高等学校计算机专业“十一五”规划教材

数据库原理与应用

陈庆奎　主　编
彭敦陆　那丽春　霍　欢　副主编

西安电子科技大学出版社
http://www.xduph.com

高等学校计算机专业“十一五”规划教材

数据库原理与应用

主　编　陈庆奎
副主编　彭敦陆　那丽春　霍　欢

西安电子科技大学出版社
2009

内 容 简 介

本书是面向应用技术人才培养的教材，主要的特色体现在应用性和实用性。本书共分 11 章：第 1、2 章介绍数据库系统和数据模型；第 3、4 章介绍关系数据库理论与 SQL 语言；第 5 章讲解对象及对象—关系数据库理论；第 6 章为数据库事务管理和数据库恢复技术；第 7 章为完整性与安全性；第 8 章为数据库存储机制、网络数据库关键技术和数据库设计技术；第 9 章为 Web 数据库设计；第 10 章为机群数据库系统；第 11 章介绍网络数据库应用系统开发案例。

本书可作为普通高等院校计算机科学与技术专业、信息管理与信息系统专业以及其他相关专业的教材，也可供从事相关工作的工程技术人员参考使用。

图书在版编目（CIP）数据

数据库原理与应用 / 陈庆奎主编. —西安：西安电子科技大学出版社，2009.9

高等学校计算机专业“十一五”规划教材

ISBN 978-7-5606-2253-8

Ⅰ. 数… Ⅱ. 陈… 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（2009）第 144508 号

策　　划　臧延新

责任编辑　徐德源

出版发行　西安电子科技大学出版社（西安市太白南路 2 号）

电　　话　(029)88242885　88201467　邮　　编　710071

网　　址　www.xduph.com　电子邮箱　xdupfxb001@163.com

经　　销　新华书店

印刷单位　西安文化彩印厂

版　　次　2009 年 9 月第 1 版　2009 年 9 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　印张 22.875

字　　数　545 千字

印　　数　1～4000 册

定　　价　33.00 元

ISBN 978 – 7 – 5606 – 2253 – 8 / TP • 1146

XDUP 2545001-1

前　言

随着信息技术的飞速发展以及 Internet 应用的日益普及，数据呈爆炸趋势增长，数据的管理变得越来越重要。为了满足信息社会发展以及应用环境的需要，现代数据库应用技术普遍具有如下特征：(1) 数据的分布越来越网络化、异构化，因而数据库厂商纷纷更新它们的数据库系统产品，以便适应对网络化、异构化数据的管理需要；(2) 数据呈海量趋势发展，数据库技术必须面对海量数据管理的需要，因而以机群数据库为代表的海量数据存储和高性能处理技术已经走入实际应用领域，数据库厂商也推出相应的数据库产品，如 Oracle 的 RAC；(3) 面向对象的设计与开发技术日趋完善，已被广泛地应用到现实应用系统的开发过程，这导致对象数据库应用技术日趋成熟，现有的主流数据库产品几乎都以各种形式来体现面向对象的特征。

我国普通高等院校的招生规模已创新高，高等教育已进入普及时期，因而普通高等院校的学生需要分层次定位培养。全国 600 所高等院校中 500 所以上院校把学生的培养目标定位在应用技术型开发人员，这些学校和她们培养的学生更需要与实际应用结合密切的教学方法和教材，使学生所学的理论和技术与社会的需求尽可能接轨，进而使学生有较强的竞争能力。

本书是编者在多年的数据库原理教学经验的基础上，结合当前数据库应用的实际需要以及编者的项目开发的实际经验，通过收集毕业同学的反馈信息，依据培养面向应用技术型人才的目标而编写的。本书具有如下特色：

(1) 简捷性：去掉了传统数据库原理中冗长、枯燥的理论描述，用简洁的语言描述原理技术；

(2) 实用性和主流性：主要介绍关系数据库系统、对象-关系数据库系统，囊括了目前主要应用的数据库系统模型；

(3) 完备性：介绍的原理基本覆盖了其他教材的内容，具有原理的完备性；

(4) 全面性：引导学生从数据库运行的实际环境来分析和设计数据库应用系统，主要因素包括数据库的存储环境、网络支撑环境、全局与局部数据环境、高性能要求环境等；

(5) 实际性：通过某集团企业的应用模型介绍数据库的分析、设计、配置等一体化过程。

本书以简化理论、注重应用为宗旨，特别适合于普通高等院校的学生使用。

本书共分 11 章：第 1 章数据库系统概论，介绍数据库系统涉及的概念、结构、组成及历史；第 2 章数据模型，主要介绍 E-R 模型、UML 模型、XML 模型；第 3 章关系数据库与 SQL 语言，内容包括关系代数、关系模型、SQL 语言和查询优化技术；第 4 章关系数据库理论，内容包括范式理论、模式分解技术等；第 5 章对象及对象—关系数据库理论，内容包括对象模型、对象数据库、对象—关系数据库理论；第 6 章数据库事务管理，内容包括事务概念及特征、并发控制、数据库恢复技术；第 7 章完整性与安全性；第 8 章数据库设计，内容包括数据库设计过程、数据库存储机制、高速网络互联、网络数据库关键技术、E-R 模型的数据库设计、对象存储技术等；第 9 章 Web 数据库设计；第 10 章机群数据库，

主要介绍如何利用计算机机群来提高数据库服务器系统的性能；第 11 章基于 Internet 的集团公司财务数据监管系统，通过一个网络数据库开发实例来全面介绍网络数据库应用的分析、设计和配置过程。

本书的第 1、3 章由那丽春副教授编写，第 2、4、6、9 章由彭敦陆博士编写，第 5、7 章和第 8 章的 1、4、5、6 节由霍欢博士编写，第 8 章的 2、3 节和第 10、11 章由陈庆奎教授执笔。全书架构由陈庆奎教授设计并统编。本书的编写得到了各位编者家人的大力支持和帮助，也得到了 08 级博士生王海峰的支持，他收集整理了大量的资料，感谢他们对本书的形成所做的各种形式的贡献，还要感谢业界学者和同行的研究成果。由于编者学识浅显、见闻有限，必有不足和遗漏，希望同行指正。

编　者

2009 年 5 月于上海

目　录

第 1 章　数据库系统概论

1.1　数据库系统应用

数据库技术是一门既传统而又年轻的计算机科学技术，说其传统是因为它已经有几十年的发展历史，说其年轻是因为 Internet 时代赋予其新的使命和新的要求。数据库系统是存储、管理和提供数据服务的系统，它以一种简单、规范、便捷的方式存储管理数据。数据库的应用领域非常广泛，它已经成为当今企业信息建设的重要组成部分。其典型的应用领域如下：

(1) 金融业：用于存储管理债券、股票等金融票据的持有、销售信息等等。

(2) 销售业：存储管理客户、产品、销售、供应等信息。

(3) 制造业：用于管理供应链，追踪产品的产量、库存及客户的订单信息。

(4) 大学：存储管理学生的信息、课程及成绩，图书管理信息等。

(5) 航空业：用于存储订票和航班的信息。航空业是最早使用地理上分布的方式应用数据库的行业之一，通过各个终端或其他数据网络来访问中央数据库系统。

(6) 电信业：用于存储管理通话记录，生成账单报表，维护预付电话卡余额信息等。

从 20 世纪 60 年代开始，数据库逐步进入社会应用的各个领域。早期的人们很少直接与数据库打交道，但是通过月底的工资表、银行的账单却间接地与数据库打交道。后来随着数据库应用范围的进一步扩大，人们开始直接和数据库进行交互。比如用户通过浏览器查询航班数据库、了解航班的起降时间和票价；学生通过校园网络查询课程数据库，确定新学期的选课情况。大多数信息系统提供了友好的应用界面，它隐藏了数据库的内部细节，使得大多数用户没有意识到自己正在和数据库打交道。今天，访问数据库已经成为人们生活中最基本的组成部分。

1.2　数据库系统与文件系统

对大量数据的管理最初是采用文件管理方式，即将成批数据单独组织成文件存储在外部存储设备上。假设银行储蓄系统要保存所有客户和储蓄账户的信息，在没有出现数据库系统前保存信息常用的方法就是将它们保存为操作系统中的文件。为了便于用户对这些客户信息进行操作，系统程序员必然根据用户的需求设计一系列的文件操作程序，包括处理账户存款、贷款的程序；创建新账户、注销账户的程序；查询账户余额的程序以及打印报表的程序等。但是用户的需求是变化的，比如系统要求增加支票类型的账户。除了要增加新的文件来保存支票账户，还要修改程序或增加新的应用程序。比如储蓄账户不存在的透支问题出现在支票账户中了，必须增加新程序来解决需求的变化。这导致更多的数据文件

和处理数据的程序增加到系统中。

这个应用实例是典型的文件处理系统，该系统由传统的操作系统支持，信息存储到不同文件中，通过不同的应用程序将信息从适当文件中取出或放回适当文件中。文件系统存在许多弊端：

(1) 数据冗余和不一致。银行系统的软件要运行很长时间，在这个过程中需求会发生变化，程序开发人员也会有调整。因此不同的数据可能采用不同的文件格式保存，使用不同的程序存取必然导致相同的信息可能在多个文件中重复存储。例如，客户的个人信息既出现在储蓄账户文件中又出现在支票账户文件中。这种信息冗余不仅导致了存储和访问工作开销增大，还可能出现数据不一致性。比如工作人员在储蓄账户记录中修改了某客户的地址而没有修改支票账户的内容。

(2) 数据孤立。在文件系统中由于信息保存在不同的文件中，因此很难看到全局的信息，比如银行的高级管理人员想了解当前所有的储蓄账户和支票账户的转账情况，应用程序员也很难编写检索这类多种文件格式的新程序。

(3) 数据访问困难。如果银行管理人员想查询某一地区开户人员中小于 30 岁的储蓄账户的信息，而这种新的查询任务以前从来没有过。因此只能有两种办法解决，一种办法是这个管理员在所有客户的巨大列表中努力寻找，这是个繁杂、易错的过程；另一种办法是要求数据处理部门编写新的应用程序来解决，这需要一定的等待时间。这两个方案都令人失望。如果几天后，这个管理人员需要储蓄账户和支票账户的信息，查询条件不改变，则上述问题又再次出现。

(4) 完整性问题。文件中保存的数据必须满足某些特定的一致性约束。比如银行新增账户时要提供身份证号码，应用程序中增加相应的代码来保证信息的约束。当没有身份证的客户开户时将给予提示。如果加入新的约束时，必须修改应用程序，当涉及多个不同文件时，这种修改过程是个噩梦。

(5) 原子性问题。计算机系统如同其他的系统一样可能出现故障。例如客户 A 要把自己储蓄账户的钱转账到客户 B 的支票账户中。如果转账过程中计算机系统出现了故障，A 账户的钱减去了而 B 账户的钱没有增加，这是现实中绝对不能允许的事情，可见文件系统不能保证数据的一致性，这就需要账户的“加”和“减”这两种操作必须是要么做、要么不做的原子性操作。可是在文件系统处理阶段，保持原子性是很难实现的。

(6) 并发访问异常。现实世界是个并发的世界，经常出现多个用户修改同一数据的事情。例如账户 A 中原来已有 5000 元，此时两个客户同时向账户 A 转账，他们分别向账户 A 转入 500 元和 1000 元。假设转账操作是先读取账户 A 的余额，然后在其上增加或减少金额，最后将结果写回。如果两个转账程序并行执行，开始读到账户 A 的余额都是 5000 元，然后分别加上自己的转账数，再分别写回 5500 元和 6000 元。由于没有并发控制，后写入的数据将覆盖前面写入的数据，导致账户 A 最后的结果要么是 5500 元要么是 6000 元，但是本转账过程真正的结果应该是账户 A 拥有 6500 元。

正是由于文件系统存在很多不足，加速了数据库系统的发展。数据库系统很好地解决了文件系统中存在的问题，并能提供以下特性：

(1) 数据独立性(Data Independence)。所谓的独立性，是指应用程序独立于数据的逻辑表示和物理存储。将文件系统中定义的数据文件从应用程序中抽取出来，由数据库系统统

一实现和管理，因此当数据逻辑结构和物理存储结构发生变化时无需对应用程序进行修改。

(2) 有效访问数据(Efficient Data Access)。数据库系统中采用复杂的技术，通过优化系统资源配置和合理分布数据等工作来提高大规模数据量的存储与检索效率。

(3) 数据完整性与安全性(Data Integrity and Security)。数据库管理系统利用数据完整性约束或限制来保证数据库的数据不受系统故障和用户误操作而导致的数据破坏。同时，数据库系统采用安全认证、数据加密、访问控制等技术来实现数据的安全性要求。

(4) 并发访问与故障恢复(Concurrent Access and Crash Recovery)。多用户同时访问同一数据的时候，产生了并发访问，这可能导致数据的冲突。数据库系统可保证并发访问的顺利进行，解决并行化所带来的问题，同时还会让用户感觉到好像自己在独享系统。

1.3 数 据 视 图

视图(View)从最原始的意义上可以理解为一个人看(既“视”)某个物体所得到的图像。如果将这个概念引申到数据库领域中，则认为不同的用户对同一数据库不同部分的理解，称为该数据库的一个视图，或数据视图。这里的用户可以是普通计算用户或是不同的应用程序，他们对同一数据库所关心的数据可能是不同的，就如同很多人对同一事物有不同的视角。数据视图不仅是一个重要的概念，也是数据库系统中的一个实用的工具。

例如，对于学生信息数据库来说，档案管理部门、教务管理部门和卫生保健部门各自关心不同的数据，档案管理人员一般只想查看学生的基本档案信息；教务管理人员只关心学生的成绩和课程信息；卫生保健人员关心的是与学生健康有关的数据。因此有了 3 个不同的局部数据视图。如果想同时观察学生的全部数据记录，就得把 3 个不同的数据视图合并形成学生的全局数据视图。局部数据视图和全局数据视图都是学生数据的一种逻辑表示结构，这种结构易于实现计算机的访问操作。

数据库中的所有数据最终是存储在物理磁盘上的，如何在磁盘上存储数据、如何将数据分布在多个磁盘上或网络存储介质上、使用什么样的底层文件系统等都是数据库系统要解决的问题。从物理存储角度来看，数据有不同的存储结构。这些存储结构又叫数据库的物理视图。因此在数据库中存储的数据具有物理视图和逻辑视图两个侧面。而逻辑视图又有局部数据视图和全局数据视图。

1.3.1 数据抽象

数据抽象(Data Abstraction)是指人们对现实事物属性的一种数据抽取的过程。数据库系统必须能够快速检索用户所需要的数据，因此设计者必须使用复杂的数据结构来表示现实的数据。而数据库用户大部分不具备专业的计算机知识，因此系统通过多个层次的抽象来隐藏复杂性。

数据抽象采用多层抽取的思想，每一层都是其上一层的基础，这个过程是将复杂变简单的渐进过程。最后用户面对最高层的抽象结果，也就是最简单的数据形式而不用关心复杂的细节。首先，物理层抽象是抽象的最低层，它描述数据库中的数据是如何在硬盘上保存的。物理层详细描述了存储的数据结构。在物理层之上，以物理层为基础抽取一个逻辑

层，在逻辑层中描述数据库中存储数据的类型及相互的关系。视图层抽象是抽象的最高层次，只描述了数据库中某一部分的数据及其关系。例如企业员工信息数据库中保存了全部员工的数据，财务部人员只关心在职人员的信息，人事退休管理部门只关心退休人员信息等。视图层抽象提供了同一数据库的多个视图。数据抽象的 3 个层次如图 1.1 所示。

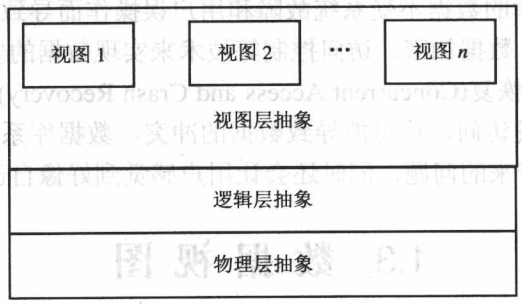

图 1.1　数据抽象层次结构

1.3.2　实例与模式

数据库所包含的数据往往是一个应用部门、集团或领域中各方面的数据，其涉及若干个现实中的对象，包括人员信息、资产信息、财务信息、物流信息、销售信息等等。这些信息存在固有的数据属性，同时信息之间也存在着天然的关联。数据库的设计与实现必须把这些属性和关联用计算机可以支持的模型表示出来。在一个数据库应用系统内描述对象的数据属性以及对象之间关联的数据框架的集合称为数据库模式。数据库设计工作就是要构建有效的数据库模式。数据库模式构建完成之后，就要向其中添加数据，随着时间的推移，数据操作会引起信息动态变化，但每一时刻数据库中的数据都会保持一个一致性状态。特定时刻存储在数据库中的所有数据的总和称为数据库模式在这一时刻的一个实例。

数据库模式和实例的区别可通过计算机程序设计中的概念加以类比说明。模式对应于程序设计语言中的变量声明，而每个变量在特定时刻都有特定的值，程序中变量在某一时刻的值是模式的一个实例。例如银行账户数据库的模式固定后，某一天的账户数据就是该模式的一个实例，第二天就有新的客户建立新账户或者老客户注销账户，因此数据的实例很容易变化，而模式也会发生改变，但是相对固定。

数据库系统根据抽象的层次不同存在几种不同的模式。在物理层描述数据库的物理存储设计是物理模式；逻辑模式是在逻辑层描述数据库的全部框架设计；数据库在视图层有子模式的概念，子模式描述了数据库不同的视图。根据分层设计的思想，程序员使用逻辑模式来设计应用程序，不直接操作模式，因此逻辑模式隐藏了物理模式的复杂性，同时也解除了应用程序与物理模式的耦合性。当物理模式改变时应用程序无需修改，这就是物理数据独立。

1.4　数据模型简介

数据是反映客观世界的事实并可以区分其特征的符号，包括字符、数字、文本、声音、图形、图像、图表、图片等，都是现实世界中客观存在的，可以输入到计算机中进行存储

和管理的数据。数据可以描述现实世界中一个客观对象，如一个企业员工的情况：工号、姓名、性别、年龄、部门等。数据通过对客观事物特性的定量化描述来支持计算机系统对客观事物的存储、加工和处理。数据模型是描述数据、数据之间关系、数据语义及一致性约束的概念集合，也是构建数据库系统的基础。下面介绍常见的几种数据模型。

1.4.1　实体－关联模型

实体－关联模型(E-R，Entity-Relationship)是分析现实世界抽象关系的有效工具之一。该模型认为现实世界是由被称为实体(Entity)的元素和实体之间的联系构成的。实体是现实世界中存在的可以相互区别的事物或者活动的数据抽象。比如员工信息管理中，职工、经理、部门、账户等都是实体。关联(Relationship)是指实体之间的相互关系，反映了现实中事物之间的联系，通常表示一种活动。比如一张订单、一场比赛、一种选题、一种销售等都是联系。属性(Attribute)是描述实体或关联特征的单元，一个实体或关联通常具有多个特征，需要多个相应的属性来描述。比如描述“课程”这个实体具有课程名称、任课教师、开设专业、学分等属性。实体集(Entity Set)是指同一类实体的集合。例如一个班级的全体学生是一个实体集。实体型(Entity Type)是对同类实体的共用特征的抽象定义。例如人的共有特征为：姓名、年龄、性别、籍贯、学历等，这 5 个特征就定义了人这个实体型。实体值(Entity Value)就是符合实体型定义的每个具体的实体。例如人的实体型使用“姓名、年龄、性别、籍贯、学历”5 个特征定义，而“李明，33，山东，教师，硕士”就是该实体型的一个实体值，它所描述的是现实中具体的人。E-R 模型可以用 E-R 图进行描述，E-R 图由以下几个元素构成。

(1) 矩形：代表实体集。

(2) 椭圆：代表属性。

(3) 菱形：代表实体集之间的关联。

(4) 线段：链接属性与实体集或链接实体集与关联的连线。

作为例子，我们看一下高校学生管理数据库中学生选课的部分，对应的 E-R 图如图 1.2 所示。在图中有学生和课程两个实体，每个实体包括一些属性，而这些属性并不是实体的全部属性，只是与该系统相关的部分。两个实体之间还存在“选课”的联系。

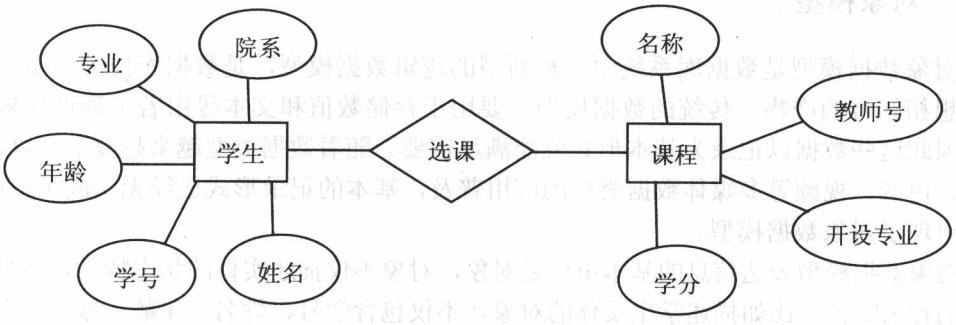

图 1.2　E-R 图示例

1.4.2　关系模型

关系模型是一种简单的二维表格结构，概念模型中的实体及实体之间的联系都可以转

换为二维表形式。每个二维表称为一个关系，一个二维表的表头，即表的所有列的标题称做关系的型或者是关系的结构，二维表内容称做关系的值。关系中的每一行数据(记录)称做元组，而它的每一列数据称做一个属性，列标题称做属性名。例如表 1.1 展示了银行系统中的客户表，它被称为客户关系。客户代码、姓名、地址、电话称为客户关系的属性，(09001001，张明，上海市杨浦区翔殷路 ABC 号，1234567890)是客户关系的一个元组。表 1.2 描述了账户信息表，表 1.3 描述的存款表是账户与客户之间的联系。

表 1.1　客户表(customer)

客户代码	姓 名	地　址	电　话
09001001	张明	上海市杨浦区翔殷路 ABC 号	1234567890
09001002	李亮	北京市海淀区上地 XWS 号	921034566
09001003	季新芳	杭州师范大学外国语学院	9653211311
09001005	王小刚	山东省济南市济微路 KUZ 号	45456546546

表 1.2　账户表

账户代码	余额
a-101	700
a-102	900
a-103	1000
a-104	2300

表 1.3　存款表

客户代码	账户代码
09001001	a-104
09001002	a-104
09001003	a-101
09001003	a-102

从这 3 个表中可以看出，张明和李亮共用一个账户，可能是合作关系；而客户季新芳开了两个账户，账户余额分别为 700 和 900。

关系模型是当前商用数据库的主导数据模型，有着坚实的理论支持，建立在集合论、数理逻辑、关系理论等严格的数学基础上。此外，关系数据模型结构简单，符合人的思维方式，大多数数据库系统都是基于这种关系模型的。

1.4.3　对象模型

对象数据模型是数据库系统中一种新型的逻辑数据模型，是数据库技术和面向对象设计思想相结合的产物。传统的数据模型主要用于存储数值和文本等由若干数据项构成的数据，因此这些数据以记录为基本形式就能满足需要。随着数据类型越来越复杂，以及图形、图像、声音、视频等多媒体数据类型的应用普及，基本的记录形式已经无法满足应用需要，因此出现了对象数据模型。

对象数据模型表达信息的基本单位是对象，对象不仅描述实体的状态特征，而且还包含对象的行为特征。比如描述学生实体的对象，不仅包含学号、姓名、年龄、性别、专业等表示学生状态的属性特征，而且还包含显示学生姓名、入学、注册、修改专业等行为特征。

对象具有封装性、继承性和多态性。这些特性都是传统数据模型所不具备的，也是对象数据模型的本质特征。封装性是指将对象的属性和行为隐藏在一起，一个对象只能发送消息给另一个对象，另一个对象收到消息后做出响应，执行封装在内部的方法，把操作结

果返回给该对象。继承性是指某一类型的对象可以继承另一种或一些类型对象的部分或全部特征，这里的特征包括对象的属性和方法。图 1.3 展示了对象的继承结构。

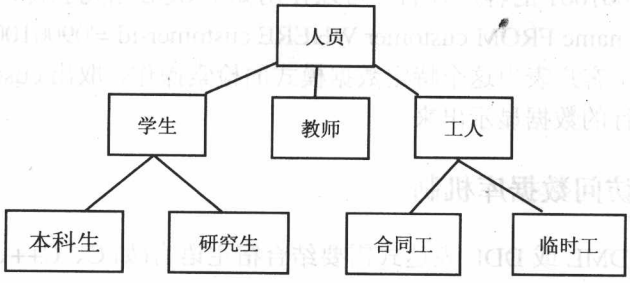

图 1.3　对象继承结构

图 1.3 所展示的继承关系中，最上面的人员是父对象模型，下端的均为子对象模型。学生对象和教师对象都继承人员对象的姓名、性别、年龄等特征，同时它们可以增加自己的新的特征，例如在学生对象中增加年级、奖学金等新特征；而在教师对象中增加职称、工龄等特征。继承性提高了软件的复用性，能够加快软件更新周期和缩短研发时间。

对象数据模型符合人类认识世界的规律，但是目前还不很成熟完善，对象数据库实现比较复杂，效率还有待提高，还需要不断发展。

1.5　数据库语言

数据库语言包括数据库系统的数据描述、操纵和控制语言。它是数据库管理系统为用户提供的定义、操纵、控制、维护数据库的工具，是数据库管理系统的重要组成部分。它主要包括两大部分：

(1) 数据定义语言，用来定义数据库的模式；

(2) 数据操纵语言，用来表达数据库的查询和更新操作。这两种类型的语言在形式上不是分离的语言，而是统一结构的数据库语言，例如目前应用最广的 SQL 语言。

1.5.1　数据定义语言

数据定义语言(DDL，Data Definition Language)是建立数据库模式的工具，数据库模式是由一系列 DDL 表达式建立的。例如，用下面的 SQL 语言定义一个保存学生信息的表：

```
CREATE TABLE student(id char(10)，name char (20)，age integer)；
```

数据库中底层复杂的存储结构和访问方式都被隐藏起来，用户不用关心这些数据模式实现的细节，而只需要掌握简单的 DDL 语句就能完成数据库模式的建立任务。同时 DDL 语言提供了数据库存储的一致性约束，例如学生年龄不能小于 10 岁。

1.5.2　数据操纵语言

数据操纵语言(DML，Data Manipulation Language)方便用户访问或操作某种特定数据模式的数据。DML 数据操纵语言命令使用户能够查询数据库以及操作已有数据库中的数据。

常见的操作有查询操作(对数据库信息进行检索)、插入操作(向数据库插入数据)、删除操作(从数据库删除数据)、更新操作(修改数据库的已有数据)。例如从数据库的客户表 customer 中找出客户号是 09001001 的客户姓名，可以使用如下 SQL 语句获得：

SELECET name FROM customer WHERE customer-id ='09001001';

这个查询指定了客户表中这个特定数据模式的检索操作，取出 customer-id 为 09001001 的行，并且把这些行的数据显示出来。

1.5.3　应用程序访问数据库机制

应用程序中的 DML 或 DDL 表达式需要结合宿主语言(如 C、C++、Java 等)来执行，目前常用两种结合方式。

第一种方式是通过提供应用程序接口(过程集)来将 DML 和 DDL 的表达式发送给数据库，执行完毕后再取回结果。开放数据库互联标准(ODBC，Open Database Connectivity)是微软公司开放服务结构(WOSA，Windows Open Services Architecture)中有关数据库的一个组成部分，它建立了一组规范，并提供了一组对数据库访问的标准 API(应用程序编程接口)。这些 API 利用 SQL 来完成其大部分任务。该标准可以支持 C、C++ 等高级程序设计语言来完成对数据库系统的访问，其可以支持目前主流的数据库系统，如 SQL Server、Oracle、Infomix、MySQL 等等。Java 数据库连接(JDBC，Java Data Base Connectivity)是一种用于执行 SQL 语句的 Java API，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。

第二种方式是通过扩展宿主语言的语义，使得嵌入其中的 DML 或 DDL 可以运行在宿主语言的程序中。通常用一个特殊字符作为 DML 调用的开始，并且通过预处理将 DML 表达式转化为宿主语言中的过程调用；也可以构建全新的可视化开发环境来支持数据库应用系统的开发，这种方式的典型代表有微软的 ADO.NET 开发环境和 Oracle 的 DEVELOPER 2000 系列等。

1.6　数据库系统体系结构

1.6.1　数据库的分层结构

三级分层结构的组织形式是一个典型的数据库的组织结构，如图 1.4 所示。这个结构是于 1975 年在美国 ANSI/X3/SPARC(美国国家标准协会的计算机与信息处理委员会中的标准计划与需求委员会)数据库小组的报告中提出的。

1．三级数据视图

数据抽象的 3 个级别又称为三级数据视图，是不同层次用户(人员)从不同角度所看到的数据组织形式。

(1) 外部视图。这是第一层的数据组织形式，是应用程序员开发应用程序时所使用的数据组织形式。外部视图是应用程序员所看到的数据的逻辑结构，是用户数据视图。外部视图可有