



广东省高等学校名牌专业教材
管理科学与工程类专业应用型本科系列规划教材

SQL Server 数据库轻松实务

——基于SQL Server 2005、SQL Server 2016

SQL Server Shujuku Qingsong Shiwu —— Jiyu SQL Server 2005、SQL Server 2016

● 赵良辉 肖建华 / 编著



华南理工大学出版社
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS



广东省高等学校名牌专业教材

管理科学与工程类专业应用型本科系列规划教材

SQL Server 数据库轻松实务

——基于SQL Server 2005、SQL Server 2016

SQL Server Shujuku Qingsong Shiwu——Jiyu SQL Server 2005、SQL Server 2016

● 赵良辉 肖建华 / 编著



华南理工大学出版社

SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

· 广州 ·

内容简介

本书以数据库入门者为主体读者,以 SQL Server 2005 和 SQL Server 2016 为基本操作界面,讲述了数据库的基本原理和 SQL Server 数据库的基本操作方法。对于数据库学习的入门者来说,本书阐述的数据库原理和基本操作适用于 SQL Server 2005 版之后的几乎所有版本,也为所有与数据库管理系统有关的操作准备了 SQL Server 2005 和 SQL Server 2016 两个版本的图例(在两者相同的情况下则只有一种图例),以利于学习者更好地了解 and 掌握数据库技术。

本书适合作为大学本科、专科及高职院校等数据库课程教材,也非常适合初学者学习。就专业而言,适合非计算机专业的以数据库应用、管理为重点的学生学习。计算机专业学生同样可以将本教材作为入门的辅助资料。

图书在版编目(CIP)数据

SQL Server 数据库轻松实务:基于 SQL Server 2005、SQL Server 2016/赵良辉,肖建华编著. —广州:华南理工大学出版社,2017.6

管理科学与工程类专业应用型本科系列规划教材

ISBN 978-7-5623-5269-3

I. ①S… II. ①赵… ②肖… III. ①关系数据库系统-教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2017)第 093126 号

SQL Server 数据库轻松实务——基于 SQL Server 2005、SQL Server 2016

赵良辉 肖建华 编著

出版人:卢家明

出版发行:华南理工大学出版社

(广州五山华南理工大学 17 号楼 邮编:510640)

<http://www.scutpress.com.cn> Email:scutc13@scut.edu.cn

营销部电话:020-87113487 87111048(传真)

策划编辑:潘宜玲 胡元

责任编辑:谢茉莉

印刷者:广州市怡升印刷有限公司

开本:787mm×1092mm 1/16 印张:12.5 字数:312 千

版次:2017 年 6 月第 1 版 2017 年 6 月第 1 次印刷

定价:32.00 元

版权所有 盗版必究 印装差错 负责调换

前言

数据库技术是通过研究数据库的结构、存储、设计、管理以及应用的基本理论和实现方法,并利用这些理论来实现对数据库中的数据进行处理、分析和理解的技术。

数据库技术发展到今天已经非常成熟,但由此也带来了一个弊端:数据库技术不再受到重视,在软件研发、信息系统管理或电子商务工作中常被置于可有可无的地位。很多企业或公司信息管理部门对于数据库的设计和维护过于随意,由此导致系统的数据管理工作在低水平上运行,甚至经常出错,造成经济损失。

出现上述问题的原因在于系统的维护和使用用户未能真正理解数据库技术的基本原理,将其视为与办公表格软件或数据统计软件类似的数据处理工具,未能从管理者的角度出发,将数据库与真实数据环境结合起来,实现数据库与对应业务系统的一一对应。

本教材以实用为特点,跳出常见的数据库教材理论框架,讲解数据库设计的基本思想和数据库操作的基本技术,力求成为帮助数据库学习者快速上手、快速理解数据库的有用工具。

本书的编者自2008年以来一直讲授“数据库原理与应用”课程,在教学经验日渐丰富的同时,通过与学生的交流,深刻把握初学者在学习过程中的所思所想,了解初学者学习中易犯的错误、通常难理解的章节,逐渐形成了一整套深入浅出、易学易用的数据库教学方案。这套方案摒弃了传统数据库教材理论完备、巨细无遗但难看难懂的缺点,以应用为核心,恰当引入基础理论,在辅导学生轻松入门的同时保证了理论框架的完备,使学习者既能轻松入门,又为后期知识体系升级打下坚实的基础。

本教材的特色:

- 基于编者丰富的教学经验,对数据库学习中的常见错误进行了分析,在帮助初学者入门上有独特的优势。

- 偏重实践,不牵涉较深的原理,加入了较多实操讲解和案例。

本教材的1~4章为数据库设计篇,5~13章为数据库操作篇,可依据需要学习。

因作者水平有限,书中难免存在错误,敬请读者批评、指正。

作者

2017年3月

第 1 章 绪论	1
1.1 为什么要学数据库	1
1.2 桌面数据库与网络数据库	4
1.3 数据库的历史	5
1.4 数据库系统基本概念	6
1.5 数据库的三级模式	7
第 2 章 概念模型与数据模型	9
2.1 用数据反映现实世界的过程	9
2.2 概念模型及其设计	10
2.3 数据模型及其设计	17
2.4 概念模型与数据模型的思考	19
第 3 章 关系数据库基本理论	20
3.1 关系数据库概述	20
3.2 关系的基本定义及术语	20
3.3 关系运算	21
3.4 关系的完整性	26
3.5 关系的规范化	28
第 4 章 数据库设计	35
4.1 概述	35
4.2 数据库设计方法	36
4.3 数据库设计的基本步骤	36
4.4 需求分析	37
4.5 概念结构设计	37
4.6 逻辑结构设计	39
4.7 物理结构设计	42
4.8 数据库实施、运行、维护	42
4.9 设计实例	42
第 5 章 SQL Server 2005/2016 安装和操作	49
5.1 SQL Server 2005 安装	49
5.2 SQL Server 2016 安装	56

第 6 章 T-SQL 语言	62
6.1 SQL 语言简介	62
6.2 基本语法	63
6.3 流程控制语句	72
6.4 函数	76
第 7 章 数据库和表	84
7.1 数据库的基本概念	84
7.2 创建、修改、删除数据库 (SQL 方式)	85
7.3 创建、修改、删除数据库 (图形界面操作)	87
7.4 创建、修改、删除表 (SQL 方式)	89
7.5 创建、修改、删除表 (图形界面操作)	95
第 8 章 数据库查询	102
8.1 查询环境准备	102
8.2 单表查询	103
8.3 连接查询	110
8.4 嵌套查询	113
8.5 基于派生表的查询	115
8.6 集合查询	116
8.7 查询语句的设计思路	118
第 9 章 视图、存储过程与触发器	120
9.1 视图	120
9.2 存储过程	122
9.3 触发器	126
第 10 章 数据库的恢复和数据转移	131
10.1 数据库备份与还原	131
10.2 数据库分离与附加	135
10.3 数据库的导入与导出	138
10.4 使用脚本完成数据库的复制	146
第 11 章 数据库安全管理	151
11.1 SQL Server 的身份验证模式	151

11.2	SQL Server 安全架构	152
11.3	权限的分配	157
第 12 章	Java 程序连接数据库	165
12.1	获取 JDBC	165
12.2	配置 SQL Server 网络环境	166
12.3	Java 应用程序连接数据库	167
第 13 章	实验	171
实验 1:	熟悉 SQL Server	171
实验 2:	流程语句编程练习	172
实验 3:	自定义函数	173
实验 4:	数据库与表的创建及操作	174
实验 5:	数据库查询 1	177
实验 6:	数据库查询 2	180
实验 7:	视图	181
参考文献		183
附录	数据库“学生成绩管理系统”完整脚本	184

第 1 章 绪 论

1.1 为什么要学数据库

学习数据库需要解决的最初始问题就是“为什么要学数据库?”

这个问题可以转化为另一个问题,即“数据库有什么用?”毫无疑问,数据库是用来记录和管理数据的,在这方面人类已经有非常多的工具,最开始是人类自己的头脑,然后是记录用的纸笔,再然后是计算机文本文件,接着是电子表格之类的表格文档。

到了表格文档这个程度,数据的记录已经是非常方便了,那么为什么一定要用数据库来代替电子表格,作为一种更先进的数据记录方式呢?下面将用几个实例来说明数据库在哪些方面比一般的电子表格(如 Excel)更有用。

1.1.1 数据访问的难题

设想你是某大学教务处的一个职员,某天教务处长交给你一个数据查询的任务:

查找本校所有 1992 年出生、至少 1 门功课不及格而且本学期修了至少 5 门课的学生。

将上述任务分解一下,可分成 3 个小任务:

- ①找出所有 1992 年出生的学生。
- ②从中找出他们的所有成绩,统计不及格课程数;保留所有不及格课程数非零的学生。
- ③从中找出本学期选课记录,统计选课门数。

假设学校有学生 8 000 人,则第①步需要查询 8 000 人次出生年份;第②步需要查询每位学生的成绩(假设);第③步需要查询每位学生的选课记录。假设平均每位学生有 10 门功课成绩,且 1992 年出生的学生占学生总数的 1/4,且每位学生单学期平均选课 8 门,那么总共需要查询的次数为:

$$8\,000 + 8\,000/4 \times (10 + 8) = 44\,000$$

如果单次查询(通过手工翻找学生名册)需要 1 秒钟的时间,则上述工作需要 12 小时左右,也就是一天半的工作时间;如果使用 Excel 查询,且查询者精通电子表格的使用,一般也需要不少于 1 小时来完成,而且不能保证准确性(从概率的角度而言肯定会出错)。

那么借助数据库管理完成上述工作需要多久呢?只要掌握了数据库查询技术,一般 1~5 分钟即可完成,只需输入下述几行 SQL 代码^①:

① 为便于和正文文字区分,本书所有 SQL 代码都采用小写,但并不是不允许大写形式的代码。秉承 Windows 系统不区分大小写的习惯,SQL 语言亦不区分大小写,因此读者可任意使用不同大小写形式进行编码。



```
select 学号 from 学生表 where year(出生日期) =1992
Intersect
select distinct 学号 from 成绩表 where 分数 <60
Intersect
select 学号 from 选课表 group by 学号 having count(*) >=5
```

— 第一行
— 第二行
— 第三行
— 第四行
— 第五行

其中，第一行的代码表示“找出所有出生日期在 1992 年的学生的学号”，第三行的代码表示“找出所有分数小于 60 分的学生的学号并去除重复学号”，第五行表示“找出所有已选课数目大于等于 5 的学生的学号”，而第二行和第四行表示“求它们的交集”。因此最后的查询结果就是上述任务所要的答案。

由此可见，特定数据的访问上，使用数据库具有压倒性的优势。

1.1.2 数据完整性问题

假设某位同学 Tom 考上了 A 大学，入学报到后在学生名册上留下了一行信息，见表 1.1 第二行。

表 1.1 学生名册

学号	姓名	性别	出生日期	备注
1011	Mary	女	1998-2-1	
1012	Tom	男	1999-9-9	
1013	Jack	男	1997-5-23	

军训结束后 Tom 选择了两门课程打算学习，于是在“选课表”上留下了一行信息，见表 1.2 第二行。

表 1.2 选课表

学号	姓名	课程编号	课程名称
1011	Mary	A01	高等数学
1011	Mary	A02	大学英语
1012	Tom	A01	高等数学
1012	Tom	A05	数据库原理
1013	Jack	A02	大学英语

但是世事无常，Tom 因为对高考失利耿耿于怀，加上对所选专业也不满意，打算休学重新参加高考。校方劝解无效之后将他的信息从学生名册中删除。至于 Tom 在学校教学系统中留下的其他信息，一来教务信息内容庞大而教学管理人手有限，二来不牵涉经济问题，因此清理得并不完善，其中也包括选课表 1.2。

这样一来，正式上课之时，教师发现 Tom 未出席，查询他的详细信息时却发现现在学生名册中根本没有这个学生。于是，一个“不存在”的学生选修了一门“数据库原理”课

程,难道学校中存在某个选课的鬼魂吗?

造成这种错误的当然不是鬼魂,但是,如果采用纸张或者电子表格来进行上述信息管理,这样的错误是会经常发生的。错误的原因就在于表和表之间的信息联系很难通过人的记忆,或表格软件有限的功能来维持。

使用数据库可以轻松解决这一问题。数据库可以保护数据的“参照完整性”,对于选课表而言,其中的学生信息(学号、姓名)作为外键存在,也就是其数据必须来自学生名册。当需要删除学生名册中某条记录时,选课表中的对应记录会自动执行事先设定好的操作,例如自动删除选课表中整条记录(称为“级联删除”),或者自动将该选课记录中的“学号”“姓名”两列的内容设置为空,具体如何操作由数据库设计者规定。也就是说,不管怎样,数据库中都不会出现一个“不存在”的学生选课事件,数据的一致性得以保证。

1.1.3 数据操作的原子性问题

让我们考虑一个银行的转账过程:你需要从你自己的A账户转账500万元到客户的B账户,则银行的操作过程为:

步骤1:读取两账户初始状态: $A=1\,000$ 万元, $B=0$ 元;

步骤2:将A的内容减去500万元, $A=500$ 万元, $B=0$ 元;

步骤3:将B的内容加上500万元, $A=500$ 万元, $B=500$ 万元。

如果银行不是使用数据库,而是使用电子表格来进行账户管理,那么有可能出现这样的状况:在刚刚完成步骤2的时候断电了,经过手忙脚乱的抢修电路之后,再次打开A账户,发现金额变成了500万元,而B账户依然为0,你的500万元凭空消失了。

当然,这样的情况一般不可能发生;但是如果没有特殊的数据操作设计,出现隐患的可能将一直存在,而普通电子表格没有相应的解决方法。

现在来看看数据库如何解决上述问题:数据库将上述从步骤1到步骤3的整个过程定义为一个“原子”,即不可再分的操作集合,所有步骤或者全部被执行,或者全部未被执行,不允许处于某一中间状态;这样的原子称为一个“事务”。有了“事务”的概念,那么不管是否断电,客户的账户或者处于事务执行前的一致性状态(步骤1执行前),或者处于事务执行后的一致性状态(步骤3执行后),都是可以接受的状态。

数据库使用事务日志和其他措施来保证数据操作的原子性问题。

1.1.4 数据冗余和不一致以及安全性问题

为了保证重要数据的安全,我们一般习惯将数据备份到其他地方,这就造成了数据的冗余,由此也带来另一个“数据不一致”的问题:当前数据更新时,备份的数据有没有更新?如果没有同步更新,很可能在很久之后需要恢复数据时,无从判别哪一处的数据才是真正有效的。

因此,数据冗余和由此导致的数据不一致是数据保存的大问题。数据库系统有一套完整的方案保证数据冗余最小,而且数据备份与当前数据之间绝不会无法分辨(基于差异备份和自动时间记录等措施)。

另外,数据的访问权限也是电子表格难以解决的问题。很多企业发工资的时候给职工的工资信息是“工资条”,即将纸质工资表剪成一条条,每条对应一个职工的工资,这样



做的目的是为了防止职工看到别人的工资。如果使用数据库的权限管理，只需为每个职工设定访问权限，则不但可限制职工只能看到自己的工资，还能进行更细致的数据访问权限设计。

通过上述实例我们可以看到，数据库确实在数据管理方面具有其他数据记录方式无可比拟的优势，只要对数据管理有要求的企业、单位或个人都应该使用数据库管理系统。一些常见的数据库使用情形如下：

- 银行：存储客户信息、账户、贷款记录。
- 航空业：订票和航班信息处理。
- 学校：学生信息储存、课程注册、成绩记录等。
- 电信业：通话记录、电话卡余额查询。
- 证券公司：股票、债券的持有、出售、买入信息。
- 制造业：供应链管理、人力资源管理。
- 公安局：身份证信息、通缉信息、指纹系统。
- 网站：用户名、密码、访问时长和次数、个人信息管理。
- 商店：货物名称、存量、价格、供应商信息、销售信息、税收信息等的管理。

1.2 桌面数据库与网络数据库

一般只用于单机的、很少联网使用的数据库产品被称为桌面数据库系统，最常见的有 Access、Visual FoxPro 等，Excel 有时也被归类为桌面数据库。桌面数据库的主要特点如下：

- (1) 广泛用于单机环境；
- (2) 用于一般桌面型操作系统如 WinXP、Win10 等；
- (3) 不提供或提供有限的网络应用功能；
- (4) 没有或提供弱化的安全方案；
- (5) 主要面向日常办公需要。

这样的数据库产品侧重于个人数据管理，强调可操作性、易开发和简单管理，一般不能用于企业数据库环境。

以 SQL Server 为代表的网络数据库系统与传统意义上的桌面数据库相比有以下优点：

- (1) 企业级的网络数据库一般需要网络操作系统支持，如 Windows Server、Linux Server 等（在一般桌面操作系统上也可以安装运行，但管理海量数据时影响效率）；
- (2) 拥有专用的数据库系统管理工具，与软件的前端开发工具相互独立；
- (3) 具有强大的网络功能和分布式管理功能；
- (4) 支持先进的大规模数据库技术、海量用户并行查询、多线程服务器等；
- (5) 提供完备、复杂的数据安全性方案，具有强大的数据恢复能力。

由此可见桌面数据库与网络数据库是两种明显不同的产品，前者一般被归入办公自动化系统。本教材以网络数据库为学习的核心。

1.3 数据库的历史

数据库技术是伴随计算机的产生而产生的。世界上第一台计算机产生于 1946 年，名字叫“ENIAC”，它是一台体积庞大、耗电量大的真空管计算机。数据处理技术随着计算机和软件的发展同步发展，具体可分为三个阶段：

(1) 人工管理阶段（20 世纪 50 年代中期以前）：计算机主要用于计算工作，数据与程序代码共存在纸带、卡片、磁带等存储介质中。

(2) 文件系统阶段（20 世纪 50 年代后期至 60 年代中期）：数据可以作为单独的文件储存在操作系统中，可以长期保存并通过文件系统管理，但仍有数据共享性差、冗余度大和数据独立性差的缺点。

(3) 数据库系统阶段（20 世纪 60 年代后期至今）：出现了统一管理数据的数据库管理系统，标志着数据管理技术的飞跃。

目前主流的商用数据库管理软件如图 1.1 所示，其中 ORACLE 占据了差不多一半的市场份额，SQL Server 约占五分之一（2008 年 Gartner 数据）。另外，MySQL 是一款非常流行的开放源代码数据库系统，它的优势是结构简单、完全免费（图 1.2）。

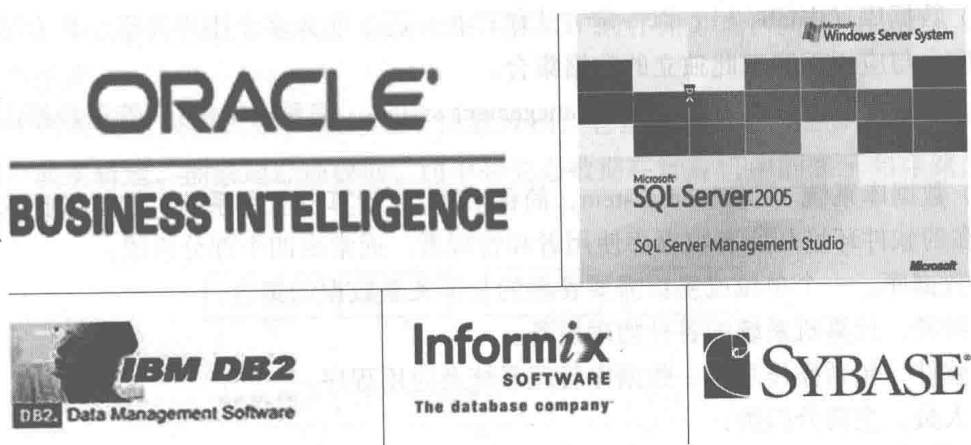


图 1.1 主流商用数据库管理软件



图 1.2 主流免费数据库产品

那么，应该选择哪个产品作为我们学习数据库的工具呢？参考如下：

- 国企、事业单位，中小型企业信息管理，建议精通 Windows/Office/C#/Microsoft SQL Server，因为这类工作岗位上会经常做小软件的快速开发及数据快速处理。



- 百度、阿里巴巴这类互联网企业，建议精通 MySQL，因为这类企业不愿意花钱购买正版软件，同时又需要对源代码进行定制，因此 MySQL 最适合这类企业。

- 专门从事大型软件项目开发，以及电信、电商、金融等资金充裕且对数据安全最重视的企业，适合使用 ORACLE。

- 对于初学数据库的学生来说，建议学习 Microsoft SQL Server (2005, 2008, 2014, 2016 等版本)，因为微软平台从 Windows 操作系统、VS 开发工具到 C# 语言编程等，无论是安装、使用还是学习都很方便，并且书籍也很多。使用这个平台，能让初学者将注意力集中在核心内容的学习上，避免很多无关因素的打扰。例如 ORACLE 是用命令行来控制且主要在 Linux 下使用，而多数初学者根本不了解 Linux 系统。

1.4 数据库系统基本概念

与数据库相关的具体概念包括以下几个方面：

(1) 数据 (data)：数据库中储存的基本对象，是我们通过观察、实验或计算得出的 (数值) 结果。数据有很多种，最简单的就是数字，也可以是文字、图像、声音等。数据可以用于科学研究、设计、查证等。

(2) 数据库 (database)：以一定方式储存在一起、能为多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合。

(3) 数据库管理系统 (database management system, 简称 DBMS)：管理数据库的计算机软件。

(4) 数据库系统 (database system, 简称 DBS)：计算机应用系统，包含数据库、数据库所存在的软件环境和所有数据库使用者和管理者。通常由四个部分组成：

- 数据库。一个单位或组织需要管理的全部关系数据的集合。
- 硬件。计算机系统的各种物理设备。
- 软件。包括操作系统、数据库管理系统及应用程序。
- 人员。主要分四类：

第一类为系统分析员和数据库设计人员。系统分析员负责应用系统的需求分析和规范说明，他们和用户及数据库管理员一起确定系统的硬件配置，并参与数据库系统的概要设计。数据库设计人员负责数据库中数据的确定、数据库各级模式的设计。

第二类为应用程序员，负责编写使用数据库的应用程序。这些应用程序可对数据进行检索、建立、删除或修改。

第三类为最终用户，他们利用系统的接口或查询语言访问数据库。

第四类用户是数据库管理员 (database administrator, DBA)，负责数据库的总体信息控制。DBA 的职责包括：具体数据库中的信息内容和结构，决定数据库的存储结构和存取策略，定义数据库的安全性要求和完整性约束条件，监控数据库的使用和运行，负责数据库的性能改进、数据库的重组和重构，以提高系统的性能。

1.5 数据库的三级模式

所谓“模式”可理解为“式样”“布局”“逻辑结构”。人们为数据库设计了一个严谨的体系结构，数据库领域公认的标准结构是三级模式结构，它包括外模式、概念模式、内模式。三级模式结构有效地组织、管理数据，提高了数据库的逻辑独立性和物理独立性。

模式：数据库的整体逻辑结构，可以简单理解为数据库包含哪些数据表、表间的关系、表内的结构等，以及在基本表的基础上设立的安全性、完整性工具。模式是数据库中全体数据的逻辑结构和特征的描述，是现实世界某应用环境（企业或单位）的所有信息内容集合的表示，是所有用户的公共数据视图。模式的其他称呼有概念模式、逻辑模式、全局模式。

外模式：数据库用户看到并允许使用的、对局部数据的逻辑结构和特征的描述，是数据库用户的视图，一般与具体的应用程序相关。由于数据库的用户有多种，因此外模式也有多个，外模式也可称为子模式、应用模式、用户模式、局部模式。

内模式：指数据库的物理结构、数据库在计算机中的存储方式，又称为内模式、存储模式、物理模式。

数据库系统的三级模式是数据的三个抽象级别，它把数据的具体组织留给 DBMS 管理，使用户能逻辑地、抽象地处理数据，而不必关心数据在计算机中的表示和存储。三种模式之间的关系见图 1.3。

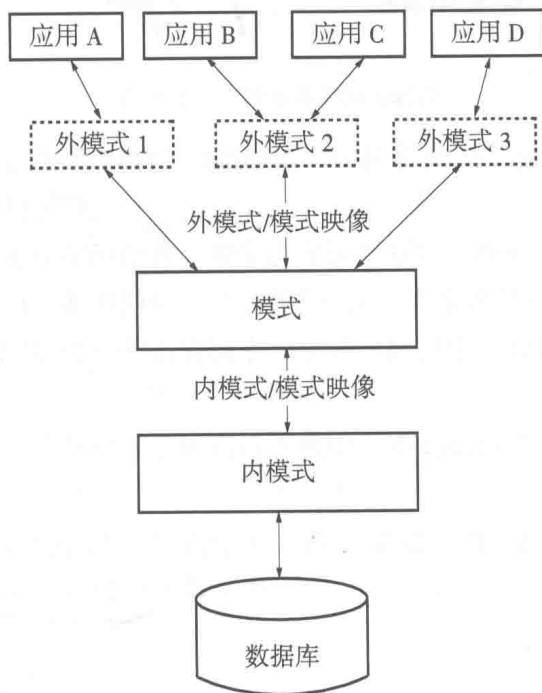


图 1.3 数据库的三级模式



为了实现这三个层次上的联系和转换，数据库系统在这三级模式中提供了两层映像：外模式/模式的映像和模式/内模式的映像。

通过数据库的二级映像可保证数据库与应用程序之间的独立性，具体可分为逻辑独立性和物理独立性。

逻辑独立性：当数据库的整体逻辑结构发生变化时，通过调整外模式和模式之间的映像，使得外模式中的局部数据及其结构不变，程序不用修改。

物理独立性：当数据库的存储结构发生变化时，通过调整模式和内模式之间的映像，使得整体模式不变，于是外模式及应用程序也不用改变。

上述独立性的保证一般都由数据库管理系统来完成（少数情境也需要人工操作）。当需要修改模式，例如增加或修改属性时，只需对外模式/模式映像进行修改，而不用对外模式进行修改，从而保证了基于外模式的应用程序可以照常使用，保证了数据的逻辑独立性。当存储结构（存储设备或存储方式）改变时，只需改变模式/内模式映像，而不用改变模式，这样即使服务器的物理存储设备不断更新，数据的逻辑结构仍然保持稳定，保证了数据的物理独立性。

【习题】

1. 为什么要学习数据库？
2. 桌面数据库和网络数据库的区别是什么？它们分别适合什么职业的人使用？
3. 数据库的逻辑独立性和物理独立性都是为了什么目的？既然逻辑独立性已经可以保证使用数据库的程序不用修改了，那么物理独立性是不是可有可无呢？

第2章 概念模型与数据模型

数据库是各个部门、企业应用所涉及的数据的集合，它不仅反映数据本身所表达的内容，而且还反映数据之间的联系。计算机不能直接处理现实世界中的具体事物，必须事先将具体事物转换成计算机能够处理的数据。

在数据库系统的形式化结构中如何抽象表示、处理现实世界中的信息和数据呢？答案就是数据模型。

2.1 用数据反映现实世界的过程

数据是为了记录、反映现实世界而存在的，其实现途径就是通过人的思维加工，将从现实世界中接收到的信息变成头脑中的概念，再把这个概念实现到计算机中来（图 2.1）。

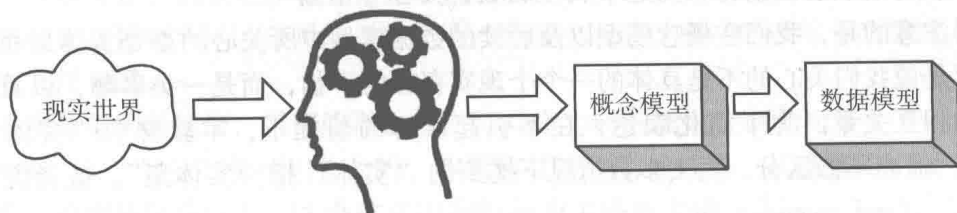


图 2.1 三种世界的模型转换

由此产生了“信息的三个世界”的说法，即现实世界、存在于头脑中的信息世界（概念）、计算机中的数据世界。

（1）现实世界：现实存在的世界。现实世界包罗万象，我们只需要观察我们感兴趣的那一部分。我们所关注的一类事物称为“实体”，这一类事物共有的属性称为特征。

（2）信息世界：现实世界在我们头脑中的反映。换言之，就是我们所关注的事物在我们头脑中形成的概念。

（3）计算机世界：把信息世界表现到计算机中而得到的世界，是信息世界在计算机中的具体表达。

现实世界在信息世界的表达，我们称为“概念模型”，它是独立于计算机世界的；概念模型在计算机世界中的表达称为“数据模型”。

2.2 概念模型及其设计

2.2.1 基本概念

概念模型是对现实世界的抽象和概括。概念模型涉及的概念有实体和实体集、联系和联系集、属性和码。

1. 实体和实体集

实体是现实世界中可与其他事物相区别的事物或“物体”“事件”。例如一个学生、一栋大楼、一个公司、一个节日等等。显然，实体可以是实实在在的物质，也可以是抽象的概念。

我们需要通过对实体属性的描述来描述实体，例如描述某个学生，我们说他的姓名是什么、学号是多少、有多高（身高属性）、有多重（体重属性），无一不是实体的属性，通过这些属性的集合我们才能获得一个综合的“学生”的概念，因此我们把具有相同属性的（即同类的）实体的集合称为实体集。属于同一实体集的所有实体具有共同的属性，但是每个实体在属性上的取值是不一样的。例如对于所有的学生，他们有共同的学号、姓名、性别、出生日期等属性，但是每个实体在这些属性上的取值不同。所有实体在某个属性上的所有可能取值的集合称为这个属性的域，又称“值集”。

值得注意的是，我们在概念模型以及后续的数据模型中所关心的都是实体集而不是实体，也就是说我们关心的不是具体的一个个现实存在的事物，而是一类事物，以及不同类事物间的相互关系。为了简化表达，在不引起误解的前提下，本教材中“实体集”和“实体”的概念不做区分，后文多数情况下提到的“实体”指“实体集”，或者说“一类实体”。

2. 联系和联系集

联系是多个实体间的相互关联，联系集则是同类型联系的集合。和前面的实体与实体集类似，我们所关心的实际上是联系集，因此后文也通常将其简称为“联系”。

多数情况下我们所关注的联系是不同类型实体之间的联系，例如学生和课程之间的联系，可命名为“选修”或“学习”。也存在同类型实体之间的联系，但出现不多。

联系一般被命名为某个动词，这是它直观上和实体的最大区别。依据联系所关联的实体的类别可将其分为二元联系（表达两类实体之间的相互关联）和多元联系（表达三类或更多类实体之间的关联）。前者如“选修”联系，管理了课程和学生两类实体；后者如“排课”联系，关联了“教师”“教室”“班级”等多类实体。

将二元联系所关联的两类实体集用 A 和 B 代表，则 A、B 在数量上经常有如下三种对应关系（称为“映射基数”）。

(1) 一对一（记为 1:1）。A 中的一个实体（最多）只与 B 中的一个实体相联系，B 中的一个实体也（最多）只与 A 中的一个实体相联系。例如公司和总经理、学校和校长。

(2) 一对多（记为 1:n）。A 中的一个实体可以同 B 中的多个实体相联系，但 B 中的一个实体最多同 A 中的一个实体相联系。例如班级和学生、家庭和成员、学生与班长，这