

万水计算机编程技术与应用系列

SQL
Database
Programming
with **JAVA**

使用JAVA进行 SQL数据库程序设计

[美] Bill McCarty 著
康博创作室 译



中国水利水电出版社
www.waterpub.com.cn



万水计算机编程技术与应用丛书

使用 Java 进行 SQL 数据库程序设计

[美] Bill McCarty 著
康博创作室 译
闪四清 审校

 中国水利水电出版社
www.waterpub.com.cn

9810082

内 容 简 介

本书主要介绍如何使用 JDBC 和 SQL 语言,并且用示例说明了如何在 Java 程序中使用 SQL 语言。其内容包括客户机服务器系统介绍、关系数据库基本知识、如何使用 Java 访问数据库、JDBC API、各种 SQL 语言,以及如何处理数据库使用过程中的并发事务和安全性问题。本书适合于从事网络和数据库工作的系统人员和应用人员阅读。

“Original English language edition published by The Coriolis Group, Inc., 14455N. Hayden Drive, Suite 200, Scottsdale, Arizona 85260 USA, telephone (602) 483-0192, fax (602) 483-0193. Copyright (c) 1997 by The Coriolis Group. All rights reserved.”

北京市版权局著作权合同登记号:图字 01-98-0455 号

本书简体中文版由中国水利水电出版社出版,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

图书在版编目(CIP)数据

使用 Java 进行 SQL 数据库程序设计/(美)麦卡蒂(Mc Carty, B.)著.
-北京:中国水利水电出版社,1998.7
(万水计算机编程技术与应用丛书)
书名原文:SQL Database Programming with Java
ISBN 7-80124-752-3

I. 使… I. 麦… II. 数据库管理系统,SQL-JAVA 语言程序设计
IV. TP311.13

中国版本图书馆 CIP 数据核字(98)第 15426 号

书 名	使用 Java 进行 SQL 数据库程序设计
作 者	[美] Bill McCarty
译 者	康博创作室
审 校	闪四清
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: sale@waterpub.com.cn 电话: (010)63202266(总机)、68331835(发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 25 印张 576 千字
版 次	1998 年 7 月北京第一版 1998 年 7 月北京第一次印刷
印 数	0001—5000 册
定 价	65.00 元 (含光盘)

凡购买我社图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻版必究

译者的话

当今的世界是一个信息的世界。Internet、Intranet 和 Web 的出现,使地球变得越来越小,人们之间的距离越来越近。人类的思维方式、生活方式和工作方式随着计算机技术的发展而悄悄地正在发生革命性的变化。

Java 作为最成功的语言之一完全改变了软件开发观念,成为软件开发人员不可缺少的工具。关系数据库以其强大的数据存储能力和方便的操纵功能,越来越显示其重要性。SQL 语言作为关系数据库中最流行的语言,日益受到人们的喜欢。如何把 Java 和 SQL 结合在一起,成为人们关心的话题。

本书的目的就在于此:一方面详细介绍了如何使用 SQL 语言;另一方面又详细介绍了如何在 Java 程序中使用 SQL 语言。

本书由康博创作室策划翻译,闪四清主译并审校。但愿本书能使读者从中得到有益的收获,这是译者的最大心愿,书中不妥之处,欢迎广大读者批评指正。

康博创作室

1998 年 6 月

引 言

为什么写这本书

我写本书的目的是帮助那些知道如何使用 JDBC(访问关系数据库的 Java API),但是又不熟悉结构化查询语言(SQL)的系统开发者。有许多很好的关于 SQL 语言的书籍,但是大多数书籍主要是介绍在培训教室之外使用的并不太多的交互式 SQL 语言。因此,在用户学习 SQL 语言的同时,还提高了 Java 技术,并且学会如何在 Java 领域中应用学到的 SQL 知识。

我假设用户可以编写或者至少了解 Java 程序,并且假设用户不了解 JDBC。本书在介绍 SQL 的同时,还讨论了满足典型的应用程序设计师要求的足够的 JDBC 知识。如果用户计划写自己的 JDBC 驱动程序,那么用户应该了解更详细的 JDBC 知识。但是,大多数程序设计员非常愿意使用其他人编写的驱动程序,这样他们就可以集中精力编写更好的应用程序。本书就是为这些程序设计员而写。

我不打算详细研究这些设备和协议,例如对象持久性、远程方法析求(RMI)或者通过对象请求代理体系结构(CORBA)。SQL 的知识是有效使用这类技术的先决条件,本书的目标就是为用户提供这些先决条件。在任何情况下,这些技术发展得如此迅速,以致某一本书并不是有关这些技术的最好的信息渠道。

本书中的源代码是基于 Java 1.1 版本,这个版本包括了 JDBC API。另外,示例代码使用了 Java 1.1 扩展的事件处理机制。由于这些原因,用户会发现使这些示例代码在 Java 1.0 中运行并不简单。然而,由于示例代码遵从 Java 1.1 的约定,所以示例代码与 Java 开发工具的生成兼容。

如果用户还正在使用 Java 1.0,那么用户就会发现请变到 Java 1.1 不是特别地困难。在写作本书时,Sun 和 Microsoft 两家公司已经发行了 Java 的免费开发者工具。Symantec 已经升级了 Cafe、Visual Cafe 和 Visual Cafe Pro,Microsoft 提供了一种允许 Netscape Navigator 运行 Java 1.1 代码的内插 Java Virtual Machine。到用户读到这些时,Java 1.1 的工具将以合理的价格和稳定的性能而被广泛使用。

本书的组织结构

本书的内容是逐步从简单到复杂,常常在有关 SQL 书籍的前一二章提及的关系数据库设计理论直到本书的中间才提到。本书的前半部分讨论已经存在的数据库的使用。对于应用程序设计人员来说,这些内容比数据库设计有更加重要和易理解的实际意义。

本书提供了 Microsoft Access 的数据库查看和设计查看设备,允许用户执行某些数据库操作。这些操作使用 SQL 时相当麻烦,但是使用 Access 时相当简单。Access 提供了一种数据库直观可视的表示,简化了基本的数据库术语和概念的描述。

JDBC 以简略形式提及,这是因为其许多设备对应用程序设计人员并无必要。SQL 本身的描述按照开发的顺序进行。然而,为了避免一般 SQL 书籍的文字形式,本书使用了示例 Java 程序和图表。本书的大多数章节是围绕示例程序组织的,使用实际示例来介绍概念。

第一章和第二章提供了基于 Internet 客户服务器系统的概述,并且使读者明白如何使用 Microsoft Access 的数据库维护设备。这些章节还介绍了基本的关系数据库术语。

第三章到第七章讨论了如何使用 Java 的 JDBC 类和 JDBC-ODBC 桥访问关系数据库,以及如何使用 SQL 执行数据库查询。

第八章和第九章提出了关系数据库设计理论,并且介绍了如何使用 SQL 的数据定义语言定义关系表。

第十章和第十一章讨论了 SQL 的更加高级的功能和设备,包括数据操纵和复杂查询。

第十二章到第十四章讨论了使用 SQL 的实用课题,包括视图、多用户考虑、数据库安全性和数据库完整性。

3 个附录帮助用户学习和使用 SQL。附录 A 给出了 SQL 语言法的方便的字母排序参考;附录 B 给出了 SQL 功能的快速参考;附录 C 总结了 Microsoft Access 数据类型。

软件和硬件要求

为了运行书中的示例程序,用户需要有一台运行 Microsoft Windows 95 或者 Windows NT 4.0 或者以上版本的 IBM 兼容 PC 机。用户的 PC 至少应该是 486DX2-66 CPU 或者等级的,并且 RAM 至少是 16MB。视频卡和监视器应该至少支持 256 种颜色。大多数示例程序基于 Microsoft Access 7.0V 或者以上版本,以及相关的 ODBC 驱动程序。用户还需要 Java Developer 的 Kit V1.1 或者以上版本。这些工具可以从 <http://www.javasoft.com> 中卸载。

如果用户对本书有什么问题或者发现了什么错误,请与我联系:bmccarty@crl.com。对于光盘的任何技术性问题的帮助要求,发送电子邮件到 techsupport@coriolis.com。

目 录

译者的话

引言

第一章 数据库程序设计和客户机/服务器系统介绍	1
1.1 Java	1
1.2 数据库和数据库管理系统	3
1.3 结构化查询语言	6
1.4 计算体系结构	10
1.5 总结	16
第二章 关系数据基础	17
2.1 分解关系数据库	17
2.2 使用 Microsoft Access	20
2.3 总结	31
第三章 使用 Java 访问数据库	32
3.1 GuestBook 程序	32
3.2 安装 ODBC 数据源	39
3.3 改进 GuestBook 程序	40
3.4 Java 和 Access 数据类型	45
3.5 总结	46
第四章 SQL 查询	48
4.1 基本查询:Select 动词	49
4.2 判断和比较	52
4.3 SQLWindow 应用程序	58
4.4 另一个 GuestBook 应用程序	62
4.5 总结	70
第五章 JDBC API	71
5.1 制作连接:访问数据库	71
5.2 StudyConnect 程序	77
5.3 制作 Statement:执行 SQL 语句	92
5.4 ResultSet 对象	94
5.5 ResultSetMetaData 方法	102
5.6 SQLWindow02应用程序	103
5.7 SQLException 和 SQLWarning 类	112
5.8 DatabaseMetaData 对象	112

5.9 总结	120
第六章 排序和分组	121
6.1 按顺序放置事情:Order By 子句	121
6.2 聚集;合计	125
6.3 Microsoft Access SQL 扩展:Distinct、Top 和 Top n Percent	127
6.4 分类汇总:Group By 组	128
6.5 Having 方式;选择组	131
6.6 使用表达式表达自己的思想	131
6.7 有趣的函数	133
6.8 转义日期	137
6.9 空值	138
6.10 GuestBook04应用程序	138
6.11 总结	149
第七章 连接和联合	150
7.1 Select 总结	150
7.2 连接	151
7.3 联合(Union)	167
7.4 SQLWindow03应用程序	169
7.5 总结	184
第八章 设计数据库	185
8.1 数据库	185
8.2 E-R 图	187
8.3 数据库规范化	195
8.4 选择数据类型	200
8.5 创建一个模型数据库试验	202
8.6 数据库设计实例研究	202
8.7 总结	208
第九章 SQL 数据定义语言	210
9.1 DBMaker 应用程序	210
9.2 定义表	212
9.3 定义索引	215
9.4 定义约束	218
9.5 DBMaker 应用程序	224
9.6 总结	244
第十章 SQL 数据操纵语言	245
10.1 Insert 语句	245
10.2 多行插入	247

10.3	Update 语句	251
10.4	条件修改	252
10.5	Delete 语句	252
10.6	条件删除	253
10.7	预备语句	254
10.8	定位修改和删除	255
10.9	GuestBook 小服务	256
10.10	PreppedMark Applet	267
10.11	总结	273
第十一章	高级 SQL 查询	274
11.1	子查询	274
11.2	相关子查询	285
11.3	其他 DML 语句中的子查询	288
11.4	EmpBrowserApplet	289
11.5	总结	307
第十二章	创建和使用视图	308
12.1	使用视图	308
12.2	使用存储过程	316
12.3	CallableDemo Applet	318
12.4	BatchQuery 应用程序	323
12.5	总结	330
第十三章	多用户考虑	331
13.1	事务	331
13.2	并发性	333
13.3	DBAnomalies Applet	337
13.4	TransDemo Applet	350
13.5	总结	358
第十四章	数据库安全性和恢复性	359
14.1	数据控制语言(DCL)	359
14.2	加密	364
14.3	备份	365
14.4	数据库恢复性	366
14.5	事件日志和审计	366
14.6	Applet 安全性	366
14.7	DBGuard Applets	368
14.8	总结	375
附录 A	快速查阅 SQL 语句	376

A. 1	Alter Table	376
A. 2	Create Database	377
A. 3	Create Index	377
A. 4	Create Table	378
A. 5	Create View	380
A. 6	Delete	381
A. 7	Drop Database	381
A. 8	Drop Index	381
A. 9	Drop Table	382
A. 10	Drop View	382
A. 11	Grant	382
A. 12	Insert	383
A. 13	Revoke	383
A. 14	Select	384
A. 15	Update	385
A. 16	Use Database	386
附录 B	快速查阅 SQL 函数	387
附录 C	快速查阅 Microsoft Access 数据类型	389

第一章 数据库程序设计和客户机/服务器系统介绍

本章提供了一些背景材料,有助于读者理解 Java、JDBC、SQL 和 DBMS 是如何一起工作来支持典型的应用程序。由于读者正在阅读本书,所以可能已经知道一些 Java 程序设计的知识,但是可能不知道这种较深入的内容:Java 如何进行基于网络的应用软件开发。读者可能熟悉或不熟悉 JDBC、SQL 和 DBMS。如果不熟悉这些术语,也不用担心,本章的目的就是解释它们,并且使读者理解如何顺应基于网络的应用程序系统。

如果读者熟悉这些术语,建议快速阅读这一章,但是不要整个跳过这一章。这一章中的一些有用的细节内容有助于理解随后章节的内容。

本章开始解释 Java 如何正在改变客户机/服务器应用程序开发人员的工作。许多开发人员发现,现在他们不仅必须掌握用来创建在公司的 intranet 上运行的应用程序的技术,而且还必须掌握创建在 Internet 和 Web 上运行的应用程序的技术。接着,本章转到数据库和 DBMS 话题,解释这些内容及其功能。接下来,描述了 JDBC,JDBC 是一种允许 Java 程序处理 SQL 语句和命令的“补偿物”。最后,介绍了几种创建客户机/服务器系统的体系结构。

1.1 Java

越来越多的程序设计人员正在学习 Java,并且把 Java 作为他们的首选的和唯一的程序设计语言。如果你也是这些人员的其中之一,那么,掌握本节的内容就特别重要。本节内容有助于理解 Java 特别之处和 Java 是如何调整客户机/服务器应用程序开发人员的工作。

1.1.1 Java 是什么

Java 是一种基于 C++ 的现代面向对象的语言。目前,C++ 仍然是比较流行的语言,但是 Java 发展很快。不久的将来,用户可能会听到这种说法:C++ 是 Java 的“中年和重荷之父”。现在,有经验的 Java 程序设计人员挣的钱已经比有经验的 C++ 程序设计人员多(尽管在最近两年某个程序设计人员如何被称为“有经验的”,这一点已经不太清楚了)。

不可否认,C++ 是一种强有力的程序设计语言。但是,实践证明,学习 C++ 非常困难,并且 C++ 不太胜任创建可靠的应用软件。简单地说,C++ 允许程序设计人员做危险的又易错的事情,例如直接操纵地址指针。这种做法可能增加程序错误的数量和严重性。

Java 程序设计人员试图创建一种比 C++ 学习和使用起来都简单的语言。他们想要一种比 C++ 更加合适的工具用于开发可靠的软件。为了实现这些目标,他们从 Java 中删除了某些他们认为是危险的和不必要的 C++ 功能,包括运算符重载、指针操纵和多继承性。

Java 的设计人员没有停止修正他们新语言的框架,他们增加了新的功能,包括垃圾收集、多线程和组网。最近,JavaSoft 增加了 JDBC 工具,允许 Java 程序员访问 SQL 数据库。总之,这些认真选择的删除和增加内容产生了一种容易使用和有趣的语言。这种语言非常适合

编写可靠的应用软件。当然,在我们了解 Java 的承诺是否可达到之前,我们还需要更多的开发和使用 Java 应用程序的实践经验。

1.1.2 为什么 Java 对于客户机/服务器开发人员很重要

Java 的易学易用和其提高软件可靠性的可能性足以使其成为一种高级语言。但是,这些原因与 Java 的主要承诺相比是次要的,其主要承诺是:便要软件。Java 的特点是“一次编写,任何地方运行”。意思是在安装 Pentium 处理器的 IBM 兼容 PC 机上,使用 Microsoft Windows NT 编写的 Java 程序,不用修改,可以在 Macintosh、运行 Solaris 的 Sun 工作站和许多其它平台上直接运行。

Internet 和 Web 的出现使以前不太重要的便携性问题成为了主要问题。想象一样,如果存储在 Sun Solaris 服务器上的 Web 页只能由使用 Sun 工作站的人们查看,那么这种结果会是如何混乱。把 Web 分割成 Microsoft 用户领域、Sun Solaris 用户领域、Macintosh 用户领域等等,将会削弱 Web 的最有价值的功能——通用性。

虽然 Web 页是可携的,但是软件程序不是。把在某一种计算机系统上编写的程序转变成可以在另一种计算机系统上运行,这种工作一般耗时多、成本高,并且常常是不可能的。

Java 通过依靠一种解释程序突破了这种可携性限制。这种解释程序自动地将 Java 程序要译成某种给定计算机平台可接受的形式。因为编写了 Java 的解释程序,所以 Java 程序不用修改就可以在任何平台上运行。目前,Java 解释程序已经广泛地应用在大多数计算机平台上。

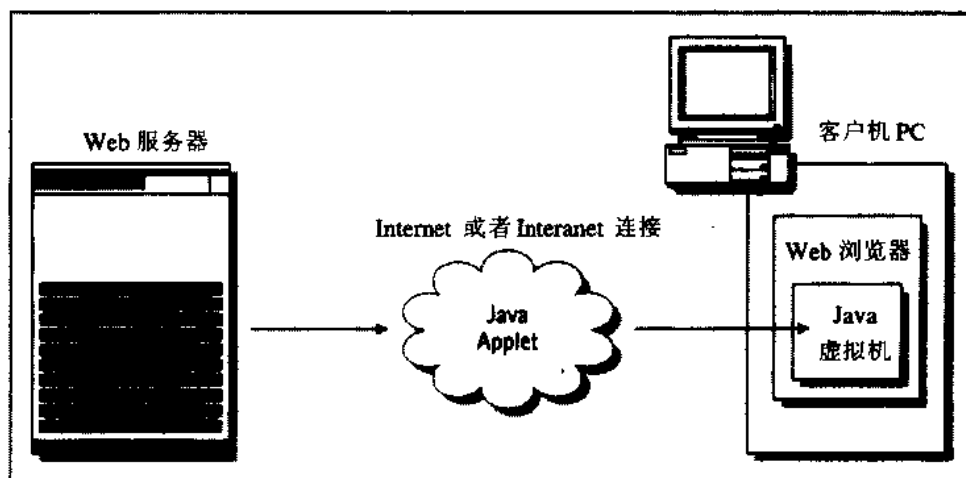


图 1.1 卸载和执行 Java 程序

Java 便携性的显著特点是 Java 程序(一般是指 Java applets)可以包含在 Web 页中,并且可以由标准的 Web 服务器下载到 Web 浏览器中,Web 浏览器可以执行 Java 程序(见图 1.1)。服务器和运行浏览器的平台是相同的或者只是相似的都没有关系,Java 解释程序把

Java 程序翻译成客户机平台上可以执行的形式。如果某种浏览器不适合作为运行 Java 程序的设备,那么这个程序可以编写成无浏览器辅助运行的应用程序。

企业对于使用 Java 作为一种保护自己的产品免遭硬件变化而淘汰的手段很感兴趣。使用便携式语言编写程序,减少了对某一种计算机平台的依赖性,允许企业扩展或者修改其计算机。使用非便携式语言编写应用程序则把商业产品捆绑到某一种平台上。这样可能由于该平台价格出人意料地上升,或者由于技术过时或者制造商企业失败而变成不能使用。Java 对企业的另外一种吸引力是其具有在 Web 页中创建可执行程序的功能。很多商业公司看到基于 Web 的销售和顾客服务是为他们带来利润的关键因素,所以他们争先恐后地构造和部署力图吸引顾客的 Web 站点。Java 程序设计人员可以创建交互式的动态的与顾客通信和娱乐的 Web 页。

随着企业对 Web 兴趣的增加,软件开发人员的工作开始发生了变化。开发人员正在创建越来越多的基于网络的系统,经常使用 Internet 链接分散的商业单位。由于其易用和强大的组网应用程序界面,因此 Java 是迄今为止编写这类系统的最好的工具。类似地,企业对于要求开发人员构造通过 Internet 或者 intranet 分布的 Web 页的 Java 程序的需求正在增加。

这种 Java 程序经常需要访问存储在数据库中的数据,毕竟大多数软件系统最终要使用数据。这就是为什么 SQL 是客户机/服务器开发人员的一种重要工具。此时,用户可能认识到自己也需要 Java 技术,想了解更多的 SQL 知识来构造未来的应用程序系统,为自己的技术和服务创建更多的功能。

1.2 数据库和数据库管理系统

许多现代应用软件系统在数据库中存储数据,使用数据库管理系统管理数据。为了了解客户机/服务器系统的革命,用户需要了解数据库。本节描述数据库和 DBMS。

1.2.1 数据库是什么

很久以前,没有数据库,计算机数据存储在文件中。这个很久以前,也只是 20 年之前。在这 20 年内,硬件和软件技术已经发生了很多变化。

以前文件很重要,现在还是很重要,因为存储在文件中的数据是永久性的。当程序完成后或者当关闭计算机后,文件不会丢失。许多程序设计语言,例如 COBOL、FORTRAN 和 Basic,提供直接支持读写文件,大多数其他程序设计语言,例如 C、C++ 和 Java,提供了包含文件输入输出程序库。

然而,经过大量的实践,程序设计人员认为以文件形式存储数据有明显缺点。大多数应用程序不能只在一个文件中存储全部数据。文件一般包含单个类型的数据,例如销售、费用或者零件库存。因为大多数应用程序处理多种实体类型,为了存储数据,这些程序需要多个文件。问题是当这些文件放在一起时,容易把文件位置放错了、日期版本弄乱了或者被破坏。这种应用程序系统缺乏保持文件同步的机制。

大多数操作系统,例如 MS-DOS 和 Windows 95,没有提供保护存储在文件中的数据避

免非授权访问或者有意或者无意修改数据的工具。无论怎样,都不能简单防止非授权用户使用文件。另外,不能使文件让多个授权用户同时使用——大多数操作系统没有提供足够的共享写访问文件数据的工具。

当然,虽然应用程序设计人员可以写出提供这些必要工具的应用程序代码,但是这样就会明显提高开发应用程序的成本。况且,只有少数应用程序设计人员具有编写这种工具的专门技能。

1.2.2 数据库管理系统

解决程序设计人员困境的方法是 DBMS,它扩展了的操作系统包括需要管理应用程序数据的工具。DBMS 在数据库中存储应用程序信息,信息集合以单个表形式存储。表存放以前存储在文件中的数据。这些表可以或者不可以作为操作系统文件形式出现。通常的方法是整个数据库作为一个单独文件存储。这种文件包含了不能使用标准文件编辑器修改的二进制。DBMS 的工作是存储和保护数据,使多个并行用户使用。典型的 DBMS 提供了允许用户使用和管理数据库的机制。用户可以:

- 从表中检索数据;
- 向表中插入新数据;
- 修改表中数据;
- 删除表中数据。

商业公司通常在一个或者几个数据库中存储其全部计算机化的数据。这样就很容易在商业公司的多个应用程序系统之间共享数据。例如,对每个应用程序,不用有单独的帐户系统文件,可以访问数据库中同样共享的表。这样就避免了同样的数据有几种不同的或者可能不一致的版本,这种问题通常称作“数据库难题”。这不是因为由数据库引起的问题,而是因为是由数据库解决的问题。

在使用过程中,数据库必须增加新信息。同样,数据库还必须清除不再需要的信息。因此,DBMS 提供了如下机制:

- 增加新表;
- 改变已经存在的表的结构;
- 删除存在的表。

这些操作一般由称作数据库管理员的特殊用户执行,他还可以定义控制访问数据和保护数据完整的策略和规则。例如,数据库管理员可以指定信贷或者应收帐款部门中的雇员可以查看客户付款的历史数据,但是在人事部门的雇员则不能查看这些数据。一般,DBMS 还提供很多实用程序,用来备份数据库或者修复损坏的数据库。

1.2.3 使用 DBMS 的其它好处

除了解决由于使用普通文件存储数据产生的问题之外,使用 DBMS 还有许多好处。其中一些好处是:

- 数据独立性

• 提高效率

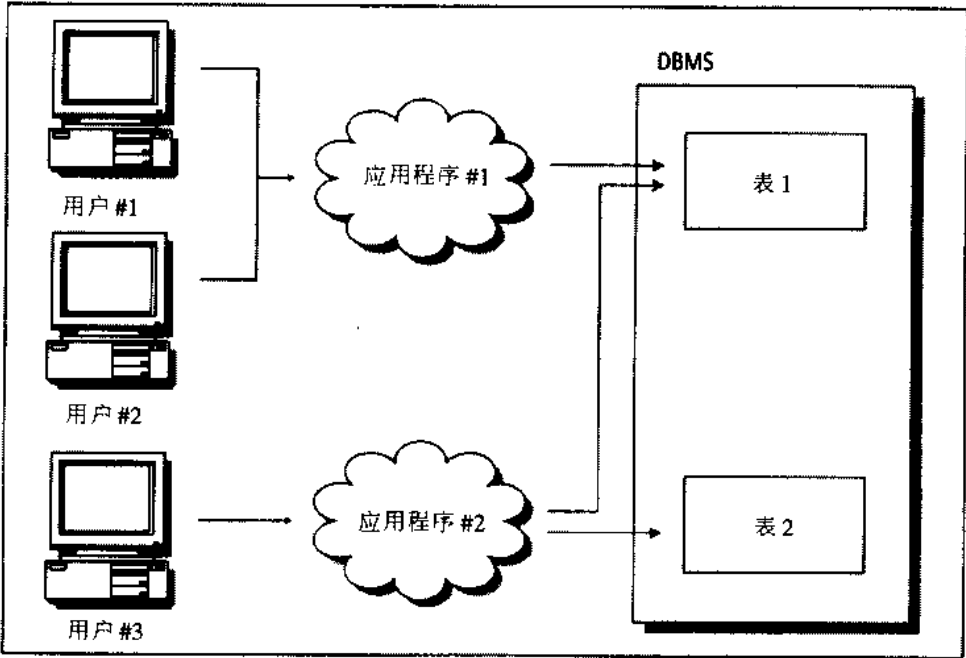


图 1.2 数据库管理系统(DBMS)

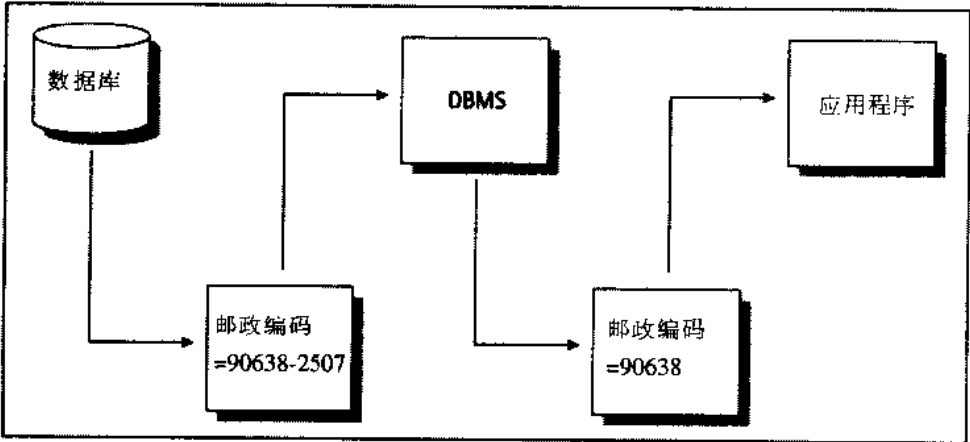


图 1.3 数据独立性

由于 DBMS 的垄断数据而形成了数据独立性,应用程序除了由于 DBMS 的许可和协助从数据库中获取数据外,应用程序再也不能获取数据,如图 1.2 所示。由于 DBMS 处于数据和程序之间的重要位置,所以它可以把数据转变或者翻译成应用程序使用更加方便的形式。

示例,假设在数据库中一个客户地址的表中,客户地址包括了 5 位数字的邮政编码。最近,邮局宣布使用在原来邮政编码后面增加 4 位数字后缀的扩展邮政编码。目前存在的程序只能处理 5 位数字编码,不能处理新的 9 位数字编码。然而,数据库必须存储全部 9 位数字。

如图 1.3 所示,DBMS 通过对新程序提供 9 位数字编码轻易地解决这个问题,但是对其它程序只提供 5 位数字编码。因为程序不能直接访问数据库中的数据,只能通过 DBMS 来访问数据,所以程序不知道数据存储的实际形式。DBMS 可以把存储的数据映射成更能满足程序需要的另外一种形式。

从提供给应用程序的数据形式到数据存储形式是一种强大的设备。它使 DBMS 以最有效的格式自由地存储数据。DBMS 简单地根据应用程序的要求重格式化数据,它仍然保留数据的实际形式。

1.2.4 一些实际数据库管理系统

现在,有许多 DBMS 供应商和产品。表 1.1 列出了一些常见的 DBMS 产品。

表 1.1 一些著名的 DBMS 供应商和其产品

供应商	DBMS	Y
Borland	Interbase	
IBM	DB2	
Informix	Universal,Server,Ohline Workgroup,Server 等等	
Microsoft	Access,SQL Server	
ORACLE	Oracle Server 7	
Sybase	SQL Server	

在 IBM 兼容 PC 领域中,一种特别重要的 DBMS 是 Microsoft Access。它作为 Microsoft Office Professional 的一部分,使用成本很低。虽然 Access 功能和性能有限,但是它是学习数据库的一种非常好的工具。另外,使用 Access 开发的大多数软件可以修改成在 Microsoft 的 flagship DBMS、SQL Server 中使用。虽然本书的示例代码是基于 Access,但是所描述的概念和技术适用于许多 DBMS 产品。

1.3 结构化查询语言

根据上一节的内容,用户已经知道,那种程序设计人员遇到的问题随着 70 年代初 DBMS 的广泛使用而解决。遗憾的是,正如软件的历史发展情况,解决了一种问题的结果是又引出了新的、更加困难的问题。

在数据库技术方面,新问题是访问数据很困难。如图 1.4 所示,数据库变成了一个类似于紧紧锁住用户数据的中世纪城堡,阻止对数据的访问和修改。无论何时用户需要的数据超出了应用程序程序化报告提供的例程范围,那么就必須写一个新程序。通常这样做的成本很高。甚至有时,到用户收到其需要的数据的时候,用户几乎回忆不起来他们为什么需要数据。简而言之,访问数据的过程太费力、费钱和费时。

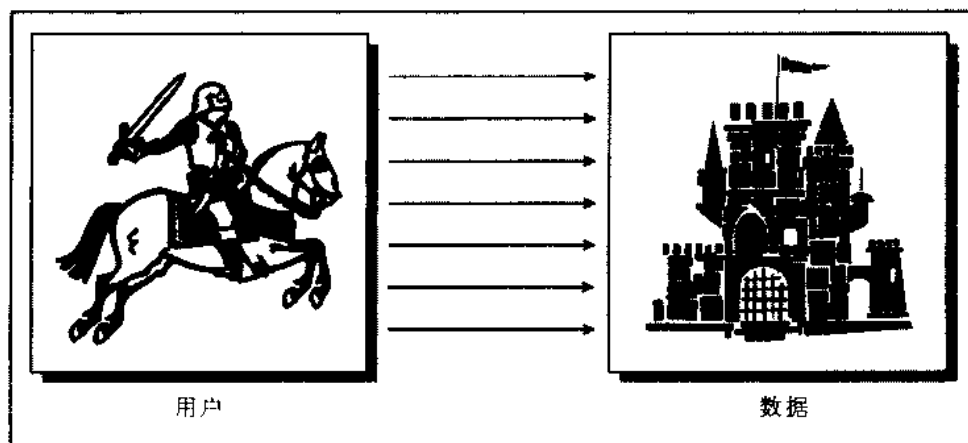


图 1.4 数据访问问题

1.3.1 SQL 拯救

1974 年, IBM 研究实验室的 E. F. Codd, San Jose, 提出了一种访问数据库中数据的新方法。他的新方法被称为结构化查询语言, 简称为 SQL。

SQL 是一种访问数据库的语言。程序设计人员或者熟练的终端用户, 通过使用 SQL, 可以对存储在数据库中的数据进行检索、修改、定义和管理。在 SQL 开始阶段, 它就被认为是突破城墙、迅速到达和轻易访问数据的方法。程序清单 1.1 显示了 SQL 查询的一个示例, 从一个关系数据库表中检索数据。注意, 其含义非常清楚, 甚至对于不熟悉 SQL 语言的人也能看懂。

程序清单 1.1 SQL 查询的一个示例

```
SELECT VendorID, VendorName  
FROM VendorTable  
WHERE DaysSincePurchase <= 90 AND YTD Purchases > 10000  
ORDER BY VendorID;
```

然而, 以前 SQL 的一个问题是, 它访问的数据库必须满足称为关系模型的结构, 这种模型本书第 2 章讨论。遗憾的是, 刚引入 SQL 时, 使用关系模型的数据库远远不符合这种规则。更糟的是, 早先的关系数据库性能很差, 访问或者修改数据常常要求比与其竞争的非关系数据库更长的时间。

9810082