



新世纪高职高专实用规划教材

• 计算机系列

数据库原理与应用

——基础、设计、实现与程序开发 (SQL Server 版)

SHUJUKU YUANLI YU YINGYONG

邵鹏鸣 编著



清华大学出版社

新世纪高职高专实用规划教材 计算机系列

数据库原理与应用

——基础、设计、实现与程序开发

(SQL Server 版)

邵鹏鸣 编著

清华大学出版社

北 京

内 容 简 介

本书以一个完整的应用案例为主线,以 SQL Server 为工具,全面阐述了数据库技术原理及应用。全书分为 7 部分:第一部分讲解数据库功能及其定义;第二部分为数据库基础,主要内容为关系模型及其实现,包括 SQL 中的 DDL(数据定义语言);第三部分为数据库设计,主要内容为实体-关系模型、实体-关系模型到数据库设计的转换和表的规范化;第四部分为实现数据库,包括创建数据库和表,表的联系及参照完整性,SQL 语言和索引;第五部分为数据库对象,包括视图、存储过程、触发器和用户定义函数;第六部分为数据库管理,包括安全管理和数据库维护;第七部分为数据库应用程序开发,包括控件及 UGI 程序设计,使用 ADO.NET 进行数据库编程等内容。

本书内容丰富、可操作性强、语言生动流畅,没有晦涩的术语,采用面向实际应用的教学方式来安排内容,使学生能够轻松地愉快地掌握从事数据库领域的职业所必需的数据库技术。

本书可作为高职高专院校计算机专业和应用型高等院校计算机专业的教材和教学参考书,也适合所有希望学习数据库技术的人员使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

数据库原理与应用——基础、设计、实现与程序开发(SQL Server 版)/邵鹏鸣编著. —北京:清华大学出版社, 2007.3

(新世纪高职高专实用规划教材 计算机系列)

ISBN 978-7-302-14619-3

I. 数… II. 邵… III. 关系数据库—数据库管理系统, SQL Server—高等学校:技术学校—教材
IV. TP311.138

中国版本图书馆 CIP 数据核字(2007)第 014524 号

责任编辑:黄 飞

封面设计:杨玉兰

版式设计:北京东方人华科技有限公司

责任校对:李玉萍

责任印制:李红英

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175

投稿咨询:010-62772015

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮购热线:010-62786544

客户服务:010-62776969

印 刷 者:北京国马印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:18.25

字 数:441 千字

版 次:2007 年 3 月第 1 版

印 次:2007 年 3 月第 1 次印刷

印 数:1~5000

定 价:25.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。

联系电话:010-62770177 转 3103

产品编号:021857-01

前 言

数据库技术是计算机领域中最重要技术之一。目前,不仅在各种机型上都配有数据库管理系统,而且各行各业的信息系统,乃至 Internet 上的信息系统也大都离不开数据库的支持。因此数据库已成为信息社会的重要基础设施。

近年来,随着 Internet 技术激动人心的发展,使得数据库应用及开发成为最热门的职业技能之一。因为数据库应用及开发的专业人才的需求量非常大,所以学习和掌握数据库技术的技能和知识是非常必要的。比较遗憾的是,在数据库的教学实践中,现有的大多数教材要么只讲理论,要么只讲一个数据库管理系统的使用方法,或者前半部分讲理论,后半部分讲一个数据库管理系统的使用方法。所有这些都会造成理论与实践的脱节,达不到应有的学习效果。而且很多教材是以 Visual FoxPro 或 Access 为工具讲解数据库的应用,远远不能满足市场和行业的需要。

SQL Server 是目前最为流行的数据库管理系统之一,它是学生在将来的工作中最可能遇到的产品,而且学生学会了 SQL Server 之后,很容易弄懂 Access。相反,学会了 Access 却不一定或很难弄懂 SQL Server。因此,笔者认为,在讲授数据库原理与应用时,应该从头到尾采用 SQL Server 来讲解知识点,这样可起到事半功倍的效果,这才是真正的理论联系实际,走向真正的实用。再者,在进行数据库应用程序开发时,行业很少采用 VBA 与宏进行数据库应用程序开发,而目前很多教材在讲数据库应用程序开发时,都会贯穿讲解 VBA 与宏。笔者认为,这与实用相违背,与市场相脱离。面对日新月异的新技术、新方法,我们必须对现有计算机课程的设置和教学内容进行调整,以适应技术进步与市场变化的需要,使教授的知识是市场上最需要的知识。

本教材根据理论与实践并重的原则编写,采用面向实际应用的教学方式来安排内容,使学生轻松掌握从事数据库领域的职业所必需的数据库技术。本书主要特色如下:

(1) 从应用与实用的角度出发,阐述数据库应用设计与实现。

(2) 将理论与实践有机地融合。如在讲解数据库建模、关系模型和规范化时,不是抽象地讲解概念和原理,而是和应用实例融合在一起,用实例说明抽象的概念和原理。这样有助于学生对概念和知识点的理解和应用,同时也可激发学生的学习热情。

(3) 新体系、新思路。无论是内容体系、教学的模式及理念等都具有一定的新意。本书的内容及体系结构使学生感到它既具有易读性又增进知识和技能,因此具有很强的实用性。

(4) 以一个具有商业应用价值的应用实例为主线,通过任务驱动的方式展开数据库知识的讲解,在学习基本原理的同时进行实践,使学生在较轻松的气氛下掌握最重要的知识点和技能,使其成为名副其实的“理论家”和“实践家”。内容叙述深入浅出,注重培养学生的应用能力和职业技能。

(5) 面向实际应用技术,逐步展现主题鲜明的各章内容的讲解。本书将焦点集中于重

要的主题来展开充分论述，而不是肤浅地涉及许多问题，罗列许多概念和术语。

(6) 先进性及实用性并重。本书的内容吸收了最新的实用方法及技术，讲解了 UML 在数据库设计中的应用，内容与实用的新技术同步。

(7) 以技术较全面、应用很广的 SQL Server 作为工具，来阐述数据库技术，而不是只讲解某一特定的数据库产品，注重数据库的共同要素，做到举一反三，触类旁通。

本书可作为高职高专院校计算机专业和应用型高等院校计算机专业的教材或教学参考书，也适合所有希望学习数据库技术的人员使用。

由于编写本书的时间略显仓促，书中难免有错漏之处，恳请专家和读者批评指正。(pmshao@163.com)，本书的源代码可在 <http://www.pyp.edu.cn/shaopm> 站点下载。如果发现书中的内容或代码有不当之处，我们会在 <http://www.pyp.edu.cn/shaopm> 站点发布勘误信息。要想与本书的作者进行交流，可加入论坛 <http://www.pyp.edu.cn/shaopm/forum/database/default.aspx>。

编 者

目 录

第 1 章 数据库简介	1	3.2.4 字段(列)约束和表约束	22
1.1 数据库举例	1	3.3 习题	23
1.2 数据库管理系统的概念	2	第 4 章 实体-关系模型	24
1.3 应用程序与数据库管理系统之间 的关系	2	4.1 实体	24
1.4 数据库的数据结构	3	4.2 属性	24
1.5 数据库的定义	3	4.2.1 简单属性和复合属性	24
1.6 习题	4	4.2.2 单值属性和多值属性	24
第 2 章 使用关系(表)存储用户数据	5	4.2.3 派生属性	25
2.1 关系模型	5	4.2.4 属性的域	25
2.1.1 关系模型的概念	5	4.2.5 关键字	25
2.1.2 关键字	6	4.3 关系	26
2.1.3 域	7	4.3.1 关系的类型	26
2.2 列的数据类型	7	4.3.2 1 对 1 关系(1:1)	26
2.3 关系的实现	9	4.3.3 1 对多关系(1:*)	26
2.3.1 创建数据库	9	4.3.4 多对多关系(*:*)	26
2.3.2 创建表	9	4.4 实体-关系图(E-R 图)	27
2.3.3 修改表	10	4.4.1 实体类的图形化表示	27
2.3.4 操作表的数据	11	4.4.2 关系的图形化表示	27
2.4 习题	13	4.4.3 属性的图形化表示	28
第 3 章 字段的属性和约束	15	4.5 实例研究	30
3.1 字段的属性	15	4.5.1 标识实体类	30
3.1.1 默认值	15	4.5.2 标识实体类间的关系	30
3.1.2 精度和小数位	17	4.5.3 标识实体的属性	33
3.1.3 标识、标识种子和标识 递增量	17	4.6 习题	38
3.1.4 ROWGUIDCOL	18	第 5 章 实体-关系模型到数据库设计 的转换	39
3.1.5 空 (NULL) 值	18	5.1 将实体类转换为表	39
3.2 约束	19	5.2 表的规范化	41
3.2.1 PRIMARY KEY 约束	19	5.2.1 函数依赖和函数依赖传递	41
3.2.2 UNIQUE 约束	20	5.2.2 更新异常	42
3.2.3 CHECK 约束	21	5.2.3 规范化的本质	43
		5.2.4 第一范式	43

5.2.5 第二范式	43	8.1.6 汇总与分组查询	78
5.2.6 第三范式	44	8.1.7 查询中的表达式	79
5.3 表示 HAS-A 关系	45	8.2 子查询	81
5.3.1 表示 1 对 1 关系	45	8.2.1 子查询的概念	81
5.3.2 表示 1 对多关系	45	8.2.2 使用别名的子查询	82
5.3.3 表示多对多关系	46	8.2.3 使用 EXISTS 和 NOT EXISTS 的子查询	83
5.4 实例研究	47	8.2.4 用于替代表达式的子查询	84
5.4.1 将 E-R 模型映射为表	47	8.2.5 UPDATE、DELETE 语句中的 子查询	84
5.4.2 对多值属性的处理	50	8.3 联接查询	84
5.4.3 再论 1 对 1 关系	50	8.4 实例研究	89
5.5 习题	51	8.5 习题	92
第 6 章 创建数据库和表	52	第 9 章 索引	94
6.1 创建数据库	52	9.1 什么是索引	94
6.1.1 理解数据库	52	9.2 索引的分类	95
6.1.2 示例	53	9.2.1 聚集索引	95
6.2 实例研究	55	9.2.2 非聚集索引	96
6.3 习题	58	9.2.3 惟一索引	96
第 7 章 创建表的关系和参照完整性	59	9.3 创建与删除索引	96
7.1 创建表的关系	59	9.3.1 在现有表上创建索引	96
7.1.1 在创建表时创建外键约束	59	9.3.2 在创建表时创建索引	97
7.1.2 在已有的表中创建外键 约束	60	9.3.3 删除索引	98
7.2 创建参照完整性	60	9.4 使用索引	98
7.2.1 参照完整性的含义	60	9.4.1 使用聚集索引	98
7.2.2 创建级联删除规则	61	9.4.2 使用非聚集索引	99
7.2.3 创建级联更新规则	62	9.4.3 使用惟一索引	99
7.2.4 对 INSERT 和 UPDATE 语句忽略外键约束	62	9.5 实例研究	99
7.3 实例研究	63	9.6 习题	100
7.4 习题	66	第 10 章 视图	101
第 8 章 结构化查询语言 SQL	67	10.1 视图综述	101
8.1 对单表进行查询	67	10.1.1 视图的概念	101
8.1.1 使用 SELECT 进行投影	67	10.1.2 视图的作用	102
8.1.2 简单条件查询	68	10.2 创建视图	102
8.1.3 复杂条件查询	69	10.2.1 创建简单视图	103
8.1.4 创建计算字段的查询	76	10.2.2 具有计算字段的视图	103
8.1.5 排序	77	10.2.3 创建视图字段的别名	104
		10.2.4 加密视图	104

10.2.5 具有数据约束的视图	105	12.5.1 修改触发器	132
10.3 修改和重命名视图	105	12.5.2 重命名触发器	132
10.3.1 修改视图	105	12.5.3 删除触发器	133
10.3.2 重命名视图	106	12.6 实例研究	133
10.4 通过视图修改数据	106	12.6.1 触发器应用分析	133
10.5 视图和查询的比较	106	12.6.2 用户定义函数	135
10.6 实例研究	107	12.7 习题	138
10.7 习题	109		
第 11 章 存储过程	110	第 13 章 安全管理	139
11.1 存储过程概述	110	13.1 安全性示例	139
11.2 创建简单的存储过程	110	13.2 创建登录账户	140
11.3 创建带参数的存储过程	111	13.2.1 用 Windows 账户访问 数据库	140
11.3.1 创建使用参数的简单 存储过程	111	13.2.2 在 SQL Server 中创建 SQL Server 登录名	141
11.3.2 创建使用参数默认值的 存储过程	112	13.3 用户	141
11.3.3 使用包含通配符的参数 默认值创建存储过程	114	13.4 授予权限	143
11.3.4 创建使用 OUTPUT 参数 返回数据的存储过程	114	13.5 角色	146
11.4 在存储过程中调用其他存储 过程	115	13.5.1 创建角色	147
11.5 执行存储过程	116	13.5.2 给角色授权	147
11.6 实例研究存储过程分析	116	13.5.3 向数据库角色添加用户	148
11.7 习题	124	13.5.4 将成员添加到固定服务器 角色	150
第 12 章 触发器和用户定义函数	125	13.6 习题	150
12.1 触发器种类	125	第 14 章 数据库维护	151
12.1.1 AFTER 触发器	125	14.1 备份	151
12.1.2 INSTEAD OF 触发器	125	14.1.1 概述	151
12.2 创建触发器	125	14.1.2 创建数据库备份	152
12.2.1 创建 AFTER 触发器	126	14.1.3 差异数据库备份	153
12.2.2 查看触发器	126	14.1.4 事务日志备份	154
12.2.3 Inserted 和 Deleted 表	126	14.2 还原	155
12.2.4 AFTER 触发器应用示例	127	14.2.1 还原数据库备份	155
12.3 创建 INSTEAD OF 触发器	129	14.2.2 还原差异数据库备份	156
12.4 触发器与约束的比较	131	14.2.3 还原事务日志备份	157
12.5 修改和重命名触发器	132	14.3 附加和分离数据库	158
		14.3.1 分离数据库	158
		14.3.2 附加数据库	159
		14.4 习题	160

第 15 章 控件及 GUI 程序设计 161

15.1	Connection 和 Command 对象	161
15.1.1	Connection 对象	161
15.1.2	Command 对象	164
15.2	创建三层应用程序——雇员信息管理应用程序	166
15.2.1	创建项目和窗体	167
15.2.2	创建问题域类: Employee 类	167
15.2.3	创建用户界面	171
15.2.4	创建数据存取类: EmployeeDA 类	173
15.2.5	编写 GUI(图形用户界面)代码	184
15.3	PictureBox(图片框)控件	188
15.3.1	PictureBox 的常用属性	192
15.3.2	PictureBox 的常用事件	193
15.3.3	Image.FromFile 方法	193
15.4	组合框控件和文本框控件	193
15.4.1	ComboBox 的常用属性 和方法	200
15.4.2	列表框控件的常用事件	202
15.4.3	TextBox 的常用属性	203
15.4.4	TextBox 的常用事件	205
15.5	列表框	205

15.5.1	列表框控件的常用属性	220
15.5.2	列表框控件的常用方法	221
15.5.3	列表框控件的常用事件	223
15.6	习题	223

第 16 章 使用 ADO.NET 进行数据库

编程.....	225
16.1 数据表 DataTable	225
16.2 数据集和数据适配器	237
16.2.1 将数据集绑定到 DataGrid 控件	257
16.2.2 行状态与行版本	260
16.3 Windows 窗体中的数据绑定	261
16.3.1 简单绑定控件属性	262
16.3.2 使用 CurrencyManager	268
16.4 创建和使用数据视图	269
16.4.1 使用数据库视图查询 数据	269
16.4.2 数据视图的常用属性 及方法	274
16.5 DataRelation 对象	276
16.5.1 实例：产品类别信息管理	276
16.5.2 导航表间关系	280
16.6 习题	281
参考文献	282

第 1 章 数据库简介

数据库技术一直是热门技术之一，它被广泛应用于许多领域，从桌面上的数据库到大型相互关联组织的分布式数据库，数据库成为了越来越重要的商业资产。市场、销售、生产、操作、财会、管理和所有的商业规则都在各自的活动中利用数据库技术来提高生产率。近年来，数据库技术对于 Internet 应用的迅猛增长起到了重要的推动作用。毕竟 Internet 只是一个通信系统，它的真正价值是从数据库中读取或存入数据和信息。由于对数据库技术专业人才的需求量非常大，因此，使得学习和掌握数据库技术、知识和应用技能成为许多大学生追求的目标之一。

1.1 数据库举例

数据库的目的是帮助人们跟踪事物。经典数据库应用涉及对诸如订单、客户、雇员或其他商人感兴趣的内容的跟踪。现今，数据库技术已被应用到了更多领域，诸如用于 Internet 的数据库或用于公司内联网的数据库。下面列举几个例子。

1. 从超市购物

当你从当地超市购买货物时，收银员使用条形码阅读器来扫描每种货物。这其实就是链接一个使用条形码从产品数据库中查询该货物价格的应用程序，然后通过该程序计算这些库存货物的数量，并在收银机上显示价格。如果记录产品的数量低于指定的最低极限值，数据库系统可能会自动设置一个订单来获得更多的产品库存。

2. 使用信用卡购物

当你使用信用卡购物时，服务人员要检查你是否有足够的剩余金额可以购买该商品。这种检查可以用电话来进行，也可以用连接到计算机系统的磁卡阅读器自动完成。无论是哪种情况，都存在一个记录有使用信用卡进行购物的信息数据库，此数据库中包含了你使用信用卡进行购物的信息。为了检查你的信用卡，存在一个数据库应用程序，此程序使用你的信用卡号码来检查你想购买的商品价格，以及你这个月已经购买的商品总额是否在信用限度内。当购买被确认有效后，则这次的购买详细信息又被添加到了这个数据库中。在确认此次购买生效之前，这个应用程序也会访问数据库，检查该信用卡是否在被盗或丢失列表中。

3. 在旅行社预定假期

当你咨询某次假期的安排时，旅行社可能访问几个包含假期和飞机详细信息的数据库。当你预定假期时，数据库系统必须进行所有必要的预定安排。在这种情况下，该系统必须确保不同的代理没有预定相同的假期或飞机上相同的座位。例如，如果在从广州到北京的飞机上只剩下一个座位，两个代理在同一时间预定这最后一个座位时，系统不得不处理这种情况，只允许一个预定有效，并通知另一个代理没有位置了。

4. 使用图书馆

图书馆一般会有一个包含所有图书的详细信息数据库，其中的信息可能还包括读者信息、预定信息等。可能会允许读者基于书名、作者或其他数据查找所需书籍。数据库系统可能会允许读者预订书籍，并在书籍可以借阅时发邮件通知读者。该系统也能够给没有按期还书的借阅者发提醒通知。一般情况下，系统都有一个条形码阅读器，类似于前面超市中描述的那种，用来记录归还和借出图书馆的书籍。

5. 出租录像

当你希望从录像租赁公司租借一盘录像时，也可能会发现这个公司维护着一个数据库，这个数据库中包括关于每个录像的片名、录像有几份拷贝、这个录像目前是可以租借的还是已经被借出了、它的成员(租借者)、它的导演和演员等详细信息。租赁公司可以利用租赁信息来监视录像带的库存，并根据租借的历史数据预测公司将来的购买倾向。

6. 使用 Internet

Internet 上的很多站点是由数据库应用程序驱动的。例如，你可能访问过一个在线书店，这个书店允许借书和买书，如 Amazon.com。这个书店允许按书的种类来进行检索，比如计算机或经管，它还允许按作者来检索。不管是哪种情况，在这个机构的 Web 服务器上都有一个数据库，数据库中包含图书的详细信息、获得方式、邮寄信息、库存级别以及排列信息。

图书信息包括书名、ISBN、作者、价格、销售历史、出版社、评论以及详细的描述。数据库允许图书被交叉检索，比如，一本图书可以列在几个种类中，比如，算法、编程语言、畅销书以及推荐图书等。交叉检索也使 Amazon 可以给出一些图书的信息，这些图书一般是与借阅人所感兴趣的图书有关的。

以上只是几种数据库系统应用，毫无疑问，读者也会知道其他的更多的应用情况。尽管我们熟知并常用这些应用，但在它们的背后却隐藏着复杂的高级技术。这种技术的核心就是数据库本身。对于尽可能有效地支持最终用户需求的系统来说，需要合适的结构化数据库。构建这种结构就是所说的数据库设计。无论要构建的是小型数据库，还是前面所说的大型数据库，数据库设计都是基础。

1.2 数据库管理系统的概念

数据库管理系统(DBMS)是一个能够让用户定义、创建和维护数据库，以及控制对数据库访问的软件系统，它是与用户、应用程序和数据库进行相互作用的软件。

1.3 应用程序与数据库管理系统之间的关系

数据库应用程序是一个通过向 DBMS(数据库管理系统)发出合适的请求(一般是一个 SQL 语句)与数据库进行交互的计算机应用程序，用户与数据库应用程序交互，数据库应用程序与 DBMS 接口交互，DBMS 访问数据库中的数据。图 1.1 说明了数据库的访问形式，它显示了销售部门和库存控制部门使用其应用程序通过 DBMS 对数据库进行访问的过程。

每个部门的应用程序处理它们的数据项、维护数据和生成报表。数据的物理结构和存储由 DBMS 管理。

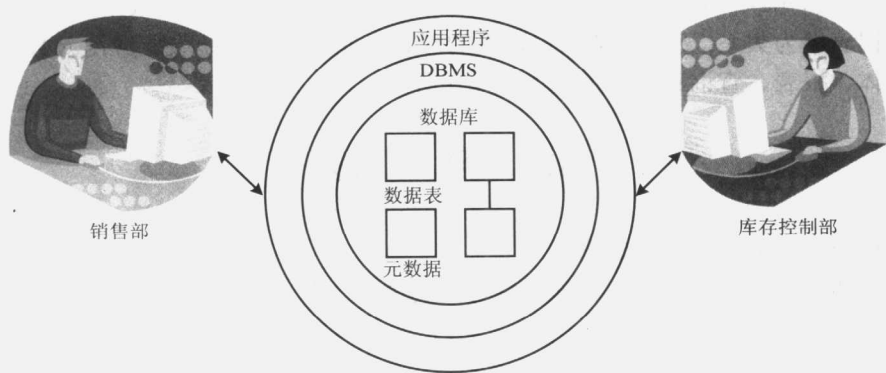


图 1.1 销售部门和库存控制部门通过应用程序和 DBMS 访问数据库的情形

1.4 数据库的数据结构

在关系数据库中，数据只能存储在结构化的数据表里，这种数据表里的每一行被称为一条数据记录，简称记录，每条记录的结构和格式是由人们在定义该数据表时决定的。例如，一个雇员的记录包括雇员编号、姓名、出生日期、雇佣日期和地址 5 个字段，如图 1.2 所示。每个字段对自己所能存储的信息类型又有着一定的要求(例如，它必须是一个有着某种特定格式的数字或者是一个字符个数不得超过某个预定义最大值的字符串)。

有 5 个字段

雇员编号	姓名	出生日期	雇佣日期	地址
0001	王飞	1975-10-6	2000-6-8	广州
0002	张池	1978-6-6	2001-5-6	北京

由字段组合成的一条记录

由字段组合成的另一条记录

图 1.2 字段与记录

1.5 数据库的定义

数据库是自描述的完整记录的集合。

数据库是自描述的：它除了包含用户的源数据外，还包含关于它本身结构的描述。这个描述称为数据字典(或数据目录，或元数据)。与图书馆相似，除了书籍以外，图书馆还包含一个描述它们的卡片目录。同样，数据字典(它是数据库的一部分，这与卡片目录是图书馆的一部分一样)描述了包含在数据库中的数据。

数据库是集成记录的集合。用户数据文件是由这些记录组合而成的。数据库不仅仅包含用户数据文件，它还包含其他内容。正如前面所说，元数据也是数据库的一部分，除此

之外,数据库还包含用来表示数据之间的关系和提高数据库应用性能的索引。最后,数据库还包含关于使用数据库的应用程序的数据,我们把这种数据称为应用元数据。

因此,数据库包含 4 个要素:用户数据、元数据、索引和应用元数据。大多数数据库把用户数据表示为关系(表),元数据也以表的形式存储,称作系统表。应用元数据用来存储用户窗体、报表、查询和其他形式的应用组件。

1.6 习 题

- (1) 列出几个数据库应用的例子。
- (2) 描述应用程序与数据库管理系统之间的关系。
- (3) 简要说明数据库的概念。

第 2 章 使用关系(表)存储用户数据

关系模型是数据库管理系统(DBMS)产品的基础，因此理解这个模型的有关概念是非常必要的。本章首先介绍关系模型的概念，然后介绍怎样实现该模型，并用该模型来存储数据。

2.1 关系模型

2.1.1 关系模型的概念

一个关系是一个二维表，如图 2.1 所示。表中的每一行保存属于某些事物(如雇员)或某些事物的某一部分数据，表中的每一列包含关于事物属性的数据。有时行称作元组，列称作属性。关系、元组和属性这些术语来自关系代数。但程序员用文件、记录和字段表达这些概念，而用户则用表、行和列来表达同样的含义。表 2.1 总结了这些关系模型的术语及含义。

EmployeeID	LastName	FirstName	Title	BirthDate	HireDate	Address
1	张	颖	销售代表	1968-12-8	1992-5-1	复兴门 245 号
2	王	伟	副总裁(销售)	1962-2-19	1992-8-14	罗马花园 890 号
3	李	芳	销售代表	1973-8-30	1992-4-1	芍药园小区 78 号
4	郑	建杰	销售代表	1968-9-19	1993-5-3	前门大街 789 号
5	赵	军	销售经理	1965-3-4	1993-10-17	学院路 78 号
6	孙	林	销售代表	1967-7-2	1993-10-17	阜外大街 110 号
7	金	士鹏	销售代表	1960-5-29	1994-1-2	成府路 119 号
8	刘	英玫	内部销售协调员	1969-1-9	1994-3-5	建国门 76 号
9	张	雪眉	销售代表	1969-7-2	1994-11-15	永安路 678 号

图 2.1 Employees 的关系

表 2.1 关系术语及其含义

术 语	含 义
关系(或表，文件)	二维窗体
属性(或列，字段)	一个关系的列
元组(或行，记录)	一个关系中的行

图 2.1 是 Employees 关系，它有 9 行(元组)，也称为 9 条记录。每一行由 7 列(属性或字段)组成，其中 EmployeeID、LastName、FirstName、Title 等为属性名或字段名。1 为字段 EmployeeID 的值。“张”为 LastName 字段的值。一个表要想成为关系，它必须满足如下条件。

- 表中的每一格必须是单值的，不允许是多值的。这样可使每行和每列的交叉点仅包含单个值。
- 每一列(字段)的所有条目都必须是同一类型。例如，有一列包含雇员的名字，另

一列包含雇员的出生日期。每一列都有惟一的名字(字段名)，列在表中的顺序不重要。每一列都有一个域，是指该列中允许取值的描述。它不仅包括数据类型的概念，也包括列中所允许的值(如数据长度和其他约束)。

- 表中任意两行不能相同，只是行的顺序无关紧要。

往后我们要讨论的所有的表都满足关系的条件，我们使用表这个术语。

2.1.2 关键字

关键字是由一个或多个属性组成的可惟一标识一行的属性或属性的组合。考虑图 2.1 所示的关系 Employees，其属性为 EmployeeID、LastName、FirstName、Title、BirthDate、HireDate 和 Address。其中 EmployeeID 的值决定了该关系中的惟一一行，因此，它是一个关键字。

关键字也可由一组属性合起来组成。考虑图 2.2 所示的关系 Order Details(订单关系表)，每一行中包含了订单号、订购的产品号及该产品的单价。如果允许一个订单可订购多种产品，那么一个 OrderID 的值就可能出现在表中的两行或多行。因此 OrderID 的值不能惟一地标识一行，这时就需要某些属性的组合，如(OrderID, ProductID)来惟一标识每一行。

应该注意的是，属性是否为关键字并非由一个抽象的规则集决定的，而是由用户意识中的模型决定的。在访问用户之后，假定发现一个订单只能订购一种产品，在这种情况下，OrderID 的值决定惟一的一行，因此，OrderID 是一个关键字。在访问用户之后，假定发现允许一个订单订购多种产品，该情形如图 2.2 所示，如前所述 OrderID 不是 Order Details 关系的关键字，例如，订单 10248 同时订购了产品 ID 值分别为 17、42 和 72 的三种产品。OrderID 的值 10248 在三个不同的行出现。事实上，对于这种关系，没有一个单独的属性是关键字，因此该表中的关键字一定是两个或多个属性的组合。

OrderID	ProductID	UnitPrice
10248	17	14
10248	42	9.8
10248	72	34.8
10249	14	18.6
10249	51	42.4
10250	41	7.7

图 2.2 Order Details(订单关系表)

考虑该表中两个属性的组合，有三种可能：(OrderID, ProductID)、(OrderID, UnitPrice) 和(ProductID, UnitPrice)。这些组合中的任意一个都是关键字吗？要想成为关键字，它必须惟一的标识一行。同样要回答这样的问题，我们必须咨询用户，不能简单地依靠图 2.2 所示的样本数据或凭自己的假设做出决定。

与用户交谈之后，假定发现几种不同的产品价格可以相同。考虑这种情况，所以 (OrderID, UnitPrice)不能惟一地决定一行。例如，订单 10248 订购四种不同的产品，两种产品的价格数都为 34.8，这就意味着(10248, 34.8)组合在表中出现两次，因此，这种组合不能成为关键字。

(ProductID, UnitPrice)组合是否可以成为关键字呢？组合(17, 14)惟一地决定一行？不

行, 它不能惟一决定一行, 因为许多订单可能订购 17 号产品。

只有(OrderID, ProductID)组合可以决定惟一的一行, 因此(OrderID, ProductID)是该关系的关键字。

 **提示:** 每个关系都至少有一个关键字。

2.1.3 域

关系的属性都有一个域。域是属性可能取值的集合, 域的特征依赖于属性的类型, 域包括物理描述和语义描述。物理描述是指属性的数据类型(例如, 数值类型和字符类型)、数据长度和其他约束(如该属性的值不能为空)。语义描述是对属性的文本描述, 用于说明属性的功能和目的。例如: Title 属性可定义为“最长为 25 个字符的字符串, 表示雇员的职务”。“最长为 25 个字符的字符串”表示该属性的数据类型为字符串, 能够表示的字符串的最大长度为 25, 它是 Title 属性域的物理描述。“表示雇员的职务”是 Title 属性域的语义描述, 它说明了 Title 属性是用来表示雇员的职务的。

2.2 列的数据类型

表中的每一列(字段)都有特定的数据类型。数据类型定义了各列(字段)所允许的数据值。SQL Server 中列的数据类型如下。

1. 整型数据

整型数据由负整数或正整数组成, 如-15、0、5 和 2509。按照整型数据数值的大小范围, 分为 bigint、int、smallint 和 tinyint 4 种。

2. 精确数值数据类型

精确数值数据类型用于存储带有小数的数据, 有 decimal 和 numeric 两种。在 SQL Server 中, numeric 数据类型和 decimal 数据类型是一样的。

3. 近似浮点数值数据类型

近似浮点数值数据类型有 float 和 real 两种, 它们不能确定所输出数值的精确度。

4. 字符数据

字符数据由字母、符号和数字组成。例如, 有效字符数据可以是 928、Johnson 和 (0*&(%B99nhjkJ 等。字符数据使用 char、varchar 和 text 数据类型存储。当列中各项的字符长度可变时可用 varchar 类型, 但任何项的长度都不能超过 8KB。当列中各项为同一固定长度时使用 char 类型(最多 8KB)。text 数据类型的列可用于存储大于 8KB 的 ASCII 字符。例如, 由于 HTML 文档均由 ASCII 字符组成且一般长度大于 8KB, 所以用浏览器查看之前应在 SQL Server 中将其存储在 text 列。

5. Unicode 数据类型

Unicode 数据使用 SQL Server 中的 nchar、nvarchar 和 ntext 数据类型进行存储。对

于存储来源于多种字符集的字符列,可采用这些数据类型。当列中各项所包含的 Unicode 字符数不同时(至多为 4000),使用 `nvarchar` 类型。当列中各项为同一固定长度时(至多为 4000 个 Unicode 字符),使用 `nchar` 类型。当列中任意项超过 4000 个 Unicode 字符时,使用 `ntext` 类型。

6. 货币数据

货币数据表示正的或负的货币值,有 `money` 和 `smallmoney` 两种货币数据类型。货币数据存储的精确度为四位小数。可以存储在 `money` 数据类型中,其取值范围是 -922 337 203 685 477.5808~+922 337 203 685 477.5807(需 8 个字节的存储空间)。可以存储在 `smallmoney` 数据类型中的值的范围是 -214 748.3648~214 748.364 7(需 4 个字节的存储空间)。如果数值超过了上述范围,则可以使用 `decimal` 数据类型代替。

7. 二进制数据

二进制数据由十六进制数表示。例如,十进制数 245 等于十六进制数 F5。二进制数据使用 `binary`、`varbinary` 和 `image` 数据类型存储。指派为 `binary` 数据类型的列在每行中都有固定的长度(最多为 8 KB)。指派为 `varbinary` 数据类型的列,各项所包含的十六进制数的个数可以不同(最多为 8 KB)。`image` 数据列可以用来存储超过 8 KB 的可变长度的二进制数据,如 Microsoft Word 文档、Microsoft Excel 电子表格、包含位图的图像、图形交换格式 (GIF) 文件和联合图像专家组 (JPEG) 文件。

数据长度超过 8 KB 时使用 `image` 数据类型存储,一般应使用 `varbinary` 类型来存储二进制数据。

8. 日期和时间数据

日期和时间数据由有效的日期或时间组成。例如,有效日期和时间数据既包括 4/01/98 12:15:00:00:00 PM,也包括 1:28:29:15:01 AM 8/17/98。日期和时间数据使用 `datetime` 和 `smalldatetime` 数据类型存储。使用 `datetime` 数据类型存储从 1753 年 1 月 1 日至 9999 年 12 月 31 日的日期(每个数值要求 8 个字节的存储空间)。使用 `smalldatetime` 数据类型存储从 1900 年 1 月 1 日至 2079 年 6 月 6 日的日期(每个数值要求 4 个字节的存储空间)。

9. 特殊数据

特殊数据包括不能用前面所述的二进制、字符、Unicode、日期和时间、数值和货币数据类型表示的数据。有五种特殊数据类型。

- **timestamp:** 用于表示 SQL Server 在一行上的活动顺序,按二进制格式以递增的数字来表示。当表中的行发生变动时,用从 @@DBTS 函数获得的当前数据库的时间戳值进行更新。`timestamp` 数据与插入或修改数据的日期和时间无关。若要自动记录表中数据更改的时间,使用 `datetime` 或 `smalldatetime` 数据类型记录事件或触发器。
- **bit:** 该数据类型只能包括 0 或 1。可以用 `bit` 数据类型代表 TRUE 或 FALSE、YES 或 NO。例如,询问客户是否为初次访问的问题可存储在 `bit` 列中。