

ORACLE7

数据库系统导引

刘向民 洪展志 姚 航 编著
姚卿达 主审



中山大学出版社

序 言

数据库的历史已有 30 多年,一直是计算机领域的热门话题。由于数据库是计算机应用和信息处理的基础,所以长期保持着繁荣的发展势头,积累了丰硕的成果。回顾数据库系统的发展,在不同年代有如下一些重要成果:

60 年代末,出现了层次模型系统 IMS (1969)、网状模型系统 IDS (1968—1969) 和 DBTG, C.W.Bachman 因对数据库系统作出重大贡献,于 1973 年得到计算机界的最高荣誉——图林奖。

70 年代是数据库开始大发展的时期,一方面层次与网状模型的系统迅速发展,出现许多 DBMS 产品,得到了广泛应用;另一方面出现了关系数据库,如 SYSTEM R, INGRES, QBE, SEQUITUR, DB2 等, E.F.Codd 提出了著名的关系理论,并于 1981 年获得图灵奖。数据库语言 SQL 雏型于 1974 年问世,1976 年发表,1979 年 ORACLE V2 问世。

80 年代是关系数据库大发展时代,1982 年 IBM 推出 SQL/SD,1981 年 dBASE II 进入市场,1984 年 dBASE III 问世,1987 年 Foxbase 问世,1986 年 ANSI 将 SQL 定为工业标准,这是数据库语言走向标准化的第一步。在 80 年代里,先后推出了 SYBASE、INFORMIX 等高性能 DBMS。在分布式数据库、知识库和演绎库、面向对象数据库方面的研究显得很活跃。

90 年代,以 FoxPro 为代表的微机数据库广泛应用,可视化的 DBMS 受到重视,如 Visual FoxPro 3.0 (95.5), Oracle Workgroup 2000 (95.3), 一些高性能数据库 (ORACLE, INFORMIX, INGRES, SYBASE) 竞相发展。90% 以上的计算机应用离不开数据库支持,以 INTERNET 为代表的信息高速公路的迅速发展普及促进了多数据库系统、联邦数据库系统的发展,1992 年提出了异种数据库互操作的标准——ODBC (开放式数据库互连),许多数据库服务器在网上运转,数据库与多媒体技术的结合,产生了多媒体数据库服务器。

展望未来,数据库前途无限,应用越来越广泛。1995 年,美国国家科学基金 NSF 工作组提出了关于未来数据库系统研究的报告 (Mag26-27, 1995), 后来著名的数据库专家 AVi Siberschatz, Mike Stinebraker, Jeff Ullman 等提出《数据库研究:迈向 21 世纪的成就与机遇》的重要报告,谈到当今社会由 Internet 携带的信息量每月增长 15%—20%,也就是每年增长一个数量级,数据库与数据库技术在这信息爆炸中将扮演一个关键角色。文中提出了面向 21 世纪的研究课题,其中包括:

1. 对多媒体对象加入到 DBMS 的支持技术。如第三级存储 (光盘箱,磁带仓),新的数据类型,服务质量,不确定查询 (Multiresolution Queries); 用户接口支持。
2. 信息分布相关的一些问题,如多数据库各成员自治程度,收费和记帐,安全和保密,复制和一致,数据集成和转换,信息检索和发现,数据质量。

3. 数据库系统的新用途。如数据采掘 (Data Mining), 知识发现, 数据仓库 (Data Warehouses)。

4. 新的事务模型。由于数据库的日益普及, 其应用方式也有了变化, 现在的事务可能跨越多个独立的数据库, 其执行时间也不再是只持续很短一段时间。为了适应新的事务处理方式, 须研究新的 workflow 管理, 新的事务模型。

5. 数据库管理使用的简易性问题。如用户界面是广大用户都关心的事, 目前 DBMS 安装、配置与升级显得复杂, 应智能化, 设计安装与维护的电子能手, 加强 DBMS 服务支持, 建立电子“物理数据库设计工具”。

ORACLE 是一个高性能数据库管理系统, 目前是最优秀的 DBMS 之一, 在国内外具有广大的用户。ORACLE 具有下列特点:

第一, 标准化程度高, ORACLE 实现了基于 SQL-89 的数据库标准的关系型数据库管理系统, 并作了一些扩充。ORACLE 7.1 与 Entry Level SQL92 完全兼容, 具有强大功能, 基本功能有 SQL * PLUS, PL/SQL, SQL * FORMS, PRO * C 及各种主语言预编译器, SQL * ReportWriter, SQL * MENU, DBA, SQL * NET。

此外有完整的工具系统, 如 CASE 设计与生成器, 应用系统和办公自动化系统开发工具; 还有图形与多媒体工具, 如 Oracle Graphics, Card, Book, Data Browser, Glue, WG/2000 等。

第二, 具有强大的网络功能和分布式功能, 可以在几乎所有计算机平台 (包括大型机, 并行机, 小型机, 微机, 工作站, 各种网络) 上运行, 可以组成各种计算模式, 如集中式, 客户服务器方式, 分布式等, 提供了不同环境的数据共享方案。

ORACLE 支持十多种网络协议, 如 TCP/IP, DECnet, SPX/IPX, Net BIOS, AP-PC, VTAM, ... 等等;

第三, 开放性好, 支持 ODBC 开放式数据库互连协议, 能方便地与异种数据库互操作和数据转换。

第四, 技术先进, 采用了一系列先进技术, 如并行查询与优化技术, 支持超大规模数据库 (VLDB——达 100GB 以上) 技术, 并行操作, 多线程服务器, 灵活的触发技术, 完整性与一致性措施, 安全性措施, 数据恢复措施, 分布式技术, 多媒体技术与可视化技术, INTERNET/WWW 连接技术等。

第五, 可移植性好, 不同平台开发成果可重用, 升级方便, 技术支持较好, 公司信誉高。

学习 ORACLE, 不仅可以掌握其使用方法, 更重要的是可以深入地解剖 DBMS 的技术细节, 真正进入数据库世界的开发与研究工作。

本书的内容包括①数据库系统综述; ②SQL * PLUS; ③PL/SQL; ④SQL * FORMS; ⑤PRO * C 及各种主语言预编译器; ⑥SQL * ReportWriter; ⑦SQL * MENU; ⑧ORACLE 体系结构及其管理。共八章, 覆盖了 ORACLE7 的基本原理和功能, 全书内容充实、详细。

这本书的前身是 ORACLE 联合大学中山大学分部所用的《ORACLE 数据库系统教程》(1994 年 5 月), 在姚卿达教授指导下, 由刘向民、洪展志、姚航共同编写, 经过

了两年的教学考验。在正式出版前，又由刘向民老师作了全面修订，补充了有关的基本概念和内容，对原教程不准确之处作了校正。全书以教材形式叙述和讨论，各章附有练习或实习题，采用边讲边用的方式教学。

本书适用面广：一可作为大专以上学生的数据库教材；二可作为 ORACLE 专业培训教材，培训数据库设计、维护和开发人员；三可作为实用工具书，在设计与使用时作参考手册；四可作为自学丛书，供具有大专以上的计算机应用人员参考。

本书编写过程中，中山大学 93 届数据库方向的研究生协助收集和编译了部分资料，ORACLE 联合大学学员在教学中提出了一些宝贵意见，中山大学出版社吴相辉同志为正式出版做了大量工作，在此对所有为本书作出贡献的同志表示感谢。

目 录

第1章 数据库系统综述.....	(1)
1.1 数据库	(1)
1.2 数据模型	(2)
1.3 数据库管理系统.....	(10)
1.4 数据独立性.....	(11)
1.5 事务处理.....	(13)
1.6 数据库管理员.....	(14)
1.7 数据库设计.....	(15)
第2章 SQL * PLUS	(17)
2.1 ORACLE 数据库的基本概念	(17)
2.2 SQL 的基本概念	(18)
2.3 数据库查询.....	(19)
2.4 数据操纵.....	(28)
2.5 创建表和视图.....	(31)
2.6 SQL * PLUS 的报表	(35)
2.7 函数.....	(41)
2.8 复杂查询.....	(47)
2.9 特权和角色.....	(51)
2.10 索引	(53)
2.11 并发控制	(54)
第3章 PL/SQL	(65)
3.1 PL/SQL 的实现结构	(65)
3.2 PL/SQL 的基本组成	(66)
3.3 SQL 和 PL/SQL	(70)
3.4 过程控制.....	(72)
3.5 游标.....	(77)
3.6 异常处理.....	(80)
3.7 子程序与程序包.....	(83)

第4章 SQL * FORMS	(91)
4.1 SQL * FORMS 简介	(91)
4.2 使用 SQL * FORMS (运行 FORMS)	(93)
4.3 FORMS 设计	(102)
4.4 用缺省块进行应用设计	(108)
4.5 块和域的修改	(114)
4.6 PAGE 的设计	(118)
4.7 触发器概述	(120)
4.8 触发器的类型	(124)
4.9 PL/SQL 的使用	(127)
4.10 用户出口程序	(134)
第5章 ORACLE 预编译器及 Pro * C	(137)
5.1 引言	(137)
5.2 程序接口的基本概念	(137)
5.3 程序首部	(143)
5.4 程序体	(153)
5.5 提交和滚回	(164)
5.6 错误检测和处理	(165)
5.7 使用宿主数组	(168)
5.8 动态 SQL 语句	(174)
第6章 SQL * ReportWriter	(180)
6.1 报表处理器功能	(180)
6.2 报表处理器的使用	(183)
6.3 灵活的报表编制功能	(194)
6.4 复杂报表的编制	(203)
第7章 SQL * MENU	(210)
7.1 SQL * MENU 概述	(210)
7.2 使用 SQL * MENU	(213)
7.3 设计菜单	(219)
7.4 用 SQL * MENU (Design) 建立一个基本的的应用	(223)
7.5 使用高级设计特性	(234)
7.6 SQL * MENU 和 SQL * FORMS 的集成	(249)
7.7 SQL * MENU 的内核和管理	(252)

第 8 章 ORACLE 体系结构及其管理	(258)
8.1 ORACLE 体系结构	(259)
8.2 启动和关闭实例	(277)
8.3 数据库存储管理	(283)
8.4 ORACLE 软件的安装	(300)
8.5 建立数据库	(303)
8.6 数据库的备份和恢复	(322)
8.7 数据库用户管理	(350)
8.8 控制数据库特权	(358)

第1章 数据库系统综述

数据库系统是信息系统的核心,一般地说,综合的信息系统就是大型数据库应用系统。数据库技术自60年代后期问世以来,无论从理论上、技术上,还是应用上,都经历了一个飞速发展的过程。在70年代,广为流行着网状和层次的数据库系统,这样的数据库系统使用记录集合上数据定义语言及数据操作语言来提供重要的DBMS功能,我们称为第一代数据库系统,DBTG和IMS是第一代数据库系统的代表;到了80年代,第一代系统大部分已被当今的这些关系DBMS所取代,我们称为第二代数据库系统。对许多应用来说,关系数据库系统被认为是在第一代系统上前进了一大步,因为它们采用了非常直观易懂的关系表做为数据的描述方法,同时使用非过程语言来操作数据,而且数据库的组织结构具有高度的数据独立性。第二代系统的典型代表是ORACLE、DB2、SYBASE和INGRES。从数据库技术的现状来看,以第二代系统为基础的实际应用已比较成熟和深入。但伴随着新技术的出现和应用的进一步深入,又出现了新的应用前景和新的应用需求,例如:计算机辅助设计,计算机辅助软件工程和超文本应用,它们实际上可能需要一种不同的具有特定功能的DBMS,这种DBMS除了传统的数据管理服务外,还应该支持更加丰富的对象结构和规则。目前面向对象数据库系统、多媒体数据库系统、联邦数据库系统等等都是数据库技术中的热门话题,随着这些研究成果的成熟并实现,数据库技术和应用将继续得到快速发展。

数据库技术中有许多概念和技术方法,它们有着特定的含义和特别的处理要求。我们只有准确地了解了这些概念和有关的处理技术,才能对数据库技术有一个全面和本质的理解,才能在此基础上有效地实现我们的应用。

1.1 数据库

数据库技术中的基本概念之一是数据库,它往往被作为数据库系统的同义词来使用。严格来说,数据库只是数据库系统的一个组成部分,是数据库系统中的核心资源。它是这样定义的:

数据库是大量的、经过加工整理的数据的集合,这些数据是现实世界中的某些有关的信息,以计算机可接受的形式表示出来并存储在计算机的存储设备上的,目的是能向使用者提供快速、准确、有效和及时的信息,为多种应用服务。

而数据库系统则是与数据库有关的计算机各成份的总和,主要包括:第一,基本的计算机软件和硬件,这是数据库系统运行的支撑环境,其中硬件设备应有中央处理器、主存储器、外存储器、数据通信设施等及其有关的处理和驱动设施;软件设备应有操作

系统、语言处理器和各种相应的实用程序；第二，数据库管理系统（DBMS），它是数据库系统的核心，根据所需的数据模型具有不同的性能和功能；第三，数据库，是数据库系统的基本成份；第四，数据库管理员（DBA），是一组负责数据库的维护和设计的人员，他们熟悉计算机数据处理技术和熟悉企业信息处理业务，保证数据库系统能正常运行，进行日常的数据库系统的管理。有关数据库系统中的一些主要成份我们在后面还会详细阐述。

从数据库的定义可知道，数据库是一种有组织、结构化的数据的集合，对数据的这种结构化处理的目的是能够为应用提供出快速和有效的信息。其实，这种处理方式和想法在日常的事务处理中也经常会用到，例如：我们往往会把自己的案头资料整理和分类，将已处理和未处理的文件分开；将属于不同主题的文件分开；按时间顺序来编排它们；等等；这样，当我们要找出某一文件时，就可以按照既定的分类方式快速准确地定位它。在数据库中我们使用数据模型来刻画数据库的结构，同时，数据模型也规定了数据的存储方法。数据库的另外一个特点是大的数据量，一个好的数据库系统应该是能有效地管理和控制大量的数据，能提供多种应用所需的数据。一般的软件系统要处理的数据可能是少量的，那么组织和管理这些数据可以简单地依靠操作系统的文件管理功能来实现，不用太多地关心数据如何安排和存放，而系统的时空效率可以令人满意；但作为数据库系统，数据库中能保存几十万到上百万条记录应该是普通的应用需要，甚至更大量的数据也应该能够处理，那么如何组织和管理这些大量的数据，如何安排和存放它们，如何快速、高效地操作它们则是要考虑和解决的问题，所以在数据库系统中要采用一些特别的技术和方法，如：规定数据存储的物理模型，进行合理的内存缓冲，采用多进程管理等等。数据库的最后一个特点是，数据库中的数据应该是能够同时服务于多个用户的多种不同应用目的的需要，也就是可共享的。因为作为一个组织的信息中心，它所服务的对象和服务的内容应该是多种多样的，而且应该在用户需要时就能提供出来，那么如何解决用户对数据库的并发访问，如何保持数据库的完整性、一致性和安全性就是要考虑的问题，作为数据库系统，对这些问题都作了妥善和合理的解决。概括起来，数据库有三大特点：结构性、有效性和共享性。

1.2 数据模型

客观世界中存在着形形色色的事物，它们的表现以及相互之间的关系是我们关心和要描述的对象，为了把它们在计算机中表示和模拟出来，并对它们进行处理，就需要把它们模型化，数据模型正是这种模型化的结果，所以，数据模型可以这样来定义：

数据模型是我们以一种形式化的方法，把客观世界的事物以及它们相互之间的关系抽象表示成计算机可以接受的一种形式，这种形式包含了一组有关的语义约束规则，规定了客观对象如何被表示，如何结构化和存储，以及规定了允许对所表示的结构进行何种操作。

数据模型规定了数据库的结构特征，是对客观世界的模拟，那么我们采用不同的方法以及从不同的角度对客观世界进行模拟就会得出不同的数据模型。换句话说，具有不同的结构形式和组织形式的数据库对应了不同的数据模型。从数据库技术的发展历程来看，确实有过多种数据模型，它们当中典型的有三种：网状数据模型、层次数据模型和关系数据模型。

1.2.1 概念模型

现实世界中事物千姿百态，但它们总是能相互区别的，因为各个事物总有自己的“特征”。例如，人的特征是：姓名、性别、年龄等。书的特征是：书名、作者、出版社、出版日期等，我们用事物的特征来区分各个事物。另一方面，事物之间又是息息相关的，例如：人和空气，植物和水，学生和教师等。为了把现实世界中我们所关心的事物的特征和关系描述出来，首先要对其进行概念的抽象，并以一种较为规范的形式来表达，这种形式不是计算机直接可接受的，但可由此方便地过渡到数据模型的表示，我们称之为概念模型。概念模型着重于现实世界的本质描述，主要表达客观对象的特征和它们之间的关系这样一些语义信息，而不考虑数据的操作和存储，是独立于具体的数据模型的，所以它是一种更加抽象的描述，有利于数据库的设计。概念模型中使用三个基本概念：实体、属性和关联。

实体 (Entity)

凡是客观世界中存在的且可识别的事物都可以认为是一个实体。例如，一个人是一个实体，一台电脑是一个实体。此外很重要的一点是，抽象的概念也可以看作是一个实体。例如，一项主张，一种看法都可以认为是一个实体，只要在实际应用中有这种需要。在一个个各自独立的实体当中，那些具有相似特征的一群实体可以组成一个实体集，例如，世界上所有的人组成了人这种实体集，所有计算机方面的出版物组成了计算机书籍这种实体集，如何划分实体集要根据实际应用的需要来确定。这里需要说明一下，在提到实体时，有时是指单独的一个实体，有时是指实体集，我们要根据上下文来理解。

属性 (Attribute)

是客观事物特征的抽象，客观事物及其特征在概念模型中则表示为实体及其属性。属性又分为二级的抽象：属性名与属性值，属性名刻画了一个实体的某一特征方面，而属性值则表示了这个实体在这个特征方面的一个具体的状态。联系于实体集和实体，一个实体集可以由一个属性名的集合来表示，而这个实体集中的一个实体则是由这个属性名集的每一个属性名下的一次取值所组成。例如，某单位的所有职工组成了一个实体集，它的属性名集合是：

部门名，职工编号，职工姓名，年龄，性别，...

而这个单位的每一个职工则可以表示为：

部门名	职工编号	职工姓名	年龄	性别	...	
人事部	1001	王 武	25	男	...	
会计部	4010	杨晓敏	45	女	...	
...	...	...	...	...	...	

表中的每一行代表了一个实体。

关联 (Relationship)

客观世界中事物间的联系可以抽象为概念模型中实体集间的关联。例如，学校中教与学的联系，可以表示为教师实体集与学生实体集间的关联；产品的生产关系可以表示为产品实体集与产品生产者实体集间的关联，等等。除了二个实体集间的关联外，也有多个实体集间的关联，如工厂，产品，用户这三个实体集间，就可以定义用户使用工厂的产品这样一个关联。还有，一个实体集可以与自己关联，如职工实体集中，实体间有上下级关系，要反映这种领导关系就要使这个实体集与自己关联起来。进一步分析关联的函数性质，我们可以总结出三种关联类型：

一对一关联： 任何一个实体集中每个实体与另一个实体集中的实体最多只有一个联系。
(1 : 1) 例如，表示管理关系的部门与经理之间的关联可以认为是一个一对一的关联，因为一个部门只有一个经理，而一个经理只管理一个部门（不考虑特殊情况）。

一对多关联： 一个实体集 E1 中的每个实体与另一个实体集 E2 中的零个或多个实体相联系，而 E2 中的每个实体最多只与 E1 中的一个实体相联系，则称 E1 与 E2 之间的关联是一对多的。例如，部门与雇员之间的关联可以认为是一对多的关联，因为一个部门可以有多个雇员，而一个雇员通常只属于一个部门。

多对多关联： 两实体集中实体间的联系没有限制。例如，学生与教师的关联可以认为是多对多的关联，因为一个教师可以教多个学生，而一个学生又可以上多个教师的课。

关联的这些函数性质也适用于多个实体集间的关联。另外，在实际应用中，我们认为某一关联具有某种函数性质总是基于一定的假设、排除了某些异常情况及对问题进行了简化处理的，否则的话，处处都是多对多的关联，问题就不容易解决。

在实际问题分析中，如何运用概念模型的这三个概念没有一个绝对的标准，也就是说，对同一客观事物或现象，有的人认为它是一个实体，有的人认为它是一个关联，甚至是属性都是可能的，这主要取决于我们看问题的层次与方法。所以说，这三个概念是可以相互转化的，对此我们应很好地认识，要根据实际问题的需要和处理的需要来掌握。

概念模型的表达通常是使用实体关联图 (E-R 图)。在 E-R 图中, 我们使用方框表示实体集, 在框内写上实体集的名称; 用椭圆框表示属性, 在框内写上属性名, 并把它与所描述的实体集的方框用线条连接起来; 用菱形框表示实体集之间的关联, 在框内写上关联名, 用线条把它分别与相关的实体集的方框连接起来, 并在线条的旁边标注关联的函数性质。它可以表示不同实体集间的关联, 也可以表示同一实体集内的关联。图 1.1 是 E-R 图的一个例子, 它简要描述了教师教学与科研的情况:

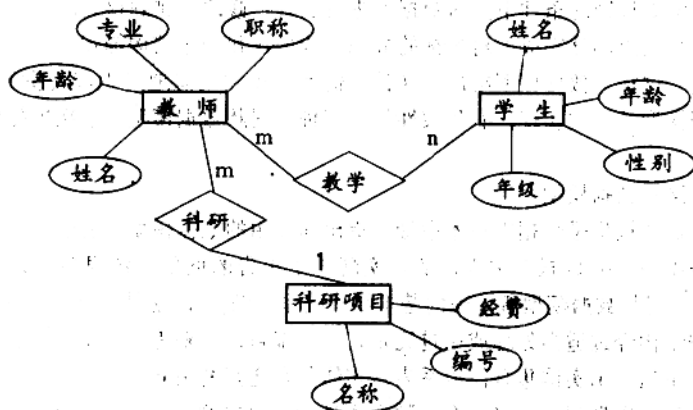


图 1.1 教师教学与科研情况的 E-R 图表示

在这个图中, 我们假定一个教师只能参加一个科研项目; 如果实际处理的问题是一个教师可以参加多个科研项目, 那么我们可以把“科研”这个关联改为多对多的。

1.2.2 数据模型

在我们建立了问题的概念模型后, 接下来是在概念模型的基础上, 建立问题的数据模型, 把它表示成计算机可以接受的形式。从前面的数据模型的定义中可归纳出, 数据模型应有三部分的内容:

- 数据及数据间的联系;
- 数据上的操作;
- 数据的语义约束规则。

数据及数据间的联系是数据模型的基本成份, 由它来代表客观世界在数据库中的表示, 即实体、属性和关联的表示。通俗来讲, 即规定构造数据库的数据结构类型。由于数据模型是向用户提供的, 因此它所规定的基本数据结构类型应该是最简单、最基本、且又能被用户理解的。当然, 还应有足够强的表达能力, 能有效地表达出数据之间的复杂的语义关系。

数据上的操作是用来操纵由数据及数据间的联系组成的数据库的。这些操作包括对数据库的任何部分进行检索、修改、定义等。数据的操作规定了对数据库的有效值的操作规则和导出规则等。

数据的语义约束规则又称为完整性规则，它规定了如何使数据库的状态（可理解为数据库的当前值）与它的数据模型相一致。当有了结构和操作的定义后，数据模型还必须给出对于该模型建立的数据库在操作后不会破坏其数据语义的通用的约束规则。只允许在满足该组规则的前提下，对数据库进行插入、修改、删除操作。

这三个组成部分完成了对数据的静态结构描述和动态结构描述，以及动态特性的描述和约定。这样，数据模型给出了在计算机上描述和动态地模拟客观世界的信息结构。作为数据模型的具体实现，有代表性的是三种：网状数据模型，层次数据模型和关系数据模型。

一、网状数据模型

网状数据模型是数据库技术中最早提出和使用的数据库模型，它是由 CODASYL 的数据库任务组在 DBTG 报告中提出的。实现这个报告的商用系统很多，其中典型的代表是 IDMS。网状数据模型所处理的基本结构是系（SET），这是一种二级树，任何一个复杂的数据结构经过适当处理或引进冗余之后，都可以分解为系的集合。在网状数据模型中，以记录型表示实体集合，以系表示实体集合之间的关联。

记录型是数据项的命名集合。数据项是命名数据的最小单位，对应于概念模型中的属性。一个记录型可对应一组记录值，两个记录型之间的关联限制为一对一或一对多的映射；当处理多对多的关联时，要增加中间记录型，称为连结记录，以使多对多关联转换为多对多关联。例如：学生 S 与课程 C 之间的关联是多对多的，设记录型分别为：

学生记录型 S: Snumber Sname Sage Sother

课程记录型 C: Cnumber Cname Cother

以 S-C 表示 S 到 C 的关联，这是多对多的；引进中间记录型 SC:

中间记录型 SC: Snumber Cnumber

以 S-SC, C-SC 分别表示 S 到 SC 和 C 到 SC 的关联，则 S-SC 是一对多的（一个学生选学若干课程），C-SC 也是一对多的（一门课程由若干学生选学），这样，通过 SC 的引入，便可将 S-C 的多对多关联转换为两个一对多的关联了。

系是记录型的命名集合，用来描述实体集间的关联。一个系有一个首记录型，以及一个或若干个属记录型，它们组成一棵二级树，如图 1.2 (a) 所示。首记录型与属记录型之间为一对多关联（指记录值的联系），首记录型又叫做主记录型，属记录型又叫做成员记录型。一个系可以包含一个或多个连结链，图 1.2 (a) 的系有二个连结链，分别代表了 DEPT 与 TEACHER 的关联和 DEPT 与 STUDENT 的关联。在实际处理中，往往需要将一个记录型作为一个或多个系的首记录型，或作为一个或多个系的属记录型；也可以作为某些系的首记录型，而同时又是其它系的属记录型，这些都是允许的，但一个记录型不能在同一个系里出现一次以上。系的一个值由一个而且只能有一个首记录的值和零个至任意多个属记录的值所组成。例如，我们要描述学校中各系（所）的教

师和学生的情况，建立了图 1.2 (a) 所示的系，图 1.2 (b) 是这个系的一个值：

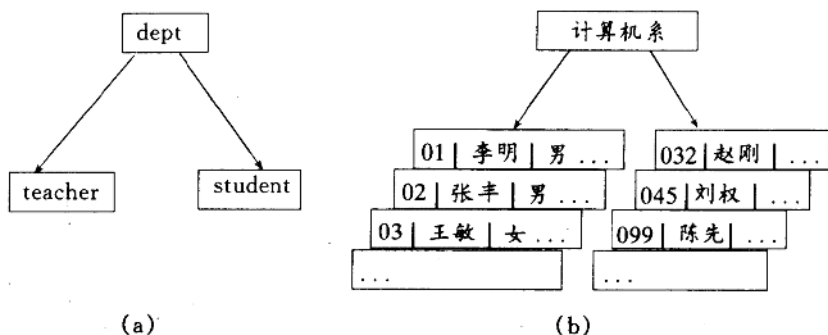


图 1.2 系和系值

对于一个给定的数据值，有几种存取途径：可以从一个首记录值向一个属记录值直接存取；从属记录值向前或向后顺序存取；从属记录值向首记录值直接存取。为了方便各种存取，应对系值中的属记录值进行排序，例如按关键字的递增或递减次序排序；按记录值插入的时间先后排序等等。在一个系值中，首记录值与一系列属记录值的联系可以用指针和阵列等办法来实现。在存取系中的一个记录值之前，往往需要先确定系值，即确定系首记录的一个值，然后再从这个系值的属记录值中选择所要的记录。

网状数据模型通过模式描述语言来描述，使成为数据库系统可接受的形式。用模式描述语言定义一个网状数据模型类似于用某种高级语言编写程序，只要掌握了相应的语法规则就可进行，关于这种语言的详细说明，可参阅有关的资料。数据库系统接受了模式描述语言所描述的数据模型后，就将它转换为目标形式，建立起数据库的框架。之后，我们就可以对数据库进行操作了。为了操作数据库，我们要使用数据库系统提供的数据库操作语言来进行，而这种语言又要嵌套在一种高级语言中使用，如嵌套在 COBOL 语言中。嵌套了数据库操作语言的高级语言程序的执行由操作系统和数据库管理系统共同支持，程序中涉及文件的传统操作由操作系统去完成，而那些牵涉到数据库的操作则由数据库管理系统来处理。

二、层次数据模型

层次数据模型所处理的基本结构是树，树由根片段型（相当于主记录型）和若干从属片段型（相当于成员记录型）组成，片段之间的层次联结称为父子联结。片段描述实体集合，片段中包含若干个字段（相当于数据项），用于描述属性。片段之间的父子联结反映了实体集之间的关联，限定为一对一或一对多的。如图 1.3 所示：

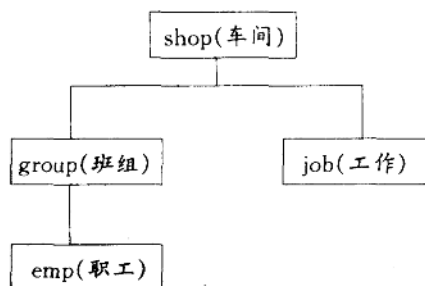


图 1.3 层次数据模型

采用层次数据模型的典型系统是 IMS。IMS 所处理的基本结构是层数不超过 15 级的树结构，为了处理网状结构，可以定义树之间的逻辑关系，还可以定义辅助数据结构，所以 IMS 模型具有描述任何复杂数据结构的能力。用 IMS 术语来讲，所处理的基本结构叫做物理数据库记录型（PDBR），一个 IMS 的数据模型是所有的 PDBR 的集合，PDBR 的一个值由一个根片段的值和它所属各层从属片段的值之全体所构成，称为一个数据库记录。一个 IMS 的数据库就是由所有的 PDBR 的所有值，再加上索引数据库、辅助索引数据库和逻辑数据库等所组成。对数据库的存取是沿着层次路径来进行的，所谓的层次路径是指树中各片段型之间及一个片段中各值之间的顺序关系。一般来说，树中各片段型的顺序是从根片段开始按自顶向下自左而右的原则规定片段顺序；而一个片段内各值的顺序是按某关键字段值来确定。

IMS 数据模型的描述是使用 IMS 提供的物理数据库描述语言来进行的。使用有关的语句，我们可以定义一个 PDBR，规定有关的存储参数，指定存取方法等等。这样，IMS 数据库管理系统就可以按照这些描述建立起数据库。对于应用程序，要操作数据库也是使用一种宿主语言如：COBOL、PL/1，再嵌套 IMS 提供的数据库操作语言来进行。

三、关系数据模型

关系数据模型是当前普遍采用的一种模型方法，它的特点是数据结构简单且单一，由此而来的数据操作语言简单且是说明性的，用户使用方便直观，且操作功能强。因为它的操作只需用“关系代数”这个概念，且操作结果仍是关系，所以可以对其复合地、连锁地进行，这种操作性质可归纳为它是面向值的操作。关系的数学定义是：

域 $D_1, D_2, \dots, D_k$ 的笛卡尔积，记为 $D_1 \times D_2 \times \dots \times D_k$ ，是所有可能的有序 k -元组 $(d_1, d_2, \dots, d_k)$ 的集合，且 $d_1 \in D_1, d_2 \in D_2, \dots, d_k \in D_k$ 。如果 r 是笛卡尔积 $D_1 \times D_2 \times \dots \times D_k$ 的一个子集，则 r 是 $D_1, D_2, \dots, D_k$ 上的关系。

在数据库环境下，把关系看作表结构时，必须对数学概念下的关系的定义给予限定或扩充。主要有两个方面：第一，关系当作表结构时讨论无限关系是无意义的，因此在

关系模型中，需限定所有的关系都是有限的，这样才能存储；第二，关系在数学概念中列的顺序是有影响的；而在数据库环境下，列的顺序可用附在关系的列上属性名使其变成无关紧要，如图 1.4，代表了相同关系的两种表示：

姓名	性别	年龄	姓名	性别	年龄
李明	男	30	男	李明	30
王敏	女	29	女	王敏	29

图 1.4 相同关系的两种表示

在关系数据模型中，关系是用来表示实体集的，对关系命名并用它的属性名集合来描述就得到了一个关系模式，例如，关系模式：学生（姓名，性别，年龄）可以代表学生实体集。关系模式描述了关系的信息结构及语义约束，关系则是关系模式在某一时刻的“当前值”，是现实世界在某一时刻的状态的真实反映；一组关系模式组成了关系数据模型，对应于这组关系模式的当前值则组成了一个关系数据库。

一个关系数据模型有如下的特点：

- 关系模式中属性个数是固定的且带有名，同一关系模式中属性不能同名；
- 关系中属性是基本成分，不可再分，不可重组；
- 关系中各个元组是互相区别的，不允许重复；
- 关系中属性次序和元组的次序是无关紧要的。

在一个关系模式内，总有某个属性或属性集合，它们的值能唯一地标识出整个元组，这个属性或属性集合称为关系的键。通常能够成为关系的键的属性或属性集合会有多个，如果一个属性集合是关系的键，而它的任何真子集都不是关系的键，则这个属性集合可称为关系的主键。所以，主键是能够唯一确定一个元组，而又是最小的。此外，在一个关系模式内，属性间还可能存在着各种依赖联系，满足一定依赖情况的关系称为规范关系，通常有三种：1NF 规范关系、2NF 规范关系、和 3NF 规范关系。我们可以按照一定的规则将一个关系变成一个规范关系，并逐步提高规范化程度，这叫做关系的规范化。关系的规范化过程是一个可逆的过程，一组关系转换为另一组关系后，仍然可以恢复到原有的那组关系，规范过程并不丢失任何信息，并保持属性间的依赖性。关系通过规范化，可以消除冗余，提高操作的可行性及便利性。

关系数据模型建立后，我们可以对它的当前值进行操作，以追踪反映现实世界的当前状态。通常有查询、更新、插入、删除操作。在查询操作下，又有一系列特别的操作，可以使我们从数据库中得到或推导出某些特定的信息和结果，主要有投影、选择和连接。查询的投影是指对一个关系，去掉某些分量（属性）后所得到的一个新的关系，当我们只想了解一个实体集或关联在某些特定属性下的取值情况时，就可以使用这个操作；查询的选择是指在一个关系中，将符合某些条件的特定的元组（行）挑选出来；查

询的连接则是在两个关系的元组之间，按照某种条件把它们连接起来，使两个关系共同地描述某种情况。

关系的完整性规则通常是对关系的某种约束，这种约束是指对给定关系插入一元组、删除一元组时是否可以接受，或者涉及到一个关系中元组的插入、更新与另一个关系之间有某种联系时，这种操作如何接受。对于关系模型通用的完整性规则一般有两个：实体完整性规则和关联完整性规则。实体完整性一般是指在一个关系中，所有的元组在某些重要的属性栏的取值不能为空，如关系的主键，因为我们不能想像唯一标识某个实体的值是未知或不存在时，那个实体还有什么意义；关联完整性是指两个有关联的关系，如果其中一个关系 R1 的某一（些）属性是另一关系 R2 的主键，则 R1 中的元组在这个属性下的取值或者为空，或者是某个值，而这个值必须是在 R2 中有出现的主键值。

目前的第二代数据库系统都采用关系数据模型，本书所要介绍的 ORACLE 数据库系统是它们当中最成功的一个代表。ORACLE 是以 SQL 语言作为它的数据描述和操作语言，使用 SQL 语言，我们可以进行关系数据模型的定义，作出关系完整性约束的描述，实施数据库的操作。在后续章节中，我们会对 ORACLE 数据库系统作出全面的介绍，到时可以看到 SQL 语言的完整的使用方法。

1.3 数据库管理系统

数据库管理系统（DBMS）是处理数据库存取和各种管理控制的软件，它是数据库系统的中心枢纽，所有对数据库的操作都通过它来进行：由它负责把数据模型的定义转换成具体的计算机上的实现；负责将用户对数据库的逻辑操作，经过映射定义的转换，变成对物理数据库的具体执行；负责在定义和操作过程中的各种管理和控制；为使用数据库的用户提供一个接口；也为支撑它的操作系统等提供通信接口，形成一个在计算机系统上为用户使用数据库时的一个清晰界面。

一个 DBMS 总是立足于一定的数据模型的，数据模型的不同，那它对从现实世界中构造的数据库的概念结构的表示方法也不同，从而 DBMS 的管理方式也不同。DBMS 到目前尚没有统一的标准来界定它的功能特性等表现，而且，即使具有相同数据模型的 DBMS 在功能实现上也各不相同。虽然如此，一个比较完善的 DBMS 应该具有数据定义、数据操作处理、数据管理和控制、数据接口等功能。

数据定义是指对构成数据库结构（模式）的定义，包括：逻辑模式、存储模式、外模式（视图）、每一个外模式与逻辑模式之间的映射、逻辑模式与存储模式之间的映射；保证数据库中存储数据的语义正确和有效的完整性规则的定义；对存储数据的使用权限的定义等。这些定义通常是由 DBA 按系统提供的数据库定义语言的源形式书写，然后由 DBMS 自动地将其转换成内部目标形式，存入数据库字典中，供以后对数据操作或控制时查阅使用。DBA 是维护 DBMS 软件的特权用户，一般用户不承担书写数据定义的责任，但允许查阅部分数据定义。

数据操作处理功能实质上包含了两层意思，其一是对数据的操作，包括数据的初始