)
5.4.2	数据库模式操纵	(161)
5.5	数据库的数据操作语言	(162)
5.5.1	数据查询语句	(163)
5.5.2	数据维护语句	(165)
5.5.3	事务管理语句	(166)
5.6	SQL 嵌入适配语句	(167)
5.6.1	嵌入 SQL 语句格式	(167)
5.6.2	SQL 模块调用	(169)
5.6.3	动态 SQL 语句	(169)

5.7	ORACLE 中的 SQL 应用	(173)
5.7.1	ORACLE 软件与 SQL 应用环境	(173)
5.7.2	SQL * PLUS 数据库定义	(176)
5.7.3	用户数据实体的修改与取消	(182)
5.7.4	SQL * PLUS 数据操纵语言(DML)	(186)
5.7.5	ORACLE 支持的运算符、表达式和函数	(188)
5.7.6	SQL 数据查询语言(QUERY)	(194)
5.7.7	SQL 高级查询语法格式	(199)
5.7.8	PRO * SQL 预编译和调用界面	(204)
第六章	面向对象数据库系统	(210)
6.1	面向对象技术	(210)
6.1.1	面向对象程序设计方法	(210)
6.1.2	面向对象的概念	(211)
6.1.3	面向对象系统特性	(213)
6.2	面向对象数据模型	(215)
6.2.1	基本数据结构	(215)
6.2.2	OPAL 的数据定义	(217)
6.2.3	OPAL 的数据操纵	(221)
6.2.4	VODAK 的数据模型及应用实例	(226)
6.3	面向对象数据库设计	(228)
6.3.1	OODBS 设计的三级方式	(228)
6.3.2	OODBS 的用户视图	(230)
6.3.3	OODBMS 实现方案设计	(234)
6.4	若干技术关键问题	(236)
6.4.1	复杂对象结构	(236)
6.4.2	对象标识机理	(237)
6.4.3	对象定义封装	(239)
6.4.4	存储结构与存取方法	(241)
6.4.5	模式进化和版本管理	(243)
6.4.6	消息传递管理	(247)
* 6.5	标准化研究与发展趋势	(248)
6.5.1	面向对象数据库系统的必备特性	(249)
6.5.2	面向对象数据库系统的可选特性	(256)
6.5.3	面向对象数据库系统的开放部分	(257)
6.5.4	第三代数据库系统的开发原则	(258)
6.5.5	第三代数据库系统的遵循命题	(259)
6.5.6	面向对象语言的谱系	(269)
第七章	图象图形数据库	(278)
7.1	图象图形数据库概述	(278)
7.1.1	图象信息系统	(278)

7.1.2	图象数据库的应用	(280)
7.1.3	主要研究内容和方法	(281)
7.2	图象图形数据库的系统构成	(282)
7.3	图象图形数据库的设计方法	(285)
7.3.1	超图数据模型	(285)
7.3.2	NF ² 模型	(286)
7.3.3	集成化设计方法	(289)
7.3.4	图标索引设计方法	(291)
7.3.5	面向对象设计方法	(292)
7.4	图象数据库的知识表示	(293)
7.4.1	图象的二维字符串表示法	(294)
7.4.2	图象的语义层次框架描述法	(295)
7.4.3	图象特征值及模糊值表达法	(296)
7.4.4	智能型图象数据库系统构造	(301)
7.5	图象数据库查询语言设计	(303)
7.6	图形数据库的参数化设计	(307)
7.6.1	参数化设计原理与应用	(307)
7.6.2	四种参数化设计方法	(309)
7.6.3	参数化编码法的 CAD 应用	(313)
7.7	先进图象图形数据库技术的发展	(319)
7.7.1	硬件环境的进步	(319)
7.7.2	相关技术的支持	(320)
7.7.3	图象图形数据库发展趋势	(321)
7.8	结语	(324)
参考文献		(325)

注:目录中凡带有*号标记者为参考阅读部分。

第一章 数据库技术基础

数据库(Data Base)技术是计算机软件的一个重要分支,是 20 世纪 60 年代后迅速发展起来的一门新兴技术。数据库技术的应用面宽,涉及领域广,在诸多技术中的地位和作用已受到普遍的关注。对数据库的各种模型和理论研究,数据库在各应用领域的技术特征和处理方法,在全球范围内正广泛开展研究。人们清楚,数据库技术对人类社会发展成为信息化社会具有重大影响,对计算机科学技术的发展具有重要意义,因此,数据库技术是培养现代化人才的必修课。

在这一章中,我们概括阐述数据库技术的基本概念和原理,基本结构和数据组织技术特点,以帮助读者尽快掌握数据库技术的基础知识,为进一步学习工程应用和其它应用技术提供条件。

1.1 数据管理的基本概念

人类社会的历史与发展,离不开对信息数据的收集、保存、利用和处理,电子计算机的出现,使得利用和处理信息进入更高级的阶段,现在对信息和数据的处理、利用已进入自动化、网络化和国际化的阶段。

1.1.1 数据、信息与数据处理

所谓**数据**,是指可被记录、识别的物理符号序列。例如:语言、文字、数字、图象、声音等均是数据。数据是用来记录和反映事物状况的客观描述。

信息(information)向人们提供关于现实世界的新的事实知识。通过各种方式可以被传送,可以用能被人感受的声音、图形、文字所表征,并且与某特定事实、主题或事件相联系的消息、情报或知识,均可被称为信息。即是说,信息是经过加工并对人类社会实践和生产实践活动产生决策影响的数据。

数据与信息在概念上是有区别的,不是所有数据都能成为信息,只有抽取加工之后,具有新的事实知识的数据才成为信息。数据经过加工处理之后成为信息,仍然以数据形式表现,此时数据是信息的载体,成为人们能动地认识和改造现实世界的媒体。

数据处理是对各种形式的数据进行收集、存储、分类、计算、加工、检索和传输的一系列活动的总和。通常数据处理也称为**信息处理**。

数据管理是数据处理的中心环节,即收集数据、保存数据和提供正确的数据。主要围绕提高数据独立性、降低数据的冗余度、提高数据共享性、提高数据的安全性和完整性等方面进行改进,让使用者能有效地管理和使用数据资源。

数据结构(data structure)是数据之间关联和组织形式,或简言之,是带有结构的数据元素的集合。而**数据元素**(data element)是数据的基本单位,即数据集合中的单体(客体)。

数据结构分为物理结构和逻辑结构。数据在物理存储设备中的存放形式称为**物理结构**,亦称**存储结构**,是数据结构在计算机中的映象(map)。数据之间的逻辑关系称为**逻辑结构**,是用

户程序操作数据的根据。一种数据的逻辑结构,可以用不同的物理结构来实现。数据库管理软件为应用程序和数据库数据之间提供了数据的物理独立性和逻辑独立性。

1.1.2 信息的三个世界

在信息系统中,信息从客观事物出发,将数据存入数据库,通过控制、决策机构,又回到客观事物,这一循环过程涉及三个不同范畴:现实世界,信息世界,数据世界。

现实世界是存在于客观事物及其相互联系之中的,是信息之源,是数据库管理的出发点,也是使用数据库的归宿。

信息世界,也称**观念世界**,是现实世界在人头脑中的反映。客观事物在观念世界中称为实体(Entity),反映实体之间的联系称为实体模型。

数据世界,也称**机器世界**,是观念世界中信息的数据化,现实世界中的事物及其联系在计算机中用数据模型描述。

三个范畴之间的区别和联系,是唯物主义和辩证法的生动关系体现,实体模型和数据模型是对客观事物及其联系的两级抽象描述,模型的正确与否,是依靠对客观事物的认识了解,并且受客观事物的检验,建立模型过程是一个实事求是的过程。

在信息世界中有一些常用术语,介绍如下:

- **实体(Entity)**:客观事物中具有可区分性质的信息对象,在信息世界中称为实体。
- **实体模型(Entity Model)**:反映实体之间的联系的客观描述。
- **属性(Attribute)**:对实体特征的抽象描述。每一属性表明实体某方面特征,实体有各种各样的属性。
- **属性值**:属性特征的具体取值。
- **域(Domain)**:属性的取值范围,如:“性别”的域是男或女。
- **实体型(Entity Type)**:用实体名及其属性名集合来抽象和描述同类实体。如学号、姓名、年龄、性别、班级、成绩……组成学生情况这一实体型。
- **实体集(Entity Set)**:同型实体的集合,如一组职工、一班学生、一批零件,他们的特性相同,即具有相同的实体型组成的集合。

在数据(机器)世界中常用术语介绍如下:

- **记录(Record)**:描述实体对象的数据,对应于每一实体的一个数据序列就是一个记录。具有相同结构框架的一类记录称为记录类型。
- **数据项(字段)(Field)**:描述实体属性的数据称为数据项。由于一个实体有若干属性,故有若干个数据项,即一个记录由若干个字段有序组成。
- **数据模型(Data Model)**:是对客观事物及其联系的数据描述,即实体模型的数据化,用于提供信息表示和操作手段的形式构架,是数据库设计的核心问题。
- **关键字(Key)**:能唯一标识记录的一个或多个数据项称为关键字。如学生实体集合中,学号具有唯一性,可标识每个学生记录,所以可用学号作为关键字。假如学生没有重名,那么数据项“姓名”也可以标识每条记录,也是关键字。通常我们选定其中一个称为**主关键字(Primary Key)**,其它的关键字称为**候选关键字(Candidate Key)**。
- **次关键字(Secondary Key)**:不能完全做到唯一标识记录的数据项,用于作次关键字。

为使数据模型能清晰、准确反映客观事物的实体模型,应注意如下几方面工作:

1. 将数据模型命名,使不同模型可以区别,便于标识。

• 2 •

2. 给每个数据项命名,以说明和区分每个记录所包含的数据项,并确定标识不同记录的主关键字。

3. 不同记录类型是数据项型的不同有序组,应指明不同记录型之间的关系,必要时给这种关系命名。

4. 必要时指出各数据项的数据特征,即其允许使用的数据类型、长度、值域。

1.1.3 信息数字化编码原则

计算机处理信息,必要条件是用事先约定的计算机能接受的形式表示信息。常用的形式是将信息设计成代码,为提高计算机对这些信息的存储、管理和加工处理的效率创造条件。

信息编码必须遵从下述原则:

1. 编码的唯一性原则。一个代码只能表示一个事物或一个概念。无论编码对象有何种名称或描述,作为一个编码对象则只能有一个唯一赋予它的代码。代码设计的唯一性原则贯穿于整个代码系统的始终。

2. 编码与数据结构关联性。要能反映数据类别和特性,要在充分分析编码数据集的结构和分类的基础上,对其子集分别给予适当编码,以利于识别各子集数据,把握总体信息。

3. 编码的标准化原则。这是要求尽可能采用已颁布的国际国内有关标准化统一编码形式。如地名,标准化编码已制订,国际上美国用 502,而香港用 110,朝鲜用 104 等;国内标准编码:上海市,31,江苏省,32,等;又如货币,用三个字母,前两个字母为联合国制定的国家码,后一字母为货币的第一个字母,人民币用 CNY,美元用 USD,等,这些已经确定了信息代码标准的,在用户自行设计系统中,如果有相应编码对象,则不能另搞一套,而应采用标准。如果信息对象尚无标准编码,则根据实际情况处理,但也应考虑靠拢标准编码形式。

4. 编码的稳定性原则。是指所制订信息代码方案在该系统生命期内具有相对稳定性。例如银行存单号,如果不以稳定性代码,而是变换使用,将导致难以预计的影响。

5. 编码要考虑可扩展性,防止因数据量扩充而重构编码结构。因此编码分析期间既要准确预测编码对象总量,还要留有余地,为不可预见性编码对象保留适当数量的备用码。

6. 编码的简短性原则。在代码长度设计满足应用要求的前提下,编码要尽可能简短,以便节省存储空间和数据处理、传送时间。编码简短原则是指缩小代码系统冗余度,而不是牺牲代码的可扩充性。

7. 编码的可识别性原则,即在正确反映编码对象内容的条件下尽量采用易于识别的代码,如字母码、数字码、数字-字母混合码等,用音、形、义、逻辑性,层次顺序等有助于记忆的代码设计,提高编码的可识别性。

8. 编码的一致性原则,是指在一个系统内各相关数据文件和程序文件中,对同一编码对象使用相同代码格式,并且前后一致,始终如一地坚持确定的代码格式。同时,代码含义应保持一致性。具体做法是编制系统码表,作为共同遵守的规范标准。

信息数字化编码过程是信息收集、加工的过程,是进行数据分析、进一步确立数据模型的过程,也是将管理对象信息规范化、进行数据定义的工作过程。这一过程将使系统设计目标具体化。

信息编码方法有多种,常用的方法是:顺序码(由一连串数字组成),区段码(由若干码段组成),还有分支法编码,混合编码法(字母、数字混合组成),字母缩写码等。有些信息并不能都用短码表示,而是采用较为自然的离散数据,如声音、图象信息,则采用相应的编码算法理论和技

术。对于专业技术信息,在编码方法上必须同工程技术人员的专业术语和符号以及其习惯表示方法尽可能接近,这有利增加编码的可识别性,给使用者带来方便。

1.2 数据库系统特点和构成

数据库系统(Data Base System,简称 DBS)是数据管理技术的一次飞跃。50 年代出现的文件管理系统把数据组织在一个个独立的数据文件中,按文件名来访问,按记录进行存取,曾是可贵的技术进步,但是大信息量数据管理要求数据存储低冗余,要求数据访问效率高,要求实现数据共享,要求反映数据之间复杂联系结构,文件系统数据管理方式不能适应,数据库系统应运而生。

1.2.1 数据库系统特点

数据库系统的管理对象是大量数据集合,数据能长久地可靠地保留,提供若干用户共享。与文件系统相比,数据库系统有如下显著特点:

1. 数据的结构化

在文件系统中,在整体上不存在结构化,不能使用公用的可控方式进行数据存取。而数据库系统不仅考虑数据项之间联系,而且考虑记录型之间联系,不仅考虑一种应用的数据结构,而且考虑相关应用的整体要求的数据结构。

2. 数据冗余度小

数据库的数据描述是面向整个系统的结构化组织,因而大大减少了数据冗余度,既节省存储空间,降低存取时间,又避免数据之间的不相容和不一致性。同时,为适应用户需求变化要求,也易于扩充和修改。

3. 具有较高的数据和程序的独立性

数据库系统把数据的定义和描述从应用程序中分离出去。数据库系统提供了两方面映象功能,一是数据的存储结构与逻辑结构之间的映象或转换功能,二是数据的总体逻辑结构与某类应用所涉及的局部逻辑结构之间的映象或转换功能,这是数据和程序的独立性基础条件。数据的存取管理交由数据库管理系统程序负责,用户不必考虑存取细节,从而简化了应用程序的编制。

4. 统一的数据控制功能

数据库是各用户共享的资源,共享必然伴随着公用可控方式和并发操作处理要求。数据库系统提供安全性(Security)控制,防止不合法的使用所造成的数据泄密和破坏;提供数据的完整性(Integrity)控制,保证数据的正确性、有效性和相容性;提供并发(Concurrency)控制,使多用户并发存取、修改数据库的操作能够正确地加以控制和协调。

从上述数据库系统特点可以清楚说明,信息管理系统的研制工作中心是围绕共享的数据库来进行。关于什么是数据库,J·Martin 在其“Computer Data-Base Organization”中曾定义如下:数据库是存储在一起的相关数据的集合,这些数据无有害的或不必要的冗余,为多种应用服务;数据的存储独立于使用它的程序;对数据库插入新数据库,修改和检索原有数据均能按一种公用的和可控制的方法进行;数据被结构化,为今后的应用研究提供基础。当某个系统中存在结构上完全分开的若干数据库时,则说该系统包含一个数据库集合。

1.2.2 数据库系统模式结构

数据库系统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软、硬件资源组成的系统。尽管实际的数据库系统软件产品多种多样,但大多数系统在总的体系结构上都具有三级模式的结构特征。

数据库系统的三级模式结构由外模式、模式和内模式组成。

- **外模式**,亦称子模式或用户模式,是数据库用户看到的数据视图。
- **模式**,亦称逻辑模式,是数据库中全体数据的逻辑结构和特性的描述,是所有用户的公共数据视图。
- **内模式**,亦称存储模式,是数据在数据库系统内部的表示,即对数据的物理结构和存储方式的描述。

模式描述的数据全局逻辑结构,外模式则是数据的局部的逻辑结构,通常是模式的子集。

数据库系统的三级模式是对数据的三个抽象级别,系统提供三级模式之间的两层映象:

- 外模式/模式映象
- 模式/内模式映象

对应于同一模式,可以有任意多个外模式,外模式/模式映象定义某一个外模式和模式之间的对应关系,这些映象定义通常包含在各自的外模式中,当模式改变时,映象要作相应的改变,以保证外模式不变。

模式/内模式映象定义数据逻辑结构和存储结构之间的关系。例如,说明逻辑记录和字段在内部是如何表示的。当存储结构改变了,模式/内模式的映象也必须作相应修改,使得模式保持不变。正是由于这二级映象功能,使数据库系统中数据具有较高的独立性。

为了规定一个标准化的数据库系统结构和特征,包括数据库系统的接口和各部分所能提供的功能,1972年11月ANSI/×3/SPARC成立了一个数据库管理系统(缩写为DBMS)研究组,该研究组于1975年2月提出一个临时报告,1978年提出一个最终报告,简称为SPARC报告。

SPARC报告也提出数据库系统应具有三级模式结构,它们是:概念模式、外模式和内模式。其中外模式和内模式与上述含义大致相当,而概念模式被认为是一个部门或组织所对应的现实世界的真实模型,概念模式仅描述实体和它们的性质,不涉及机器世界的概念。

内模式对应于物理级模式,主要是定义数据的组织方法,如数据库文件的格式,索引文件组织方法,数据库在网络上的分布方法等;**概念模式**定义抽象现实世界的方法,是借助于数据模型来描述数据库的整体逻辑结构,数据库系统性能与数据模型直接相关;**外模式**对应于用户级部分,用户根据DBMS提供的子模式操纵处理数据。DBMS的中心工作就是实现三级模式之间的映射和转换,把用户操作转化到物理级执行。如图1-1所示。

由此可见,概念模式和模式都是描述信息或数据的整体结构的,它们是现实世界在不同层次上的抽象,概念模式是信息世界范畴的。

对于一个数据库系统,物理级数据库是访问数据的基础,设立概念级的目的是为了把用户视图有机地结合成一个逻辑整体,因此概念模式具有数据库整体逻辑描述和数据处理涵义。概念模式(Schema)的描述有图示(如:E-R图,实体—联系方法。)与语言描述,前者画出数据模型图,直观形象;后者使用数据语言,定义数据特性和操作,严格准确;用语言书写的模式称为**源模式**,将源模式用机器代码表示而变为机器使用的模式称为**目标模式**,为源模式设计良好的

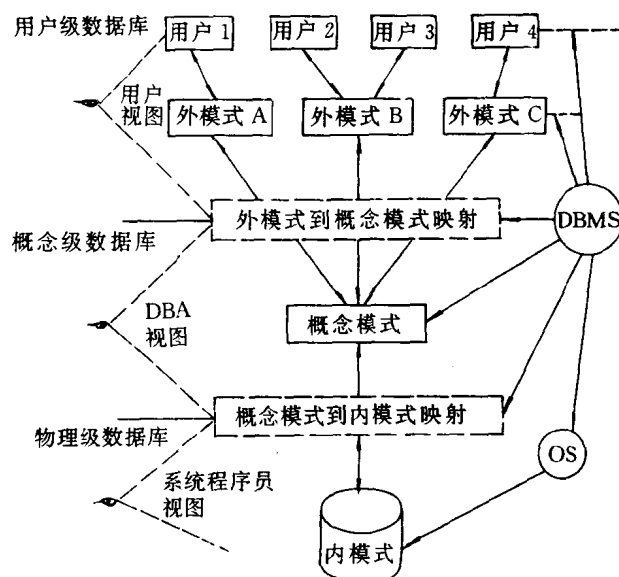


图 1-1 数据库系统分级模式

目标模式是很重要的工作。从模式导出子模式(Subschema),是用户与数据库的接口,每个用户可定义使用某个子模式,用户在授权情况下可作出不同于模式的改变,例如在子模式中,改变模式中某些数据项的数据特征,改变模式中的安全、完全约束条件等,子模式也有图示与语言描述两种表示方法,亦存在相应的源形式与目标形式,给用户以子模式的好处是:简化用户接口,方便使用,提供数据共享,减少冗余,孤立数据,安全保密。

广义地讲,实际的数据库系统由以下几部分构成:

1. 硬件及数据库。
2. 软件,包括操作系统、DBMS、系统工具软件、应用软件等。
3. 人员,包括数据库系统管理、开发和用户人员。其中,数据库管理员(DBA)负责全面管理和控制数据库系统,其职责包括:决定数据库的信息内容和结构,决定数据库的存储结构和存取策略,定义数据的安全性要求和完整性约束条件,监督和控制数据库的使用和运行,数据库系统的改进和重组等。

1.2.3 三种流行数据模型及数据库

数据组织是数据库技术的核心问题。数据库技术在处理数据组织时,是从全局出发,对数据内部联系和用户要求进行综合平衡来考虑的。在数据库设计中,曾流行三种数据组织方法:

1. 层次方法:数据按层次结构的形式进行组织;
2. 网状方法:数据按“有向图”的结构形式进行组织;
3. 关系方法:数据按关系形式,即表结构的形式进行组织。

上述不同形式的数据组织方法,对应着不同的**数据结构**,也对应着不同类型的数据库数据的**操作集合**和**完整性规则集合**。这三者组成了数据库的**数据模型(Data Model)**三要素。世界上流行的三种数据库数据模型是:层次模型、网状模型和关系模型。

一、层次模型(Hierarchical Model)

数据按层次划分,只有纵向联系,没有横向联系,实质上是一种有根结点的定向有序树。

在数学中,“树”(tree)被定义为一个无回路的连通图。树根被称为根结点,处于层次结构的最高层,并且仅有一个根结点(Root),树枝被称为边,枝端被称为结点(枝点),以结点表示记录类型,用边表示记录类型间的联系,两个记录类型之间的1对多联系是基本层次联系。层次模型是一种树结构,以记录类型为结点的有序树。

树结构的重要特征有:

1. 所有结点之间的连接都是从父结点到子结点。
2. 两个结点之间最多有一种连接。
3. 没有回路(loop)存在,即没有从自身出发又回到自身的连接。
4. 只有一个结点无父结点,该结点称为树的根。
5. 除根外,任何结点都有且仅有一个父结点。

所谓**有序树**,是对树中任一结点的所有子树都规定了先后次序的树。对有序树中每个结点都访问一遍,称为树的**遍历**(tree traversal),对结点扫描的先后次序规定,即树的遍历规则。

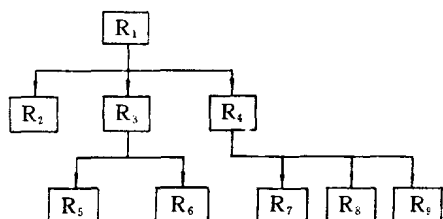


图 1-2 层次结构模型

由此可见,层次模型的特点是:记录类型间只有简单的层次联系,并且满足这样的条件,有一记录类型没有父结点,其它记录类型有且仅有一个父结点,父结点与子结点的联系是一对多,访问数据必须从根结点开始,按从左到右遍历子树。

图 1-2 是用层次模型表示的一个数据模型,每个方框为一结点,代表记录(数据集合),其中 R_1 为根结点。

按层次模型建立的数据库,称为**层次模型数据库系统**。

二、网状模型(Network Model)

数据的网状模型是以记录类型为结点,以它们的联系为络所构成的网络结构。

网状模型具有下述特点:

1. 允许有零个或多个结点无双亲;
2. 允许结点有多个双亲;
3. 允许两个结点之间有两种或多种联系,这种联系称为**复合链**。
4. 可能有回路存在。

在网状模型中,记录类型之间的联系允许有复合链,为了区分其中不同的联系,将对联系命名,这些命名的联系称为**系型**(Set type)或**络型**。

网状模型如图 1-3 所示。用数学方法可将网状模型结构转化为层次数据结构。

按照网状数据模型建立的数据库系统,称为**网状数据库系统**。

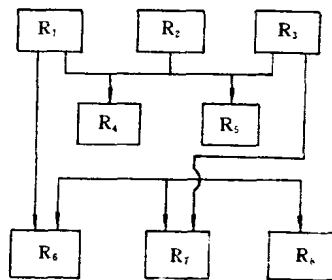


图 1-3 网状结构模型

网状模型的优点是有广泛的适应性,有强大的数据表示和组织能力,可描述实体间复杂关系。

DBTG(Data Base Task Group)是美国研究网状模型具有重大影响者,也是网状系统的典型代表。

三、关系模型(Relational Model)

把复杂的数据集合归结为满足一定条件的二维表格形式加以描述。二维表中的每一列对应实体的一个属性,其中给出各实体的相应属性值。每一行形成一个由多种属性组成的多元组,与一特定的实体相对应,表的顶端行指明了构成此关系的所有属性名。用这样的二维表形式组织数据,以集合运算为数学基础,建立关系操作和完整性规则,组成关系模型。

关系模型的特点是:实体的对象和联系均用关系描述,或者通过关系之间连接运算来建立联系。

关系模型是1970年美国E. F. Codd博士首先提出的,由于理论基础完整,关系二维表的概念又易于为用户接受,因而逐渐发展为占主导地位的模型。关系模型的优点是结构简明,操作方便,可通过集合运算、谓词演算等表示查询,一个关系就是一个数据表格,使数据库的设计简化。

关系模型常用术语:

- **关系**:用于定义一个数据表格,有 m 行, n 列。从理论上定义,关系是笛卡尔积的任意一个子集。
- **域**:每个列称为一个域或字段,其中含有属性值,对应于数据项。
- **元组**:每一行称为一个元组(tuple),如几个元素组成一个元组,称为 n 元组,即表示每个元组含有 n 个域的关系,每个元组都是一个 n 元组。元组对应于记录。
- **基数**:关系中行或元组的数目称为基数。
- **阶数**:关系中列的数目称为阶数。
- **属性值**:关系中各个元素称为属性值。
- **关键字**:标识一个元组(记录)的属性或属性组。

关系二维表的充要条件是:

1. 表中每列命名必须是唯一的。
2. 表中每列必须是基本数据项(即不可再分解)。
3. 表中每列具有相同数据类型。
4. 表中有一列能唯一标识它所在的行。
5. 表中各行是不重复的,即没有完全相同的行。
6. 表中行列的次序不影响所表示信息含义,可以任意交换。

按关系模型建立的数据库系统称为**关系数据库系统**。在关系数据库中,对管理对象可划分为一个或多个关系(表),通过对这些关系表的分类、合并、连接或选取等运算来实现对数据的管理。

四、数据模型间的相互转化

上述三种数据模型中,关系模型发展较晚一些,但应用发展速度很快,其普及型应用软件已经为社会广泛运用。这三种数据模型随着应用需求和技术进步,今天已得到多种扩展和改进。

同一客体可以用不同的数据模型描述,并且可以互相转化。如对网状模型加以限制就能变成层次模型,网状模型中的每一个记录类型和系型可变成关系框架,即关系模型。反之,把描述对象和描述联系的关系框架都变成记录类型的络型,关系模型就变成了网状模型。