

DEITEL™ DEVELOPER SERIES

Java

Web 服务高级教程

Java Web Services for Experienced Programmers

包含
103 个 Live-Code™

示例

(美) Harvey M. Deitel 等著
邱仲藩 等译

- ▶ XML/DTD/DOM/XSLT
- ▶ SOAP
- ▶ WSDL 和 UDDI
- ▶ JAX-RPC
- ▶ JAXM
- ▶ SAAJ
- ▶ JAXR
- ▶ JAXP
- ▶ Apache Axis
- ▶ JWSDP
- ▶ Web 服务安全性
- ▶ J2ME 和无线 Web 服务
- ▶ Servlet/JSP

介绍

Java Web Services Developer Pack

机械工业出版社
China Machine Press

DEITEL DEITEL GADZIK LOMELI SANTRY ZHANG

DEITEL™ DEVELOPER SERIES

Java

TP312
1170

Web 服务高级教程

Java Web Services for Experienced Programmers

包含

103 个 Live-Code™

示例

(美) Harvey M. Deitel 等著

邱仲潘 等译

- ▶ XML/DTD/DOM/XSLT
- ▶ SOAP
- ▶ WSDL 和 UDDI
- ▶ JAX-RPC
- ▶ JAXM
- ▶ SAAJ
- ▶ JAXR
- ▶ JAXP
- ▶ Apache Axis
- ▶ JWSDP
- ▶ Web 服务安全性
- ▶ J2ME 和无线 Web 服务
- ▶ Servlet/JSP

北方工业大学图书馆



00534982

介绍



机械工业出版社
China Machine Press

Java Web Services Developer Pack

DEITEL DEITEL GADZIK LOMELI SANTRY ZHANG

本书首先概要介绍作为Web服务技术基础的XML的基本知识,然后深入探讨一些更为实用的内容,包括SAOP(简单对象访问协议)、WSDL(Web服务描述语言)、JAX-RPC(基于XML的远程过程调用的Java API)。并通过丰富的实例来介绍编程和软件工程原则。

本书简明、易懂、实用性强,适于有一定Java编程经验并希望深入了解Web服务技术的专业人员。

Harvey M. Deitel, et al: Java Web Services for Experienced Programmers (ISBN 0-13-046134-2).

Authorized translation from the English language edition entitled Java Web Services Experienced Programmers by Harvey M. Deitel, published by Pearson Education, Inc, publishing as Prentice-Hall, Copyright © 2003 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2003 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号:图字:01-2003-0995

图书在版编目(CIP)数据

Java Web服务高级教程/(美)戴特(Deitel, H. M.)等著;邱仲潘等译.-北京:机械工业出版社,2003.7

(Sun公司核心技术丛书)

书名原文:Java Web Services for Experienced Programmers

ISBN 7-111-12213-5

I. J... II. ①戴... ②邱... III. Java语言-程序-设计-教材 IV. TP312

中国版本图书馆CIP数据核字(2003)第039008号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:刘卫宏 贾梅

北京中加印刷有限公司印刷·新华书店北京发行所发行

2003年9月第1版第1次印刷

787mm×1092mm1/16·35.5印张

印数:0 001-4 000册

定价:59.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前言

在自己喜欢的两样东西之间做出选择是很困难的。无论是 PHP 还是 MySQL,多年来都提供了令人愉悦的 Web 应用程序编程体验。但我首先写的是 PHP。自《Core PHP Programming》一书出版以来,我亲眼目睹了 PHP 的逐渐流行。两年后,我终于有时间来写作本书。MySQL 和 PHP 是水乳交融的两个工具。我希望你会发现本书是我的前一本书的有价值的姊妹篇。

你可能已经注意到本书封面上的 MySQL 徽标,这代表着本书两个重要的特点。首先, Monty Widenius 和 Kaj Arnö 已在本书出版之前以个人名义审阅了它。这样便保证了书中的信息有很高的价值。其次,MySQL AB 公司也将从本书的销售中获利。这意味着通过购买本书,你将为 MySQL 的后续开发贡献出力量。

通过其他渠道来支持 MySQL 的开发是非常重要的。你可能是以 GNU Public License 的名义获得 MySQL,这意味着为此花的钱是微不足道的。此外,更重要的是,你有权修改源代码,并同其他人分享你所做的修改,只要一切都遵守许可证制定的规则。尽管开放源代码软件模型并不是什么新东西,但它确实是最近才流行起来的。MySQL AB 和传统的软件公司不同,不能靠出售打包软件来赢利。由于任何人都可出于不同的目的而下载和使用 MySQL,所以 MySQL AB 公司要通过提供与产品相关的卓越服务来维持赢利。

本书从一般意义上描述了关系型数据库,具体的讨论则围绕 MySQL 展开。我希望这样的写作思路能被所有聪明的、敢于接受新思想的读者理解。你无需有任何数据库经验,但应该理解如何操作现代化的计算机。要学习 MySQL,你应该会安装软件,并会使用计算机的命令行 shell。当然,稍微有一些编程经验更好。

我设想你会经常参考本书,所以在书的中间位置安排了一个函数和命令参考。之所以这样做,是因为你可能经常性地将书摊开,从中查找需要的函数和命令。另外,我会尽可能地简化文字。一本印刷好的书应该能够方便读者自行控制阅读速度,并可在任何时候回过头去重新阅读。因此,我极少重复讲过的内容。

本书第 I 部分从常规意义上描述了关系型数据库。第 1 章为那些没有经验的读者简要介绍了数据库,然后讨论了安装以及同服务器的交互。第 I 部分的其他章则探讨了数据库功能的细节,其中包括 SQL 语言。

第 II 部分是对 MySQL 所支持的命令与实用程序的一个完全参考。信息涉及到数据类

型、变量和运算符。在这一部分,你可找到针对每个函数、SQL 命令和实用程序的说明。第 15 章讨论了用于同 MySQL 交互的 C API。

第 III 部分讨论了几种流行的编程语言,并描述了同一个 MySQL 服务器交互的过程。在此涉及的语言包括 C, C++, Java, Perl, PHP, VBScript 以及 Python。这些章的讨论是假定读者已经非常熟悉这些语言。

最后一部分讲解了一些高级主题,其中讨论了 MySQL 对物理资源的使用情况,以及如何对服务器进行管理。此外,还讨论了复制、对象映射以及编写自己的 SQL 函数。

本书的讲解统一基于 MySQL 版本 3.23,这是我写作本书时最新的稳定版本。在写作过程中,还发布了几个小的版本升级,书中也将它们包含在内。读者读到本书时,MySQL 4.0 可能已经成为稳定版本发布。所以,我在书中对 3.23 可能发生改变的功能进行了说明。

致 谢

本书的面世,离不开我的妻子 Vicky。她不仅是一名出色的技术编辑,还不断地给予我鼓励、理解和支持。

除了 Vicky 之外,在此我还要感谢另外几名优秀的技术编辑。Dan Staten 和 Dan Livingston 从非专业的角度提供了出色的反馈意见。Monty Widenius 和 Kaj Arnö 虽然工作很忙,但都为我提供了详细而深刻的意见。我还要感谢 Sinisa Milivojevic,她和我一起对 MySQL++ API 进行了颇有助益的讨论。

同 Mark Taub 和 Prentice Hall 合作令人愉快。尤其高兴的是,我拥有较大的自由来安排本书的结构。另外,非常高兴 Mark 批准了我同 MySQL AB 的会面。这正是开放源代码社区的精神。

目 录

第 I 部分 MySQL 和关系模型

第 1 章 MySQL 入门	3
1.1 人们如何使用数据库	3
1.2 数据库比文件好的原因	11
1.3 数据库不是万灵药	13
1.4 为什么选择 MySQL	13
1.5 MySQL 的历史	14
第 2 章 安装 MySQL	16
2.1 下载安装文件	16
2.2 用 Linux RPM 安装	17
2.3 Windows 安装程序	18
2.4 编译源代码	20
2.5 授予权限	20
第 3 章 同 MySQL 交互	22
3.1 TCP/IP 上的客户机/服务器	22
3.2 命令行实用程序	23
3.3 图形客户机	24
3.4 ODBC	26
3.5 Web 接口	26
第 4 章 数据库概念	28
4.1 历史	28
4.2 术语	29
4.3 数据库管理系统	30
4.4 平面文件数据库	31
4.5 层次化数据库	31
4.6 关系型数据库	33
4.7 对象数据库	34
4.8 对象关系型数据库	35

第 5 章 关系模型	36
5.1 关系代数	36
5.2 表、行和列	36
5.3 键	38
5.4 关系	40
5.5 关系运算	41
5.6 MySQL 是否为真正的 RDBMS	47
第 6 章 结构化查询语言	48
6.1 SQL 是第 4 代语言	48
6.2 数据定义	49
6.3 插入行	51
6.4 更新行	52
6.5 删除行	52
6.6 查询	52
6.7 联接	53
6.8 排序	56
6.9 分组	56
6.10 限制	57
6.11 更改表	58
第 7 章 数据库设计	59
7.1 需求规格	59
7.2 设计规格	63
7.3 制图	64
7.4 建模语言	66
7.5 ER 图	67
7.6 创建图	68
7.7 实现设计	68
7.8 测试	71
7.9 规划数据库生命周期	71

第 8 章 正规化	72	第 13 章 SQL 语句	163
8.1 为什么要正规化	72	13.1 注释	163
8.2 第一范式	73	13.2 Alter Table 语句	164
8.3 第二范式	75	13.3 Analyze Table 语句	168
8.4 第三范式	77	13.4 Backup Table 语句	168
8.5 Boyce-Codd 范式	78	13.5 Begin [WORK] 语句	169
8.6 第四范式	79	13.6 Change Master 语句	169
8.7 反正规化	81	13.7 Check Table 语句	170
第 9 章 事务处理和并发性	83	13.8 Commit 语句	171
9.1 并发性	83	13.9 Create Database 语句	171
9.2 事务	84	13.10 Create Function 语句	172
9.3 锁定	87	13.11 Create Index 语句	172
9.4 序列	89	13.12 Create Table 语句	172
 第 II 部分 MySQL 参考		13.13 Delete 语句	178
第 10 章 数据类型、变量和表达式	93	13.14 Describe 语句	179
10.1 数据类型	93	13.15 Drop Database 语句	180
10.2 变量	96	13.16 Drop Function 语句	181
10.3 运算符	97	13.17 Drop Index 语句	181
10.4 表达式	104	13.18 Drop Table 语句	181
10.5 名称中的空格	105	13.19 Explain 语句	182
第 11 章 列和索引类型	106	13.20 Flush 语句	183
11.1 数值	106	13.21 Grant 语句	184
11.2 字符串	109	13.22 Insert 语句	186
11.3 时间值	111	13.23 Kill 语句	187
11.4 列类型别名	113	13.24 Lock Tables 语句	188
11.5 索引	114	13.25 Load Data Infile 语句	189
第 12 章 内建函数	116	13.26 Load Table 语句	190
12.1 调试与配置	116	13.27 Optimize Table 语句	191
12.2 流程控制	119	13.28 Purge Master Logs 语句	191
12.3 分组	122	13.29 Rename Table 语句	191
12.4 数学运算	126	13.30 Repair Table 语句	192
12.5 字符串	133	13.31 Replace 语句	192
12.6 时间	147	13.32 Reset Master 语句	193
12.7 杂项	159	13.33 Reset Slave 语句	193
12.8 过程	161	13.34 Restore Table 语句	193
		13.35 Revoke 语句	193

13.36 Rollback 语句	194	14.16 mysqlbug	247
13.37 Select 语句	194	14.17 mysqlc	247
13.38 Set 语句	199	14.18 mysqld	247
13.39 Set Transaction 语句	202	14.19 mysqld-max	260
13.40 Show Columns 语句	203	14.20 mysqld-nt	261
13.41 Show Create Table 语句	203	14.21 mysqld-opt	261
13.42 Show Databases 语句	204	14.22 mysqld_multi	261
13.43 Show Grants 语句	204	14.23 mysqldump	263
13.44 Show Index 语句	205	14.24 mysqldumpslow	269
13.45 Show Logs 语句	206	14.25 mysqlhotcopy	271
13.46 Show Processlist 语句	206	14.26 mysqlimport	273
13.47 Show Status 语句	206	14.27 mysqlshow	276
13.48 Show Table Status 语句	208	14.28 pack_isam	278
13.49 Show Tables 语句	209	14.29 perror	278
13.50 Show Variables 语句	210	14.30 replace	279
13.51 Slave 语句	213	14.31 safe_mysqld	280
13.52 Truncate 语句	214	第 15 章 C API	282
13.53 Unlock Tables 语句	214	15.1 类型	282
13.54 Update 语句	214	15.2 客户机函数	286
13.55 Use 语句	215	15.3 数组函数	312
第 14 章 命令行实用程序	216	15.4 字符集函数	313
14.1 环境变量	216	15.5 文件函数	315
14.2 选项文件	217	15.6 错误函数	318
14.3 comp_err	218	15.7 散列函数	319
14.4 isamchk	218	15.8 列表函数	320
14.5 make_binary_distribution	219	15.9 内存管理函数	320
14.6 msqld2mysql	219	15.10 选项函数	322
14.7 my_print_defaults	219	15.11 密码函数	322
14.8 myisamchk	220	15.12 字串函数	322
14.9 myisamlog	225	15.13 线程函数	325
14.10 myisampack	227	第 III 部分 编写 MySQL 客户机程序	
14.11 mysql	228	第 16 章 C API 编程	329
14.12 mysql_install_db	238	16.1 准备编写程序	329
14.13 mysqlaccess	238	16.2 获取数据	331
14.14 mysqladmin	239	16.3 处理数据	334
14.15 mysqlbinlog	245		

第 17 章 JDBC	339	24.3 表的类型	385
17.1 准备编写程序	339	24.4 列	391
17.2 获取数据	341	24.5 表锁	393
17.3 处理数据	343	24.6 索引	393
第 18 章 VBScript 及 ODBC	346	24.7 文件描述符	395
18.1 准备编写程序	346	24.8 系统内存	396
18.2 获取数据	348	24.9 日志文件	396
18.3 处理数据	350	第 25 章 灾难应对	401
第 19 章 PHP	354	25.1 检查与修复表	401
19.1 准备编写程序	354	25.2 备份与恢复	403
19.2 获取数据	355	第 26 章 优化	410
19.3 处理数据	358	26.1 优化之前	410
第 20 章 PERL	361	26.2 瓶颈	411
20.1 准备编写程序	361	26.3 设计优化	415
20.2 获取数据	362	26.4 应用程序内部的优化	416
20.3 处理数据	364	26.5 优化查询	417
第 21 章 PYTHON	366	26.6 优化 SQL 语句	421
21.1 准备编写程序	366	26.7 维护表	422
21.2 获取数据	367	26.8 调节服务器配置	422
21.3 处理数据	369	26.9 重新编译 MySQL	424
第 22 章 MySQL++ API	371	第 27 章 安全性	428
22.1 准备编写程序	371	27.1 权限方案	428
22.2 获取数据	372	27.2 设置权限	434
22.3 处理数据	374	27.3 确保安全性	436
第 IV 部分 高级主题		第 28 章 转换	439
第 23 章 数据库管理	379	28.1 切换数据库服务器	439
23.1 职责	379	28.2 把不支持的特性转化到 MySQL 中 ...	440
23.2 使数据可用	379	28.3 使用 ANSI 模式	445
23.3 维护数据库完整性	380	28.4 独一无二的 MySQL 特性	446
23.4 准备应对灾难	381	第 29 章 分布式数据库	452
23.5 支持用户	382	29.1 分布式数据库的概念	452
23.6 编写并强制执行标准	383	29.2 被延迟的同步	455
第 24 章 物理存储	384	29.3 MySQL 的复制功能	457
24.1 数据库/表模型	384	29.4 运行多个服务器	462
24.2 专用分区	385		

第 30 章 对象映射	465	C.1 MySQL 的许可证	499
30.1 面向对象模型	465	C.2 GNU 通用公共许可证	499
30.2 序列化对象	467	C.3 稳定性	504
30.3 对象-关系映射	470	C.4 支持	506
第 31 章 添加功能	481	附录 D 保留字	507
31.1 调试库	481	附录 E MySQL 的错误码	511
31.2 添加字符集	484	附录 F SQL 样式指南	521
31.3 函数	489	F.1 一般样式	521
31.4 过程	492	F.2 标识符	521
附录 A 在线资源	494	F.3 表	522
A.1 官方邮件列表	494	F.4 语句	522
A.2 邮件列表归档	495	附录 G 数据库设计示例	523
A.3 Web 站点	496	G.1 示意图	523
附录 B 进一步阅读	498	G.2 SQL 模式	524
附录 C 商务问题与法律问题	499		

第 I 部分 MySQL 和关系模型

本部分将介绍关系型数据库的一些基本概念。第 1 章“MySQL 入门”简要介绍了关系型数据库服务器提供的各种功能。第 2 章“安装 MySQL”引导你安装 MySQL。第 3 章“同 MySQL 交互”讨论了用于同数据库服务器交互的程序,其中包括如何运行命令行实用程序,如何运行具有图形用户界面的应用程序,以及如何使用 ODBC 接口。

第 4 章“数据库概念”讨论了数据库的演变及其核心概念。我们将学习关系模型问世之前的数据库模型,以及用于取代关系模型的一些新式模型。第 5 章“关系模型”详细讲述了关系模型,以及 MySQL 具体是如何实现它的。第 6 章“结构化查询语言”介绍了在 MySQL 以及其他大多数关系型数据库服务器中用于处理数据的结构化查询语言。

第 7 章“数据库设计”针对数据库的设计提供了一些建议,涉及的内容从编写规格文档一直到制图。第 8 章“正规化”讨论了正规化过程,它有助于消除冗余数据。第 9 章“事务处理和并发性”讨论了与并发性有关的问题,即允许多个用户同时使用数据库,还提供了使用事务处理和锁来管理并发性的一些策略。

第 1 章 MySQL 入门

本章要点：

- 人们如何使用数据库
- 数据库比文件好的原因
- 为什么选择 MySQL
- MySQL 的历史

本章将介绍基本的数据库概念，并解释 MySQL 如何实现这些概念。我们将学习 MySQL 如何试图解决常见的数据管理问题，还将了解 MySQL 为什么是众多数据库服务器中的一个最好的选择。

由于本章提供的只是一个概述，所以会涉及到大量主题，但都不会非常深入，后面的章节将详细解释这些主题。如果你是一名数据库专家，那么大多数内容对你而言只是一次复习。如果愿意，你完全可以略过本章。但是，如果你还拿不准数据库的确切含义，那么本章对于你学习以后的内容将十分重要。

阅读本书的过程就像在一座大楼里参观一个博物馆。首先要从底层开始，爬一些不太长的楼梯，到达后续的每个楼层。各种概念相互依赖，直至你到达足够的高度，可以轻松地把握全局为止。不过本章是一个例外，你可以把本章想象成一驾直升飞机，乘坐它可以直接到达博物馆的入口。你从一个较高的角度来看待所有的事物，并在下降的过程中，快速地经过不同的楼层。当你经过时，楼层上的人可能向你挥手致意，但你最终还是要亲自爬几层楼，才能确切地知道他们为何如此激动。

1.1 人们如何使用数据库

许多人都在谈论数据库。计算机已深入人们的生活，所以往往能听到这样的对话：“我在数据库里查查你的记录。”从这个角度来说，数据库似乎就是一个无底洞，电话公司在此填充所有的客户信息，有时他们还会在这里保存与你有关的其他信息。

好了，如果真的存在这样一个地方，那么从技术上说它应该是一个数据库，一个充满着大量数据的场所。不过，我们在谈到数据库时，还包含另一层意思，那便是数据非常容易查找。除此之外，我们谈论的是计算机化的系统。

1.1.1 计算机化的填报系统

我们之所以让计算机进入自己的生活,是因为可以用它们进行编程,从而自动地完成一些麻烦的、重复性的任务,或者帮助我们做一些不凭借其计算速度或数据存储能力便无法做到的事情。将信息印在纸上,然后将纸归入文件夹中,再对文件柜进行归档整理,这是一个提高精确性的过程,但是,对人们而言,如果这一过程开始变成将电子文档移动到硬盘上的文件夹中,其好处是不言而喻的。

数据库接管了过去对文件进行分类存档的过程;它们引入了一种系统,可对信息进行索引,从而不需要翻阅厚厚的卷宗,经过很长时间才能找到自己需要的信息,这很像老式图书馆采用的卡片目录系统。

并非所有的数据库都一模一样,但它们都遵循一个基本原则,那便是信息应组织成记录。每条记录都是一个固定的字段集合。这些记录要置入一个表中,而同一个数据库可能有几个表。

以一个二手车经销商为例。Ken 是汽车经销商,任何时候他手上拥有的车辆都超过 100 辆。由于车子太多,当有人问起时,他往往记不住库存的车辆有哪些。为此,他决定对信息进行组织,以便利用数据库来使用这些信息。

他的数据库里有一个表,表中包含每辆车的专有信息,包括制造商、型号和使用年限。他还记录自己的批发价,以及他对车子进行的每一次修理。美国的每辆车都有一个车辆标识号(Vehicle Identification Number, VIN),这个号码是独一无二的。Ken 在自己的汽车表里添加了这个字段,以便于区分不同的车辆。

1.1.2 数据库管理系统

Ken 的数据库已经有了一个简单的设计,但他需要一个数据库管理系统(Database Management System, DBMS)来容纳自己的数据库。DBMS 是一个软件系统,负责数据管理的方方面面,它提供了创建表、插入数据、查找数据以及删除数据的途径。

幸运的是,Ken 选择了 MySQL,这是一种出色的方案,因为它很快、很可靠、很便宜。Ken 的利润并不高,所以无法承担企业级解决方案的花费。但是,他也无法承担由于数据库崩溃而造成的损失。因此,他挑选了一种开放源代码的方案,用起来会让他更有信心。除此之外,他注意到 MySQL 邮件列表上提供了丰富的、免费的技术支持。如果他需要有保证的、专业的支持,还可以选择从 MySQL AB 那里购买支持服务。

MySQL 可在硬盘上对记录 and 表进行高效的存储。Ken 不必关心具体的实现细节,他只需执行恰当的命令即可。

和许多数据库一样,MySQL 按照一种客户机/服务器的关系来运行。这个概念描述的是一种网络体系结构,网络上的计算机充当客户机或服务器的角色。通常服务器功能更强,

宗旨是为一组客户机提供服务。处理能力和数据集中在服务器上,客户机是轻量级的,它们是用户获取服务器资源的一个接口。图 1-1 展示了 Ken 的计算机和服务器硬盘驱动器之间的信息流动情况。

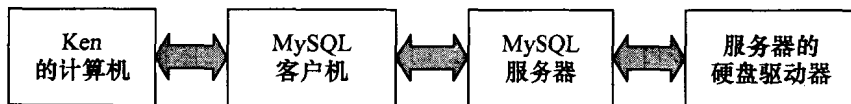


图 1-1 数据流

Ken 运行 MySQL 客户程序,它是一个命令行实用程序,通过网络连接到服务器。当 Ken 执行命令时,服务器会尝试执行它们,大多数时间都在本地硬盘驱动器执行读和写操作。

Ken 喜欢使用命令行实用程序。例如,他经常使用 vi 编辑器。然而,他同时还在运行 MySQL GUI,这是用于同 MySQL 服务器交互和对其进行管理的众多图形化客户机之一。第 3 章讨论了可供选择的客户机。

通过像这样分担工作,Ken 获得了极大的灵活性。目前,他在自己的桌面机上同时运行客户机和服务器,但事实上以后可以非常容易地将服务器放在一台单独的机器上运行。

1.1.3 使用语言

MySQL 使用一种名为结构化查询语言(Structured Query Language, SQL)的语言。有的人喜欢把它读作“sequel”。包括我在内的另一些人则把它分解成字母再念出来,即“esk-cue-ell”。大多数人对这两个名字都明白,但 MySQL 的开发者更喜欢后者,他们将自己的产品念成“my-esk-cue-ell”。把汽车表里的各个字段定义好之后,Ken 必须使用 SQL 在自己的 MySQL 数据库中创建表。首先,他启动 MySQL 外壳程序,这是一个命令行实用程序,然后执行清单 1.1 中的命令。

清单 1.1 创建一个表

```
CREATE TABLE car(  
    VIN VARCHAR(17) NOT NULL,  
    Make VARCHAR(16) NOT NULL,  
    Model VARCHAR(16) NOT NULL,  
    ModelYear INT(4) NOT NULL,  
    WholesalePrice FLOAT(6,2) NOT NULL,  
    Color VARCHAR(8) NOT NULL,  
    Mileage INT(11) NOT NULL,  
    Comments TEXT,  
    PRIMARY KEY(VIN)  
);
```

该命令展示了比我们以前讨论的内容更多的细节,所以我们有必要仔细地研究一下它们。很明显,这是一个用于创建表的命令,表的名字是 car。一系列格式化好的行描述了表中的各个字段,所有这些行都包括在后续的一对圆括号中。

第一个字段名为 VIN,它的类型是 VARCHAR,即 Variable-length Character(变长字符)的缩写形式。它的值最多可容纳 17 个字符,而这正好是美国地区的一个 VIN 的长度。该行以 NOT NULL 结尾,作用是向 MySQL 指出该列必须容纳一些值。MySQL 使用 NULL 来表示不存在一个值,这和空字符串是有区别的。第 6 章将更详细地解释这一概念。

还有其他一些字段,它们用于容纳不同类型的数据。其中,WholesalePrice(批发价)字段容纳着一个 FLOAT 值,即浮点数。在小数点之前,它最多可包含 6 个数位;在小数点之后,最多可包含 2 个数位。ModelYear 字段则是一个 4 位整数。

字段列表的最后一项并非字段定义,而是对一个索引的定义。它将 VIN 字段指定为该表的主键。正如前面说过的那样,Ken 希望 VIN 成为每辆车独一无二的标识符。通过将此字段设为主键,可加强这一限制。如果插入具有相同 VIN 的两辆车,就属于一个错误,是 MySQL 所不允许的。

1.1.4 提问

在数据库领域,我们将命令称为语句或查询。尽管发送给数据库服务器的任何命令都可叫做一个语句,但查询特指能返回信息的语句。查询是向数据库提出的问题,有点像玩“老处女”(Old Maid)游戏。如果想在数据库中查找一条记录,你可能会问:“有没有颜色是白色的记录?”如果数据库有,就会返回这些记录。

清单 1.2 中的查询是一个选择语句,它选择与条件相符的大量记录。这个条件是:颜色字段为白色。然而,它不会返回记录中的所有东西,它只返回 4 个字段:VIN,Make,Model 和 ModelYear。

清单 1.2 搜索白色汽车

```
SELECT VIN, Make, Model, ModelYear
FROM car
WHERE Color= 'White';
```

注意,选择(SELECT)语句读起来和普通的英语句子几乎完全一样。SQL 是一种第 4 代语言(4GL)。根据记载,第 1 代语言是纯粹的机器码,CPU 可直接理解它们。第 2 代语言是汇编语言。第 3 代语言是目前编程用得最多的高级语言,比如 C 和 PHP 等。所以,第 4 代语言是指更像人类自然语言的一种计算机语言。

现在,Ken 可执行清单 1.2 中的查询,或根据需要进行任意变化,但结果都是相同的。

根本没有记录返回,因为他还没有插入任何记录! 所以,事先还得执行一个或多个插入 (INSERT)语句。

正式地说,插入语句并非一个查询。它将信息置入数据库,而不是获取它。平常闲谈时,你可能听见有人提到插入查询。然而,根据 4GL 的标准,插入语句读起来像一个普通的句子。清单 1.3 中的查询可在 car 表里插入一条记录,值的顺序与表内各列的顺序相对应。

清单 1.3 插入记录

```
INSERT INTO car VALUES (  
    '12345678901234567'  
    'Plymouth',  
    'Roadrunner',  
    1969,  
    5500.00,  
    'Blue',  
    148123,  
    'Unrestored'  
);
```

Ken 继续为库存的每辆车执行一个插入语句。之后,他可在选择语句中置入条件,从而对车子进行搜索。如果客户问他库存里是否有一辆 Ford(福特)车低于 10000 美元,他可以执行如清单 1.4 所示的一个查询。

清单 1.4 搜索便宜的 Ford 车

```
SELECT *  
FROM car  
WHERE Make = 'Ford'  
AND WholesalePrice < 9000.00;
```

星号(*)要求返回所有的字段。Ken 指出 1000 美元是他在任何车上的最低利润,所以为批发价使用了一个恰当的数字。

在尝试查找 Ford 车的过程中,Ken 注意到有条记录明显有错。其中一辆车的颜色应该是“White”。为了解决这个问题,Ken 必须执行一个更新查询。在清单 1.5 中,Ken 指出该车的 VIN,并构建了一个更新(UPDATE)语句。

清单 1.5 更新汽车颜色

```
UPDATE car  
SET Color = 'White'  
WHERE VIN = '10203040506070809';
```
