

普.通.高.等.学.校
计算机教育“十二五”规划教材

SQL Server 2008 数据库应用教程 (第2版)

*SQL SERVER 2008
DATABASE APPLICATION
(2nd edition)*

邱李华 李晓黎 任华 冉兆春 ◆ 编著



人民邮电出版社
POSTS & TELECOM PRESS

普通高等院校
计算机教育“十二五”规划教材

SQL Server 2008 数据库应用教程

(第2版)

*SQL SERVER 2008
DATABASE APPLICATION
(2nd edition)*

邱李华 李晓黎 任华 冉兆春 ◆ 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

SQL Server 2008数据库应用教程 / 邱李华等编著
— 2版. — 北京: 人民邮电出版社, 2012. 8
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-28493-8

I. ①S… II. ①邱… III. ①关系数据库—数据库管
理系统—高等学校—教材 IV. ①TP311.138

中国版本图书馆CIP数据核字(2012)第126706号

内 容 提 要

本书以介绍 SQL Server 2008 数据库管理系统为主, 同时介绍一定的数据库基础知识和数据库应用程序开发等方面的知识。全书共分 14 章, 内容包括: SQL Server 2008 数据库系统简介、服务器与客户端配置、Transact-SQL 基础、数据库管理、表和视图管理、存储过程和触发器管理、游标管理、维护数据库、SQL Server 安全管理、SQL Server 代理服务以及使用 Visual C# 程序设计和开发数据库应用程序。

本书由多年讲授数据库原理及应用、SQL Server 数据库管理系统的教师及多年从事 SQL Server 数据库应用系统开发的工程师联合编写。全书包含大量示例, 每章后都附有大量的理论练习题和上机练习题, 练习题内容重点突出, 且知识点覆盖全面。

本书还为教师配备了电子教案, 每章后的理论练习题和上机练习题参考答案, 以及第 14 章介绍的 Visual C# 数据库应用程序的源程序, 为教师开展教学提供了方便。

本书可作为大学本科有关课程教材, 也可供大专、高职使用, 或作为广大 SQL Server 2008 数据库管理员和开发人员的参考资料。

普通高等学校计算机教育“十二五”规划教材

SQL Server 2008 数据库应用教程 (第 2 版)

-
- ◆ 编 著 邱李华 李晓黎 任 华 冉兆春
责任编辑 邹文波
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 22.25 2012 年 8 月第 2 版
字数: 583 千字 2012 年 8 月河北第 1 次印刷

ISBN 978-7-115-28493-8

定价: 42.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

第 2 版前言

数据库技术是数据管理的最新技术,是计算机科学中发展最快的领域之一。随着网络技术的迅猛发展,以及数据库技术与网络技术的紧密结合,数据库技术已经广泛应用于各种领域,小到工资管理、人事管理,大到企业信息的管理等,数据库技术已经成为计算机系统应用最广的技术之一。同时,数据库技术及其应用也成为国内外高等学校计算机专业和许多非计算机专业的必修或选修内容。

SQL Server 2008 是当今深受欢迎的关系数据库管理系统,是一个杰出的数据库平台,它建立在成熟而强大的关系数据模型的基础之上,可以很好地支持客户机/服务器模式,可用于大型联机事务处理、数据仓库以及电子商务等,能够满足各种类型的企事业单位构建网络数据库的要求,是目前各类学校学习大型数据库管理系统的首选对象。

本教材以介绍 SQL Server 2008 关系数据库管理系统为主,全书内容可以分为三大部分。第一部分(第 1 章)为数据库基础知识部分,比较全面而简洁地介绍数据库系统的基本概念和基本原理,使得没有学过数据库基础知识的读者可以快速了解数据库的基本概念和基本原理,为系统学习后面的 SQL Server 2008 关系数据库管理系统打下基础。第二部分介绍 SQL Server 2008 关系数据库管理系统的主要功能及其使用,由第 2 章到第 12 章组成,包括 SQL Server 2008 数据库系统简介、服务器与客户端配置、Transact-SQL 基础、数据库管理、表和视图管理、存储过程和触发器管理、游标管理、维护数据库、SQL Server 安全管理、SQL Server 代理服务。第三部分(第 13、14 章)介绍如何使用 Visual C# 开发基于 SQL Server 2008 数据库的应用程序。这一部分内容介绍了如何将数据库设计与应用程序设计结合起来,开发出完整的数据库应用程序。

本教材在内容的选择、深度的把握上力求做到深入浅出、循序渐进。教材结合了大量的示例,为概念的理解提供了捷径。为适应多媒体教学的需要,我们为使用本教材的教师制作了配套的电子教案,并为教师提供各章后的练习题参考答案,以及最后一章的数据库应用实例的数据库和源程序。另外,我们还为需要使用 ASP.NET 技术开发数据库应用程序的教师和其他读者提供了相关的参考内容及实例,如有需要,可以通过人民邮电出版社的网站 <http://www.ptpedu.com.cn> 下载。

由于作者水平有限,书中难免存在错误或不足之处,敬请读者批评指正。

编 者

2012 年 7 月

目 录

第 1 章 数据库基础	1	3.1.2 注册服务器.....	42
1.1 数据库系统基本概念.....	1	3.1.3 启动、暂停和停止 SQL Server.....	43
1.2 数据管理技术的发展.....	2	3.1.4 服务器的连接与断开.....	45
1.3 数据模型.....	4	3.1.5 配置服务器属性.....	45
1.3.1 现实世界.....	5	3.1.6 配置服务器端网络连接.....	47
1.3.2 信息世界.....	5	3.1.7 服务器配置选项.....	49
1.3.3 机器世界.....	7	3.1.8 配置链接服务器.....	51
1.4 关系数据库.....	7	3.2 配置客户端.....	55
1.4.1 关系模型.....	7	3.2.1 配置客户端网络.....	55
1.4.2 关系数据库的规范化理论.....	10	3.2.2 配置 ODBC 数据源.....	55
1.5 数据库系统的体系结构.....	16	练习题.....	57
1.5.1 数据库系统的三级模式结构.....	16	第 4 章 Transact-SQL 语言基础	60
1.5.2 数据库的二级映像与数据的 独立性.....	17	4.1 Transact-SQL 简介.....	60
1.6 数据库系统设计简介.....	18	4.2 Transact-SQL 的语法规则.....	61
练习题.....	21	4.3 标识符.....	62
第 2 章 SQL Server 简介	26	4.4 数据类型.....	63
2.1 概述.....	26	4.4.1 整型数据类型.....	63
2.1.1 SQL Server 的发展.....	26	4.4.2 定点数据类型.....	63
2.1.2 SQL Server 的客户机/服务器 体系结构.....	26	4.4.3 浮点数据类型.....	63
2.2 SQL Server 2008 的安装.....	28	4.4.4 字符数据类型.....	64
2.2.1 SQL Server 2008 的版本.....	28	4.4.5 日期和时间数据类型.....	65
2.2.2 SQL Server 2008 的系统要求.....	29	4.4.6 图形数据类型.....	65
2.2.3 SQL Server 2008 的安装.....	30	4.4.7 货币数据类型.....	65
2.3 SQL Server 2008 的管理工具简介.....	33	4.4.8 位数据类型.....	66
2.4 SQL Server 的数据库体系结构.....	37	4.4.9 二进制数据类型.....	66
2.5 SQL Server 2008 的系统数据库简介.....	38	4.4.10 其他数据类型.....	66
2.6 SQL Server 2008 的系统表简介.....	39	4.5 常量.....	67
练习题.....	39	4.6 变量.....	68
第 3 章 服务器与客户端配置	41	4.6.1 局部变量.....	68
3.1 配置服务器.....	41	4.6.2 全局变量.....	69
3.1.1 创建服务器组.....	41	4.7 运算符与表达式.....	70
		4.7.1 算术运算符.....	70
		4.7.2 字符串串联运算符.....	70
		4.7.3 赋值运算符.....	70
		4.7.4 比较运算符.....	70

4.7.5 逻辑运算符.....	71	5.4.2 使用 DROP DATABASE 语句 删除数据库.....	97
4.7.6 位运算符.....	72	5.5 分离数据库和附加数据库.....	97
4.7.7 一元运算符.....	72	5.5.1 分离数据库.....	97
4.7.8 运算符的优先次序.....	73	5.5.2 附加数据库.....	99
4.8 函数.....	73	5.6 收缩数据库.....	101
4.8.1 数学函数.....	74	5.6.1 查看数据库磁盘使用情况.....	101
4.8.2 字符串函数.....	75	5.6.2 使用图形界面工具收缩数据库.....	102
4.8.3 日期和时间函数.....	76	5.6.3 使用 DBCC SHRINKDATABASE 语句收缩数据库.....	102
4.8.4 转换函数.....	77	5.6.4 使用 DBCC SHRINKFILE 语句 收缩指定的数据库文件.....	103
4.8.5 聚合函数.....	79	5.6.5 设置自动收缩数据库选项.....	104
4.9 流程控制语句.....	79	5.7 移动数据库.....	104
4.9.1 BEGIN...END 语句.....	79	练习题.....	105
4.9.2 IF...ELSE 语句.....	80	第 6 章 表和视图	108
4.9.3 CASE 函数.....	81	6.1 表.....	108
4.9.4 WHILE 循环.....	82	6.1.1 表的概念.....	108
4.9.5 GOTO 语句.....	82	6.1.2 创建表.....	109
4.9.6 WAITFOR 语句.....	83	6.1.3 查看和管理表中的数据.....	111
4.9.7 RETURN 语句.....	83	6.1.4 查看表的磁盘空间信息.....	112
4.10 注释.....	83	6.1.5 修改表.....	113
4.11 Transact-SQL 语句的解析、 编译和执行.....	84	6.1.6 删除表.....	115
练习题.....	85	6.2 表约束.....	115
第 5 章 数据库管理	87	6.2.1 表约束的类型.....	116
5.1 数据库的存储结构.....	87	6.2.2 管理主键约束.....	116
5.1.1 数据库文件.....	87	6.2.3 管理唯一性约束.....	118
5.1.2 数据库文件组.....	88	6.2.4 管理检查约束.....	119
5.2 创建数据库.....	89	6.2.5 管理默认约束.....	121
5.2.1 使用 SQL Server Management Studio 的图形界面工具创建数据库.....	89	6.2.6 外部键约束.....	121
5.2.2 使用 CREATE DATABASE 语句 创建数据库.....	90	6.3 表的更新.....	123
5.3 修改数据库.....	92	6.3.1 插入数据.....	123
5.3.1 使用数据库属性对话框修改 数据库.....	92	6.3.2 删除数据.....	125
5.3.2 使用 ALTER DATABASE 语句 修改数据库.....	94	6.3.3 修改数据.....	127
5.4 删除数据库.....	96	6.4 表的查询.....	128
5.4.1 使用图形界面工具删除数据库.....	97	6.4.1 使用图形界面工具查询数据.....	129
		6.4.2 SELECT 语句.....	129
		6.4.3 最基本的 SELECT 语句.....	130
		6.4.4 设置查询条件.....	132

6.4.5 对查询结果排序.....	135	8.1.5 存储过程的返回值.....	171
6.4.6 使用聚合函数.....	136	8.1.6 获取存储过程信息.....	172
6.4.7 对查询结果分组.....	136	8.1.7 修改和重命名存储过程.....	173
6.4.8 指定组或聚合的搜索条件.....	137	8.1.8 删除存储过程.....	174
6.4.9 生成汇总行.....	138	8.1.9 系统存储过程.....	174
6.4.10 连接查询.....	138	8.2 用户定义函数.....	176
6.4.11 子查询.....	141	8.2.1 什么是用户定义函数.....	176
6.4.12 合并查询.....	141	8.2.2 创建标量值函数.....	177
6.4.13 保存查询结果.....	142	8.2.3 创建内连表值函数.....	177
6.5 视图管理.....	143	8.2.4 创建多语句表值函数.....	179
6.5.1 视图概述.....	143	8.2.5 修改和重命名用户定义函数.....	180
6.5.2 创建视图.....	143	8.2.6 删除用户定义函数.....	181
6.5.3 修改视图.....	146	8.3 触发器.....	181
6.5.4 删除视图.....	147	8.3.1 触发器的基本概念.....	181
练习题.....	147	8.3.2 deleted 表和 inserted 表.....	182
第 7 章 规则和索引	152	8.3.3 事务的概念及应用.....	182
7.1 规则.....	152	8.3.4 创建触发器.....	185
7.1.1 规则的概念.....	152	8.3.5 修改触发器.....	188
7.1.2 创建规则.....	152	8.3.6 删除触发器.....	189
7.1.3 查看规则.....	153	8.3.7 禁用和启用触发器.....	190
7.1.4 绑定规则.....	153	练习题.....	190
7.1.5 解除绑定规则.....	154	第 9 章 游标	193
7.1.6 删除规则.....	154	9.1 游标概述.....	193
7.2 索引.....	155	9.1.1 游标的概念.....	193
7.2.1 设计索引.....	155	9.1.2 游标的分类.....	194
7.2.2 创建索引.....	156	9.2 游标的使用.....	195
7.2.3 修改索引.....	158	9.2.1 声明游标.....	195
7.2.4 删除索引.....	159	9.2.2 打开游标.....	196
7.2.5 查看索引信息.....	159	9.2.3 读取游标数据.....	197
7.2.6 使用索引优化数据库查询效率.....	163	9.2.4 关闭游标.....	199
7.2.7 无法使用索引的 SELECT 语句.....	164	9.2.5 获取游标的状态和属性.....	200
练习题.....	165	9.2.6 修改游标结果集中的行.....	206
第 8 章 存储过程、函数和		9.2.7 删除游标结果集中的行.....	207
触发器	167	9.2.8 删除游标.....	207
8.1 存储过程.....	167	练习题.....	208
8.1.1 什么是存储过程.....	167	第 10 章 维护数据库	210
8.1.2 创建存储过程.....	168	10.1 导入和导出数据.....	210
8.1.3 执行不带参数的存储过程.....	169	10.1.1 将表中数据导出到文本文件.....	210
8.1.4 带参数的存储过程.....	170		

10.1.2 将表中数据导出到 Access 数据库	212	11.4.1 角色管理	245
10.1.3 从文本文件向 SQL Server 数据库导入数据	213	11.4.2 管理角色中的用户	248
10.1.4 从 Access 数据库中导入数据	215	11.5 权限管理	249
10.2 备份数据库	216	11.5.1 权限的种类	249
10.2.1 数据库备份方式	216	11.5.2 设置权限	250
10.2.2 SQL Server 2008 的备份和恢复数据库模式	218	练习题	256
10.2.3 创建和删除备份设备	219	第 12 章 SQL Server 代理服务	260
10.2.4 使用图形界面工具对数据库进行备份	221	12.1 配置 SQL Server 代理服务	260
10.2.5 使用 BACKUP DATABASE 语句备份数据库	222	12.2 作业管理	262
10.3 恢复数据库	224	12.2.1 创建作业	262
10.3.1 使用使用图形界面工具恢复数据库	225	12.2.2 启动、停止和禁用作业	265
10.3.2 使用 RESTORE DATABASE 语句恢复数据库	227	12.2.3 修改和删除作业	265
10.3.3 使用 RESTORE LOG 语句恢复事务日志	228	12.3 警报管理	265
10.4 数据库快照	229	12.3.1 创建警报	265
10.4.1 数据库快照的概念	229	12.3.2 修改和删除警报	266
10.4.2 创建数据库快照	230	12.4 操作员管理	267
10.4.3 查看数据库快照	231	12.4.1 创建操作员	267
10.4.4 恢复到数据库快照	231	12.4.2 修改和删除操作员	269
10.4.5 删除数据库快照	232	练习题	271
练习题	232	第 13 章 Visual C# 程序设计基础	273
第 11 章 SQL Server 安全管理	235	13.1 C# 语言基础	273
11.1 安全管理概述	235	13.1.1 C# 语言的基本特点	273
11.2 登录	235	13.1.2 .NET Framework 和 C#	274
11.2.1 身份验证模式	236	13.1.3 使用 Visual Studio 2008	275
11.2.2 创建登录名	237	13.1.4 编写一个简单的 C# 例子	277
11.2.3 修改和删除登录名	240	13.2 数据类型	280
11.3 数据库用户	242	13.2.1 值类型	280
11.3.1 数据库用户概述	242	13.2.2 引用类型	283
11.3.2 新建数据库用户	243	13.2.3 类型转换	286
11.3.3 修改和删除数据库用户	244	13.3 常量和变量	287
11.4 角色	245	13.3.1 常量	287
		13.3.2 变量	287
		13.4 运算符和表达式	288
		13.4.1 算术运算符和算术表达式	288
		13.4.2 逻辑运算符和逻辑表达式	289
		13.4.3 递增递减运算符和递增递减表达式	290

13.4.4 关系运算符和关系表达式	290	14.1.1 ADO.NET 的结构	309
13.4.5 赋值运算符和赋值表达式	291	14.1.2 ADO.NET 的命名空间	310
13.5 流程控制语句	292	14.2 ADO.NET 中的常用 SQL Server	
13.5.1 选择控制语句	292	访问类	311
13.5.2 循环控制语句	295	14.2.1 SqlConnection 类	311
13.5.3 跳转控制语句	297	14.2.2 SqlCommand 类	313
13.5.4 异常处理语句	298	14.2.3 SqlDataReader 类	314
13.6 类和对象	300	14.2.4 DataSet 类	315
13.6.1 面向对象程序设计思想	300	14.2.5 SqlDataAdapter 类	318
13.6.2 创建类	301	14.2.6 DataView 类	319
13.6.3 函数	304	14.3 人力资源管理系统 (C/S 版)	320
练习题	306	14.3.1 数据库设计	320
第 14 章 Visual C#数据库程序		14.3.2 应用程序的设计	323
设计	309	练习题	346
14.1 ADO.NET 的结构和命名空间	309		

第 1 章

数据库基础

随着科学技术和社会经济的飞速发展，人们掌握的信息量急剧增加，要充分地开发和利用这些信息资源，就必须有一种新技术能对大量的信息进行识别、存储、处理与传播。随着计算机软硬件技术的发展，20 世纪 60 年代末，数据库技术应运而生，并从 70 年代起得到了迅速的发展和广泛的应用。数据库技术主要研究如何科学地组织和存储数据，如何高效地获取和处理数据。数据库技术作为数据管理的最新技术，目前已广泛应用于各个领域。对于一个国家来说，数据库的建设规模、数据库信息量的大小和使用频度已经成为衡量这个国家信息化程度的重要标志。

1.1 数据库系统基本概念

1. 信息

信息是人脑对现实世界中的客观事物以及事物之间联系的抽象反映，它向我们提供了关于现实世界实际存在的事物及其联系的有用知识。

2. 数据

数据是人们用各种物理符号，把信息按一定格式记载下来的有意义的符号组合。广义地讲，数据并不仅仅是数字，它还可以是文字、图像、声音等各种表现形式，这些形式的数据经过数字化后都可以存入计算机中。数据是数据库中存储的基本对象。

数据和它的语义是不可分割的。例如，对于数据：

（李明，78）

我们可以赋予它一定的语义，它表示李明的期末考试平均成绩为 78 分。如果不了解其语义，则无法对其进行正确解释，甚至解释为李明的年龄为 78。

3. 数据处理

数据处理是指对各种形式的数据进行收集、整理、加工、存储和传播的一系列活动的总和。其目的之一是从大量的原始数据中提取出对人们有价值的信息，作为行动和决策的依据；目的之二是为了借助计算机科学地保存和管理大量的复杂的数据，以便人们能方便地充分利用这些信息资源。

4. 数据库

数据库（DataBase，DB），简单地讲就是存放数据的仓库。不过，数据库不是数据的简单堆积，而是以一定的方式保存在计算机存储设备上的相互关联的数据的集合。也就是说，数据库中的数据并不是相互孤立的，数据和数据之间是有关联的。

5. 数据库管理系统

数据库管理系统 (Database Management System, DBMS) 是一种系统软件, 介于应用程序和操作系统之间, 用于帮助我们管理输入到计算机中的大量数据。例如, 用于创建数据库, 向数据库中存储数据, 修改数据库中的数据, 从数据库中提取信息等。具体来说, 一个数据库管理系统应具备如下功能。

(1) 数据定义功能: 可以定义数据库的结构, 定义数据库中数据之间的联系, 定义对数据库中数据的各种约束等。

(2) 数据操纵功能: 可以实现对数据库中数据的添加、删除、修改, 可以对数据库进行备份、恢复等。

(3) 数据查询功能: 可以以各种方式提供灵活的查询功能, 使用户可以方便地使用数据库中的数据。

(4) 数据控制功能: 可以完成对数据库中数据的安全性控制、完整性控制、多用户环境下的并发控制等多方面的控制。

(5) 数据库通信功能: 在分布式数据库或提供网络操作功能的数据库中还必须提供数据库的通信功能。

数据库管理系统在计算机系统中的地位可以用图 1-1 来表示。它运行在一定的硬件和操作系统平台上。人们可以使用一定的开发工具, 利用 DBMS 提供的功能, 创建满足实际需求的数据库应用系统。

根据对信息的组织方式的不同, 数据库管理系统又可以分为关系、网状和层次 3 种类型。目前使用最多的数据库管理系统是关系型数据库管理系统 (RDBMS), 如 SQL Server、Oracle、Sybase、Visual FoxPro、DB2、Informix、Ingres 等都是目前常见的关系型数据库管理系统。

6. 数据库管理员

应当指出的是, 数据库的建立、使用和维护只靠 DBMS 是不够的, 还需要有专门的人员来完成, 这些人员称为数据库管理员。

7. 数据库系统

数据库系统 (DataBase System, DBS) 是指在计算机系统中引入数据库的系统, 除了相关的硬件之外, 数据库系统还包括数据库、数据库管理系统、应用系统、数据库管理员和用户。

可以看出, 数据库、数据库管理系统和数据库系统是 3 个不同的概念, 数据库强调的是数据, 数据库管理系统是系统软件, 而数据库系统强调的是系统。

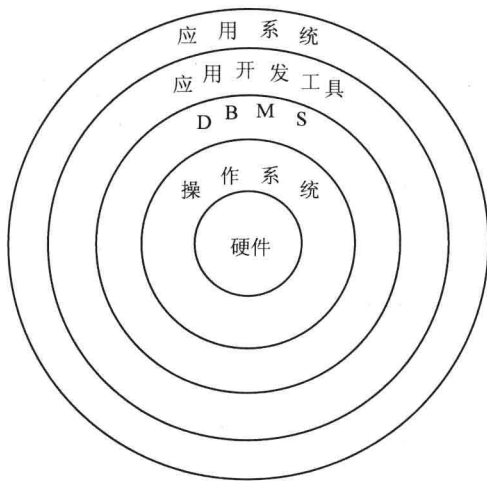


图 1-1 数据库管理系统在计算机系统中的地位

1.2 数据管理技术的发展

随着计算机硬件和软件技术的发展, 数据管理技术的发展大致经历了人工管理阶段、文件系统阶段和数据库系统阶段。

1. 人工管理阶段

在计算机发展的初级阶段,计算机硬件本身还不具备像磁盘这样的可直接存取的存储设备,因此也无法实现对大量数据的保存,也没有用来管理数据的相应软件,计算机主要用于科学计算。这个阶段的数据管理是以人工管理的方式进行的,人们还没有形成一套数据管理的完整的概念。其主要特点如下。

(1) 数据不保存。因为计算机主要用于科学计算,一般只是在需要进行某个计算或课题时才将数据输入,计算结果也不保存。

(2) 还没有文件的概念。数据由每个程序的程序员自行组织和安排。

(3) 一组数据对应一个程序。每组数据和一个应用程序相对应,即使两个应用程序要使用相同的一组数据,也必须各自定义和组织数据,数据无法共享。因此,可能导致大量的数据重复。

(4) 没有形成完整的数据管理的概念,更没有对数据进行管理的软件系统。这个时期的每个程序都要包括数据存取方法、输入/输出方法和数据组织方法,程序直接面向存储结构,因此存储结构的任何修改都将导致程序的修改。程序和数据不具有独立性。

人工管理阶段的特点可以用图 1-2 来描述。

2. 文件系统阶段

随着计算机软硬件技术的发展,如直接存储设备的产生,操作系统、高级语言及数据管理软件的出现,计算机不仅用于科学计算,也开始大量用于信息管理。数据可以以文件的形式长期独立地保存在磁盘上,且可以由多个程序反复使用;操作系统及高级语言或数据管理软件提供了对数据的存取和管理功能,这就是文件系统阶段。这个阶段的数据管理具有以下特点。

(1) 数据可以长期保存在磁盘上,因此可以重复使用。数据不再仅仅属于某个特定的程序,而可以由多个程序反复使用。

(2) 数据的物理结构和逻辑结构有了区别,但较简单。程序开始通过文件名和数据打交道,不必关心数据的物理存放位置,对数据的读/写方法由文件系统提供。

(3) 程序和数据之间有了一定的独立性。应用程序通过文件系统对数据文件中的数据进行存取和加工,程序员不必过多地考虑数据的物理存储细节,因此可以把更多的精力集中在算法的实现上。并且,数据在存储上的改变不一定反映在程序上,这可以大大节省维护程序的工作量。

(4) 出现了多种文件存储形式,因而,相应地有多种对文件的访问方式,但文件之间是独立的,它们之间的联系要通过程序去构造,文件的共享性也还比较差。数据的存取基本上以记录为单位。

文件系统阶段的特点及程序和数据之间的关系可以用图 1-3 来表示。

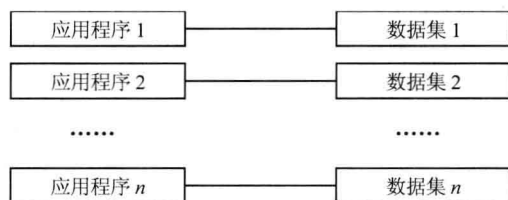


图 1-2 人工管理阶段

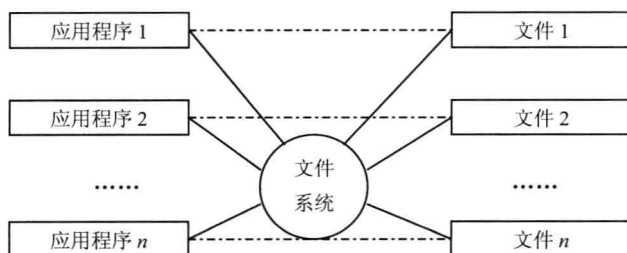


图 1-3 文件系统阶段

虽然文件系统比人工管理有了长足的进步,但是文件系统所能提供的数据存取方法和操作数据的手段还是非常的有限。例如,文件结构的设计仍然是基于特定的用途,基本上是一个数据文件对应于一个或几个应用程序;程序仍然是基于特定的物理结构和存取方法编制的。因此,数据的存储结构和程序之间的依赖关系并未根本改变;文件系统数据冗余大,同样的数据往往在不同的地方重复出现,浪费存储空间;数据的重复以及数据之间没有建立起相互联系还会造成数据的不一致性。

随着信息时代的到来,人们要处理的信息量急剧增加,对数据的处理要求也越来越复杂,文件系统的功能已经不能适应新的需求,而数据库技术也正是在这种需求的推动下逐步产生的。

3. 数据库系统阶段

数据库系统阶段使用数据库技术来管理数据。数据库技术自 20 世纪 60 年代后期产生以来就受到了广大用户的欢迎,并得到了广泛的应用。数据库技术发展至今已经是一门非常成熟的技术,它克服了文件系统的不足,并增强了许多新功能。在这一阶段,数据由数据库管理系统统一控制,数据不再面向某个应用而是面向整个系统,因此数据可以被多个用户、多个应用共享,概括起来具有以下主要特征。

(1) 数据库能够根据不同的需要按不同的方法组织数据,以最大限度地提高用户或应用程序访问数据的效率。

(2) 数据库不仅能够保存数据本身,还能保存数据之间的相互联系,保证了对数据修改的一致性。

(3) 在数据库中,相同的数据可以共享,从而降低了数据的冗余度。

(4) 数据具有较高的独立性,数据的组织和存储方法与应用程序相互独立,互不依赖,从而大大降低了应用程序的开发代价和维护代价。

(5) 提供了一整套的安全机制来保证数据的安全、可靠。

(6) 可以给数据库中的数据定义一些约束条件来保证数据的正确性(也称完整性)。

数据库系统阶段应用程序和数据库之间的关系可以用图 1-4 来表示。

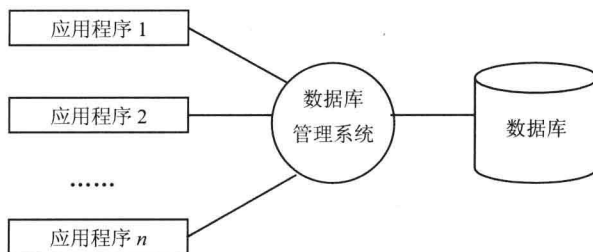


图 1-4 数据库系统阶段

1.3 数据模型

数据库是某个企业、组织或部门所涉及的数据的综合,它不仅要反映数据本身的内容,而且要反映数据之间的联系。由于计算机不可能直接处理现实世界中的具体事物,所以人们必须事先把具体事物转换成计算机能够处理的数据。在数据库技术中使用数据模型来抽象、表示现实世界中的数据和信息。

现实世界中的数据要进入到数据库中,需要经过人们的认识、理解、整理、规范和加工。可以把这一过程划分成3个主要阶段,即现实世界阶段、信息世界阶段和机器世界阶段。现实世界中的数据经过人们的认识和抽象,形成信息世界;在信息世界中用概念模型来描述数据及其联系,概念模型按用户的观点对数据和信息进行建模,不依赖于具体的机器,独立于具体的数据库管理系统,是对现实世界的第一层抽象;根据所使用的具体机器和数据库管理系统,需要对概念模型进行进一步转换,形成在具体机器环境下可以实现的数据模型。这3个阶段的相互关系可以用图1-5来表示。

数据在每一个阶段都有其相应的概念和术语。

1.3.1 现实世界

在现实世界阶段,我们把现实世界中客观存在并可以相互区分的事物称为实体。实体可以是实际存在的东西,也可以是抽象的。例如,学生、课程、零件、仓库、项目、案件等都是实体。

每一个实体都具有一定的特征。例如,对于“学生”实体,它具有学号、姓名、性别、出生日期等特征;对于“零件”实体,它具有名称、规格型号、生产日期、单价等特征。

具有相同特征的一类实体的集合构成了实体集。例如,所有的学生构成了“学生”实体集;所有的职员构成了“职员”实体集;所有的部门构成了“部门”实体集,等等。

在一个实体集中,用于区分实体的特征称为标识特征。例如,对于学生实体,学号可以作为其标识特征,因为通过不同的学号可以区分不同的学生实体,而性别则不能作为其标识特征,因为通过性别“男”并不能识别出具体是哪个学生。

1.3.2 信息世界

人们对现实世界的对象进行抽象,并对其进行命名、分类,在信息世界用概念模型来对其进行描述。信息世界涉及的主要概念如下。

- (1) 实体:对应于现实世界的实体。例如,一个学生、一门课程、一个仓库、一个零件等。
- (2) 属性:对应于实体的特征。一个实体可以由若干个属性来刻画。例如,学生实体可以有学号、姓名、性别、班级、年龄等属性;课程实体可以有课程编号、课程名称、学分等属性。
- (3) 码:对应于实体的标识特征,是唯一标识实体的属性。例如,学生实体可以用学号来唯一标识,因此学号可以作为学生实体的码。
- (4) 域:属性的取值范围称为该属性的域。例如,姓名的域为字符串集合;年龄的域为不小于零的整数;性别的域为(男,女)。
- (5) 实体型:具有相同属性的实体必然具有相同的特征和性质。用实体名及其属性名集合来描述实体,称为实体型。例如,学生实体型描述为

学生(学号,姓名,性别,年龄)

课程实体型可以描述为

课程(课程号,课程名,学分)

- (6) 实体集:同型实体的集合构成了实体集。例如,全体学生构成了学生实体集。

(7) 联系:现实世界中的事物之间通常都是有联系的,这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常指组成实体的各属性之间的联系;实体之间的

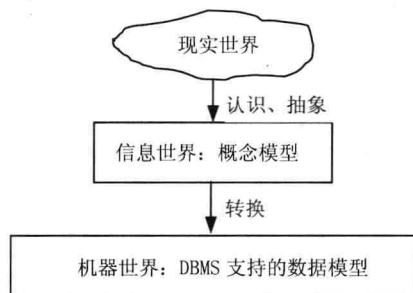


图 1-5 对现实世界数据的抽象过程

联系通常指不同实体集之间的联系。这些联系总的来说可以划分为一对一联系、一对多（或多对一）联系以及多对多联系。

● 一对一联系：如果实体集 A 与实体集 B 之间存在联系，并且对于实体集 A 中的任意一个实体，在实体集 B 中至多只有一个实体与之对应；而对于实体集 B 中的任意一个实体，在实体集 A 中也至多只有一个实体与之对应，则称实体集 A 和实体集 B 之间存在着一对一的联系（表示为 1:1）。

例如，“公司”是一种实体，“总经理”也是一种实体。如果按照语义，一个公司只能有一个总经理，而一个总经理只能管理某一个公司，则“公司”和“总经理”实体之间的联系就是一对一的联系。这种关系可以用图 1-6 来表示，这里把公司和总经理之间的关系称为“管理”关系。

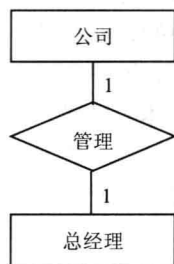


图 1-6 一对一联系

● 一对多联系：如果实体集 A 与实体集 B 之间存在联系，并且对于实体集 A 中的任意一个实体，在实体集 B 中可以有多个实体与之对应；而对于实体集 B 中的任意一个实体，在实体集 A 中至多只有一个实体与之对应，则称实体集 A 到实体集 B 的联系是一对多的联系（表示为 1:n）。

例如，“部门”是一种实体，“职工”也是一种实体。如果按照语义，一个部门可以有多个职工，而一个职工只能归属于一个部门，则“部门”实体和“职工”实体的联系就是一对多的联系，如图 1-7 所示。这里把部门和职工之间的关系称为“属于”关系。

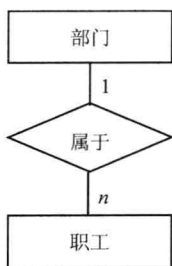


图 1-7 一对多联系

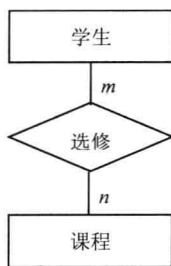


图 1-8 多对多联系

● 多对多联系：如果实体集 A 与实体集 B 之间存在联系，并且对于实体集 A 中的任意一个实体，在实体集 B 中可以有多个实体与之对应；而对于实体集 B 中的任意一个实体，在实体集 A 中也可以有多个实体与之对应，则称实体集 A 到实体集 B 的联系是多对多的联系（表示为 $m:n$ ）。

例如，“学生”是一种实体，“课程”也是一种实体。“学生”实体和“课程”实体的联系就是多对多的联系。因为一个学生可以学习多门课程，而一门课程又可以有多个学生来学习。它们之间的关系如图 1-8 所示。这里把课程和学生之间的关系称为“选修”关系。

两个以上的实体之间也存在一对一、一对多和多对多的联系。例如，“邮局”是一种实体，“邮票”是一种实体，“顾客”是一种实体，这 3 个实体之间存在着一一种关系，这里称之为“购买”。它表示，一个顾客可以在多个邮局中买多种邮票；一个邮局可以把多种邮票出售给多个顾客；一种邮票可以在多个邮局中被多个顾客所购买。这样的三元关系可以表示成如图 1-9 所示。

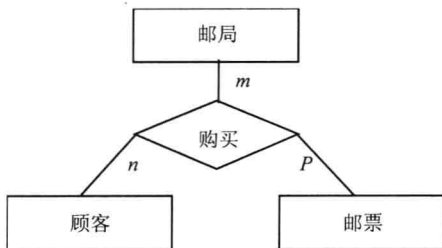


图 1-9 三个实体之间的多对多的联系

（8）概念模型：概念模型是对信息世界的建模，因此，概念模型应该能够方便、准确地表示出上述信息世界中的常用概念。概念模型有多种表示方法，其中，最常用的是“实体—联系方法”（Entity Relationship Approach），简称 E-R 方法。E-R 方法用 E-R 图来描述现实世界的概念模型，E-R 图提供了表示实体、属性和联系的方法，具体如下。

● 实体型：用矩形表示，在矩形内写明实体名。图 1-10 所示为学生实体和课程实体。

● 属性：用椭圆形表示，并用无向边将其与实体连接起来。例如，学生实体及其属性用 E-R 图表示如图 1-11 所示。



图 1-10 实体的表示

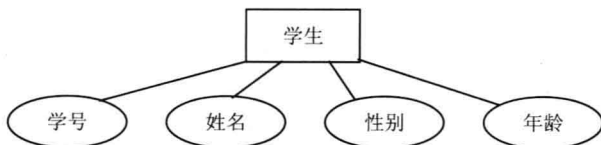


图 1-11 学生实体及其属性

● 联系: 用菱形表示, 在菱形框内写明联系的名称, 并用无向边将其与有关的实体连接起来, 同时, 在无向边旁标上联系的类型。例如, 前面的图 1-6、图 1-7 和图 1-8 分别表示了一对一、一对多和多对多的联系。需要注意的是, 联系本身也是一种实体型, 也可以有属性。如果一个联系具有属性, 则这些属性也要用无向边与该联系连接起来。例如, 图 1-12 表示了学生实体和课程实体之间的联系“选修”, 每个学生选修某一门课程会产生一个成绩, 因此, “选修”联系有一个属性“成绩”, 学生和课程实体之间是多对多的联系。

用 E-R 图表示的概念模型独立于具体的 DBMS 所支持的数据模型, 是各种数据模型的共同基础, 因此比数据模型更一般、更抽象、更接近现实世界。

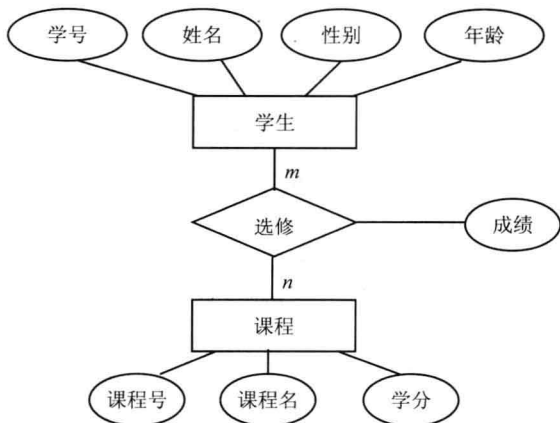


图 1-12 学生实体及课程实体之间的联系

1.3.3 机器世界

当信息进入计算机后, 则进入机器世界范畴。概念模型是独立于机器的, 需要转换成具体的 DBMS 所能识别的数据模型, 才能将数据和数据之间的联系保存到计算机上。在计算机中可以用不同的方法来表示数据与数据之间的联系, 把表示数据与数据之间的联系的方法称为数据模型。数据库领域常见的数据模型有以下 4 种:

- (1) 层次模型 (Hierarchical Model);
- (2) 网状模型 (Network Model);
- (3) 关系模型 (Relational Model);
- (4) 面向对象的模型 (Object Oriented Model)。

其中, 关系模型是目前使用最广泛的数据模型, 其他数据模型在 20 世纪 70 年代至 80 年代初比较流行, 现在已经逐步被关系模型的数据库系统所取代。因此, 本书将只讨论关系模型。需要注意的是, 以下介绍的关系模型是用户眼中看到的数据模型, 实际上关系模型是可以在某种 DBMS 的支持下、用某种语言进行描述的, 通过 DBMS 提供的功能实现对其进行存储和实施各种操作。我们把支持关系模型的数据库管理系统称为关系数据库管理系统, 简称 RDBMS。

1.4 关系数据库

1.4.1 关系模型

关系模型由关系数据结构、关系操作集合和关系完整性约束 3 部分组成。

1. 关系数据结构

在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。

例如，表 1-1 所示的学生信息表。

(1) 关系 (Relation): 一个关系对应于一张二维表，每个关系都有一个关系名。如表 1-1 所示的学生信息表可以取名为“学生信息”。

表 1-1 学生信息表

学 号	姓 名	性 别	年 龄
980010101	张涛	男	18
980010102	李明	男	18
980010103	刘心	女	19
.....			

(2) 元组 (Tuple): 表中的一行称为一个元组，对应于存储文件中的一个记录。

(3) 属性 (Attribute): 表中的一列称为一个属性，给每个属性起一个名字，称为属性名。属性对应于存储文件中的字段。

(4) 候选码 (Candidate key): 如果在一个关系中，存在多个属性 (或属性组合) 都能用来唯一标识该关系的元组，这些属性 (或属性组合) 都称为该关系的候选码 (或候选关键字)。例如，假设以上“学生信息”关系中的姓名没有重名现象，则学号和姓名都是候选码。

(5) 主码 (Primary key): 在一个关系的若干个候选码中指定作为码的属性 (或属性组合) 称为该关系的主码 (或主关键字)。例如，可以将以上“学生信息”关系的学号作为该关系的主码。

(6) 主属性 (Primary Attribute): 包含在候选码中的属性称为主属性。例如，学号和姓名 (假设无重名) 都是主属性。

(7) 非主属性 (Nonprimary Attribute): 不包含在任何候选码中的属性称为非码属性或非主属性。例如，性别和年龄都是非主属性。

(8) 关系模式 (Relation Schema): 对关系的描述称为关系模式，一般表示为

关系名 (属性 1, 属性 2, ..., 属性 n)

例如，以上的“学生信息”表是一个关系，其关系模式为

学生信息 (学号, 姓名, 性别, 年龄)

(9) 全码 (All-key): 如果一个关系模型的所有属性一起构成这个关系的码，则称其为全码。

(10) 域 (Domain): 属性的取值范围称为域。例如，性别的域是 (男、女)，年龄的域是大于 0 的整数。

(11) 分量 (Component): 元组中的一个属性值称为分量。例如，表 1-1 中的“李明”。

在关系模型中，实体和实体之间的联系都是用关系来表示的。例如，图 1-12 所表示的概念模型中的学生、课程和选修关系可以表示为以下 3 个关系模式：

学生信息 (学号, 姓名, 性别, 年龄)

课程 (课程号, 课程名, 学分)

选修 (学号, 课程号, 成绩)

2. 关系操作

关系操作主要包括查询、插入、修改和删除数据，这些操作的操作对象和操作结果都是关系，也就是元组的集合。