TURING 图灵程序设计丛书 数据库系列

Apress®

Expert PL/SQL Practices
for Oracle Developers and DBAs

Oracle PL/SQL实战

[美] John Beresniewicz
[美] Melanie Caffrey
[美] Dominic Delmolino
[美] Connor McDonald
[乌克兰] Michael Rosenblum
[英] Adrian Billington
[美] Ron Crisco
[印度] Sue Harper
[美] Arup Nanda
[美] Robyn Sands
[瑞士] Martin Büchi
[美] Lewis Cunningham
[丹] Torben Holm
[瑞士] Stephan Petit
[美] Riyaj Shamsudeen
著

卢涛 译
苏旭晖 李颖 贾书民 审

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 数据库系列

Expert PL/SQL Practices
for Oracle Developers and DBAs

Oracle PL/SQL实战

[美] John Beresniewicz
[美] Melanie Caffrey
[美] Dominic Delmolino
[美] Connor McDonald
[乌克兰] Michael Rosenblum
[英] Adrian Billington
[美] Ron Crisco
[印度] Sue Harper
[美] Arup Nanda
[美] Robyn Sands
[瑞士] Martin Büchi
[美] Lewis Cunningham
[丹] Torben Holm
[瑞士] Stephan Petit
[美] Riyaj Shamsudeen
著

卢涛 译
苏旭晖 李颖 贾书民 审

人民邮电出版社
北京

图书在版编目（CIP）数据

Oracle PL/SQL实战 / (英) 比林顿(Billington, A.) 等著 ; 卢涛译. -- 北京 : 人民邮电出版社, 2012.11
(图灵程序设计丛书)
书名原文: Expert PL/SQL Practices: for Oracle Developers and DBAs
ISBN 978-7-115-29485-2

Ⅰ. ①O… Ⅱ. ①比… ②卢… Ⅲ. ①关系数据库－数据库管理系统 Ⅳ. ①TP311.138

中国版本图书馆CIP数据核字(2012)第232411号

内 容 提 要

本书共15章，分别由15位业内顶级专家担纲撰写。一条SQL语句的误用，可能导致作业的运行时间增加几百倍，Riyaj Shamsudeen巧妙地回答了何时使用PL/SQL这一问题；Michael Rosenblum说明了如果只有到最终运行时，才知道所要运行的SQL语句到底是什么，那该如何编写代码；Dominic Delmolino介绍了用PL/SQL进行并行处理的方法，以及这样能获得的益处和产生的额外负担。本书还有更多精彩内容等待读者一探究竟。

本书适合有一定PL/SQL基础的读者学习。

图灵程序设计丛书

Oracle PL/SQL实战

◆ 著 [美] John Beresniewicz [英] Adrian Billington
[瑞士] Martin Büchi [美] Melanie Caffrey
[美] Ron Crisco [美] Lewis Cunningham
[美] Dominic Delmolino [印度] Sue Harper
[丹] Torben Holm [美] Connor McDonald
[美] Arup Nanda [瑞士] Stephan Petit
[乌克兰] Michael Rosenblum [美] Robyn Sands
[美] Riyaj Shamsudeen
译 卢 涛
审 苏旭晖 李颖 贾书民
责任编辑 王军花
执行编辑 李 静

◆ 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
三河市海波印务有限公司印刷

◆ 开本：800×1000 1/16
印张：27.25
字数：650千字 2012年11月第1版
印数：1－3 500册 2012年11月河北第1次印刷
著作权合同登记号 图字：01-2011-8034号
ISBN 978-7-115-29485-2

定价：89.00元
读者服务热线：(010)51095186转604 印装质量热线：(010)67129223
反盗版热线：(010)67171154

译 者 序

2010年，我参与编写了一本有关Oracle开发的书《剑破冰山：Oracle开发艺术》，这本书的重点是SQL编程，因此曾想过再写一本关于PL/SQL开发的书。2011年10月，在《剑破冰山：Oracle开发艺术》上市大约1周年的时候，我在图灵公司的网站发现了本书英文版正在征集译者，看了下目录，感觉它提到了很多我想表达的内容，而且比较全面、系统。于是，就萌生了翻译的想法。本书由15位作者共同编写，每人负责一章，所以每章都能独立成篇。而把各章内容结合到一起，又基本上涵盖了PL/SQL开发的各个方面。需要指出一点，本书不是PL/SQL入门教材，读者需要有一定PL/SQL基础。实际上，本书大部分内容在Oracle官方的培训计划中属于高级课程*Oracle Database 11g: Advanced PL/SQL*的范畴。

决定挑战这个任务后，我通过图灵副总经理谢工联系到了傅志红副总经理，有幸通过了试译。这本书英文原作有500多页，加上我之前还没有翻译过这么大规模的文字，所以觉得一人翻译在时间上有些吃紧，因此想邀请上本书的几位合著者共同翻译。不过由于各种原因没有成功，最终我还是独自承担了这个任务，在5个月的时间里，完成了初译、复查、修改等工作。

如果没有大家的帮助，我一个人无法完成这本译作。

首先感谢老战友苏旭晖（ITPUB Oracle开发版版主，网名newkid），他现定居加拿大多伦多，从事数据库应用系统的设计与开发工作。虽未见面，但他在我翻译的5个多月时间里，一直不辞辛劳地帮助我解决多处语言和技术难点，并给出了更好的译法。更难得的是，对原书个别不合理的叙述，提出了中肯的意见，使读者不仅能通过本书了解原作者的观点，还能引发进一步的思考。

其次感谢我妻子李颖，她是英语专业毕业生，作为本书的第一读者，帮助我检查出很多生硬的译文，并把它们修改得更加通顺。得益于她的修改，本书才有现在的可读性。同时，她还承担了培养教育孩子的重任，使我可以专心翻译本书。

还要感谢老大哥贾书民，他有10多年的Oracle开发经验，做过许多基于Oracle的大型项目，他通读了我的译文，提出了许多中肯的修改建议。在我试译时，他也帮助我对译文做了很多修改，对我顺利通过试译起到了很大的作用。

感谢图灵公司李鑫编辑、李静编辑、李松峰主任，他们对译文进行把关，并对语言进行润色，使表达更加地道。

最后希望这本书能帮助读者提高Oracle PL/SQL开发水平。由于译者经验和水平有限，译文中难免有不妥之处，恳请读者批评指正！

序

作为图书编辑，我一般满足于在幕后做好自己的编辑工作，很少站到前台来为我负责编辑的图书作序。但是，这次我却破例了。因为我也曾做过开发，这本书的内容勾起了我的很多回忆。

本书旨在教你如何有效地使用PL/SQL。它不是描述语法的书，而是介绍如何把语法及其特性与良好的开发实践相结合，创建更具有可靠性、可扩展性的应用，并使之具备长期可维护性。

不管使用什么工具，首先需要明白的是在什么情况下使用它比较合适。在本书的开篇章节“避免误用”中，Riyaj Shamsudeen巧妙地回答了何时使用PL/SQL这一问题。我把这一章放在本书的开始，也是因为我个人的经历：那是在20世纪90年代末，我进行了最成功的一次性能优化，那次我把客户端PC上的一堆过程代码改写为一个SQL语句以后，一个作业的运行时间从超过36小时缩短到仅仅几分钟。虽然PL/SQL不是导致性能问题的罪魁祸首，但是我得到了一条经验——基于集合的方法（如果有可能）通常优于过程性代码。

Michael Rosenblum紧接着写了动态SQL那一章，其内容相当精彩。他说明了如果只有到最终运行时才知道所要运行的SQL语句到底是什么，那么该如何编写代码。这使我想起20世纪90年代初的一段经历，那时我在陶氏化学公司利用Rdb的扩展动态游标功能集，为一个医疗记录系统写数据装载应用程序。我感觉那是我开发过的最有趣的应用程序之一。

Dominic Delmolino介绍了用PL/SQL 进行并行处理的方法，以及这样能获得的益处和产生的额外负担。谨小慎微，永不为过！我在当数据库管理员时犯下的最大错误之一就是，有一次头脑一热，在一个关键的应用表上设置了一个并行度，其目的是为了想让其中的一个报表程序运行得更快。可是，事与愿违。好像键盘上的回车键连着我的电话似的，修改后大约不到一分钟，我的电话铃就响了，电话线另一端的主管对我的修改非常不满。无须多言，从此我就决定在实施并行前一定要深思熟虑。Dominic写的那章可以帮助我们避免陷入这样的窘境。

本书的多章内容涵盖了代码规范和极其实用的编程经验。Stephan Petit向我们介绍了一套实用的命名和编码规范。Torben Holm讲述了PL/SQL警告和条件编译。Lewis Cunningham阐述了代码分析，使我们认识到真正理解自己所写的代码及其工作原理的重要性，发人深思。Robyn Sands的“渐进式数据建模”一章，令我们对渐进式数据模型的良好设计及灵活性有了更多的思考。Melanie Caffrey介绍了经常使用的各种游标类型，帮助我们在不同条件下正确地选择游标。

其他章节介绍如何进行调试和故障排除。Sue Harper介绍了PL/SQL单元测试，特别是一些如今已经集成在SQL Developer中的功能集（这使我想起了在纸上写单元测试脚本的日子），它可以帮助避免犯回归错误。利用自动化单元测试，能够容易且方便地验证程序，以确保自己不会在

修复一个错误的同时制造了更多的错误。

还有John Beresniewicz介绍“合同导向编程”那一章。John给出的方法的一个关键部分是，在代码的各种特定地方使用断言来验证条件。记得我首次了解断言技术是在做PowerBuilder编程时，那已经是很久很久以前的事了。我很高兴看到John把这项技术和PL/SQL联系起来，将其发扬光大。

Arup Nanda的论述能够帮助我们控制依存和失效问题。依存问题可能会导致发生类似随机的、难以重现的应用程序错误。Arup展示了如何完全掌控那些必将发生的事情，这样你才不会落入意外错误的陷阱。

一般情况下，我们不得不考虑性能和扩展性。Ron Crisco告诉我们通过剖析代码，找到尽可能优化代码的方法。Adrian Billington讨论了从SQL语句中调用PL/SQL的性能问题。Connor McDonald论述了批量SQL操作惊人的性能优势。

关于可扩展性，通常会被遗漏的一面是应用程序的规模和参与开发的人数。PL/SQL是否适合数十到数百位程序员的大规模开发？Martin Büchi在“大规模PL/SQL开发”那章，描述了他在管理由170多位开发者维护的具有1100万行代码的应用中的成功经验，说明了PL/SQL是非常适合于这种任务的。

不难看出我为这本书感到兴奋。作者都是顶级专家，分别介绍了自己热衷的并且特别在行的PL/SQL的某个方面。如果你已经学习了语法，那么坐下来读一读这本书，充分利用PL/SQL和Oracle数据库的强大功能，投身到开发应用程序的事业中来吧！

Jonathan Gennick

Apress编辑主任助理

作者简介

- John Beresniewicz（约翰·贝雷斯尼维奇）是位于加州红木城红木岸（Redwood Shores）的Oracle总部技术团队的一名咨询顾问。他于2002年加入Oracle，负责企业管理器的数据库性能领域，他对诊断和调优包、实时应用测试、支持工作台和Exadata的设计作出了重要贡献。多年以来，他经常在Oracle全球大会和其他会议上发言，发言主题包括数据库性能和PL/SQL编程。他与Steven Fellerstein合著了*Oracle Built-in Packages*（O'Reilly & Associates，1998年）一书，并且是OakTable网络的创始人之一。
- Adrian Billington（阿德里安·比林顿）是应用设计、开发和性能调优方面的顾问。自1999年以来，一直从事Oracle数据库方面的工作。他是www.oracle-developer.net网站的发起人，这个网站为Oracle开发人员提供各种SQL和PL/SQL功能、实用工具和技术。阿德里安还是Oracle ACE，同时也是OakTable网络的成员。现在，他与妻子安吉和三个孩子：格鲁吉亚、奥利弗和伊莎贝拉一起居住在英国。
- Martin Büchi（马丁·步琪）自2004年以来，任Avaloq公司首席软件架构师。该公司是一个标准化的银行软件供应商，其产品基于Oracle RDBMS构建，包含1100万行PL/SQL代码。他与两位同事一起设计了系统架构，并评审了170名全职PL/SQL开发人员的设计和代码，以追求软件的简明、效率和健壮性。马丁经常在Oracle大会上发言。2009年，他被*Oracle Magazine*评选为PL/SQL年度开发人员。从事Oracle数据库工作之前，马丁曾在面向对象的系统、形式化方法和近似记录匹配等领域工作。他拥有瑞士联邦技术研究所的硕士学位和芬兰土尔库计算机科学中心的博士学位。业余时间，马丁喜欢与他的家人一起进行各种户外运动。
- Melanie Caffrey（梅拉妮·卡弗里）是Oracle公司高级开发经理，为不同客户的业务需求提供前端和后端的Oracle解决方案。她是多部技术出版物的合著者，包括*Oracle Web Application Programming for PL/SQL Developers*、*Oracle DBA Interactive Workbook*、*Oracle Database Administration: The Complete Video Course*等，这些书全部由Prentice Hall出版。她在纽约哥伦比亚大学的计算机技术与应用课程中指导学生，教授先进的Oracle数据库管理和PL/SQL开发。她也经常在Oracle会议上发言。
- Ron Crisco（罗恩·克里斯科）28年来分别担任软件设计师、开发人员和项目负责人，并有21年的Oracle数据库工作经验。他在R方法（Method R）公司从事软件设计和开发、软件产品管理（如R方法剖析器、MR工具和MR跟踪）、咨询、教授课程等工作。他的特长

是简化复杂的工作，这在帮助他身边的人完成非凡工作时尤显宝贵。

- Lewis Cunningham（刘易斯·坎宁安）在IT领域已经工作了20多年。自1993年以来一直与Oracle数据库打交道。他的专长是应用程序设计、数据库设计，以及大容量的VLDB数据库编码。目前他任职于佛罗里达州圣彼得堡的一家金融服务公司，担任高级数据库架构师，负责超大规模、高事务率分析型数据库和应用程序的工作。他花了大量时间来与最新的技术和趋势保持同步，并在用户组发表演讲，举办网络研讨会。刘易斯也是一位Oracle ACE总监和Oracle认证专家。他在Oracle技术网发表了数篇文章，并在http://it.toolbox.com/blogs/oracle-guide维护一个Oracle技术博客。刘易斯写了两本书：*EnterpriseDB: The Definitive Reference*（Rampant Tech press，2007年）和*SQL DML: The SQL Starter Series*（CreateSpace，2008年）。他与他的妻子及两个儿子起住在佛罗里达州。可以通过电子邮件lewisc@databasewisdom.com与他联系。
- Dominic Delmolino（多米尼克·德莫里诺）是Agilex技术公司首席Oracle和数据库技术专家，这是一家专门协助政府和私营企业实现信息价值的咨询公司。多米尼克拥有24年以上的数据库经验，其中担任过20多年的Oracle数据库工程和开发专家。他是OakTable网络的成员，并定期出席各种学术会议、研讨会，以及欧洲和美国的用户组会议。他还维护www.oraclemusings.com网站，该网站专注于与数据库应用程序开发相关的数据库编码和设计实践。多米尼克拥有纽约州伊萨卡康奈尔大学的计算机科学学士学位。
- Sue Harper（苏·哈珀）是数据库开发工具组中的Oracle SQL Developer和SQL Developer数据建模器的产品经理。她自1992年以来一直在Oracle公司工作，目前在伦敦办事处工作。苏是一些杂志的特约撰稿人，维护着一个技术博客，并在世界各地的许多会议上发言。她撰写了技术书籍*Oracle SQL Developer 2.1*（Packt，2009），业余时间，苏喜欢步行和摄影。同时，她还花时间到新德里的贫民区做慈善工作，帮助那里的妇女和儿童。
- Torben Holm（托尔·霍尔姆）自1987年以来一直从事开发工作。自1992年以来，他一直致力于与Oracle相关的工作，前四年担任系统分析师和应用程序开发人员（Oracle 7、Forms 4.0/Reports 2.0和DBA），然后做了两年开发（ORACLE6/7、Forms 3.0和RPT以及DBA）。他在Oracle丹麦公司的高级服务组工作了数年，担任首席高级顾问，执行应用程序开发和DBA任务。他还担任过PL/SQL、SQL和DBA课程的讲师。现在，托尔在Miracle A/S（www.miracleas.dk）工作，担任顾问，负责应用开发（PLSQL、`mod_plsql`、Forms、ADF）和数据库管理。10年来他一直在Miracle A/S公司工作。他是Oracle认证开发人员，并且也是OakTable网络成员。
- Connor McDonald（康纳·麦当劳）自20世纪90年代初一直从事Oracle相关工作，他非常熟悉Oracle 6.0.36和Oracle 7.0.12。在过去11年中，康纳曾为位于西欧、东南亚、澳大利亚、英国和美国的公司开发过系统。他已经认识到，虽然世界各地的系统及方法非常多样，但开发在Oracle上运行的系统往往有两个共同的问题：要么避免使用Oracle特定的功能，要么就是采取不太理想的用法或随意乱用它们。正是这种观察，促使他创建了一个提示和技巧的个人网站（www.oracledba.co.uk），并努力在Oracle演讲者组织中发表更多演讲，

以提高PL/SQL的业内认知度和普及度。

- Arup Nanda（奥雅纳·南大）自1993年以来，一直是Oracle DBA，他熟悉数据库管理的所有方面，从建模到灾难恢复。目前，他在纽约州白原市的喜达屋酒店（即喜来登、威斯汀等连锁酒店的母公司）领导全球DBA团队。他是独立Oracle用户协会（IOUG）旗下出版物*SELECT Journal*的特约编辑，在许多Oracle技术盛会，如Oracle全球和本地用户组（如纽约Oracle用户组）中发表演讲，并为印刷出版物如*Oracle Magazine*和网络出版物如*Oracle Techndogy Network*撰写了许多文章。奥雅纳与他人合著了两本书：*Oracle Privacy Security Auditing*（Rampant，2003年）和*Oracle PL/SQL for DBAs*（O'Reilly，2005年）。由于他的专业成就和对用户社区的贡献，Oracle评选他为2003年年度DBA。奥雅纳与他的妻子Anindita和儿子阿尼什住在康涅狄格州的丹伯里。可以通过arup@proligence.com联系他。
- Stephan Petit（斯蒂芬·佩蒂特）于1995年在位于瑞士日内瓦的欧洲粒子物理实验室（CERN）开始了他的职业生涯。他现在是一个软件工程师和学生团队的负责人，负责为实验室和其他部门提供应用程序和工具。工程和设备数据管理系统是这些工具之一，也称为CERN EDMS。像CERN的大型强子对撞机（LHC）项目有40年或以上的生命周期。EDMS是实验室的数字化工程的内存/记忆体。电子文件管理系统中存储了与一百多万件设备有关的一百多万份文件，EDMS也供CERN的产品生命周期管理（PLM）和资产跟踪系统使用。 EDMS几乎完全是基于PL/SQL的，并旨在拥有一个至少与LHC一样长的生命周期。

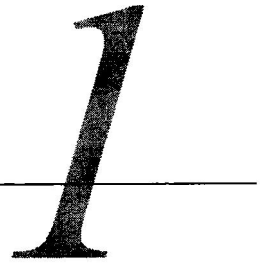
  斯蒂芬和他的团队一直在完善PL/SQL编码规范和最佳实践，以满足他们非常有趣的各种挑战的组合：几十年的可维护性、可靠性、高效的错误处理、可扩展性、模块的可重用性。团队成员的频繁轮换，其中大部分只是暂时在CERN实习的学生，加剧了这些挑战。最古老的一段代码是在1995年写的，现在仍然在使用——并且成功地运行！除了完善PL/SQL，斯蒂芬还喜欢不时登台表演，比如担任CERN摇滚夏季音乐节的摇滚乐队歌手，以及在多部戏中出演角色。
- Michael Rosenblum（迈克尔·罗森布鲁姆）是Dulcian公司的软件架构师/开发DBA，他负责系统调优和应用程序架构。迈克尔通过编写复杂的PL/SQL例程和研究新功能支持Dulcian开发人员。他是*PL/SQL for Dummies*（Wiley，2006年）一书的作者之一，并在IOUG *Select Journal*和ODTUG *Tech Journal*发表了许多篇与数据库相关的文章。迈克尔是一位Oracle ACE，也经常出席不同地区和国家的Oracle用户组大会（Oracle OpenWorld大会、ODTUG、IOUG Collaborate、RMOUG、NYOUG等），他是ODTUG万花筒2009年“最佳演讲奖”得主。在他的家乡乌克兰，他获得了乌克兰总统奖学金，并拥有信息系统理学硕士学位并以优异成绩获得基辅国立经济大学毕业证书。
- Robyn Sands（罗宾·桑兹）是思科系统公司的软件工程师，她为思科的客户设计和开发嵌入式Oracle数据库产品。自1996年以来，她一直使用Oracle软件，并在应用开发、大型系统实现和性能测量方面具有丰富经验。罗宾的职业生涯始于工业工程和质量工程，她

将自己对数据的挚爱结合到以前接受的教育和工作经验中，寻找新方法来建立性能稳定、易于维护的数据库系统。她是OakTable网络成员，并是下面两本Oracle书籍的作者之一：*Expert Oracle Practices*和*Pro Oracle SQL*（都由Apress出版，2010）。罗宾偶尔在http://adhdocddba.blogspot.com发表一些博客。

- Riyaj Shamsudeen是OraInternals公司首席数据库管理员和主席，这是一家从事性能调优/数据库恢复/EBS11*i*等领域的咨询公司。他专门研究真正的应用集群（RAC）、性能调优和数据库内部结构。他还经常在其博客http://orainternals.wordpress.com上发表这些技术领域的文章。他也经常出席许多国际会议，如HOTSOS、COLLABORATE、RMOUG、SIOUG、UKOUG等，他是OakTable网络的骄傲一员。他拥有16年以上使用Oracle技术产品的经验，并担任了15年以上的Oracle/Oracle应用程序数据库管理员。

目　录

第1章 避免误用



Riyaj Shamsudeen

感谢您购买本书。PL/SQL是你值得拥有的好工具，不过，你应当理解PL/SQL并不适用于所有场景。本章将讲述何时使用PL/SQL编写应用程序，怎样编写可扩展的代码，更重要的是，何时不使用PL/SQL编写代码。滥用某些PL/SQL的结构会导致不可扩展的代码。本章将讨论大量的案例，它们因误用PL/SQL而不可扩展。

PL/SQL 与 SQL

SQL是一门集合处理的语言，如果以集合级别的思维编写SQL语句，则会使其具有更好的可扩展性。PL/SQL是过程语言，而SQL语句可以嵌入到PL/SQL代码中。

SQL语句在SQL执行器（也就是通常说的SQL引擎）中执行。PL/SQL语句则由PL/SQL引擎执行。PL/SQL的威力在于能将PL/SQL的过程化数据处理能力和SQL的集合处理能力结合起来。

1.1 逐行处理

在一个典型的逐行处理程序中，代码首先打开一个游标，然后遍历从游标返回的行，并且处理每一行数据。这种基于循环的处理方式是非常不赞成使用的，它会导致不可扩展的代码。代码清单1-1显示了一个使用这种结构的例子。

代码清单1-1 逐行处理

```
DECLARE
  CURSOR c1 IS
    SELECT prod_id, cust_id, time_id, amount_sold
    FROM sales
    WHERE amount_sold > 100;
  c1_rec c1%rowtype;
  l_cust_first_name customers.cust_first_name%TYPE;
  l_cust_lasT_name customers.cust_last_name%TYPE;
BEGIN
  FOR c1_rec IN c1
```

```
  LOOP
    -- 查询客户的详细信息
    SELECT cust_first_name, cust_last_name
    INTO l_cust_first_name, l_cust_last_name
    FROM customers
    WHERE cust_id=c1_rec.cust_id;
    --
    -- 插入数据到目标表
    --
    INSERT INTO top_sales_customers①(
      prod_id, cust_id, time_id, cust_first_name, cust_last_name,amount_sold
      )
      VALUES
      (
        c1_rec.prod_id,
        c1_rec.cust_id,
        c1_rec.time_id,
        l_cust_first_name,
        l_cust_last_name,
        c1_rec.amount_sold
      );
  END LOOP;
  COMMIT;
END;
/

PL/SQL procedure successfully completed.

Elapsed: 00:00:10.93
```

在代码清单1-1中，程序声明了一个游标c1，然后用游标for循环隐式地打开了这个游标，对从游标c1取出的每一行，程序查询customers表，并把first_name和last_name的值填充到变量，随后插入一行数据到top_sales_customers表。

代码清单1-1的编程方法很有问题。即使在循环中调用的SQL语句是高度优化的，程序的执行还是会消耗大量时间。假设查询customers表的SQL语句消耗0.1秒，INSERT语句也消耗0.1秒，那么在循环中每次执行就要0.2秒。如果游标c1取出了100 000行，那么总时间就是100 000乘以0.2秒，即20 000秒，也就是大约5.5小时。很难去优化这个程序的结构。基于显而易见的理由，Tom Kyte把这种处理方式定义为慢之又慢的处理（slow-by-slow processing）。

注意　本章的例子采用SH模式（schema），这是Oracle公司提供的范例模式之一。要安装这个范例模式，Oracle提供了安装软件。可以从http://download.oracle.com/otn/solaris/oracle11g/R2/solaris.sparc64_11gR2_examples.zip下载Solaris平台11gR2版本的样例软件。软件解压缩后的目录中含有安装说明。Oracle网站也提供了其他平台和版本的范例软件的Zip压缩包。

代码清单1-1的代码还有一个固有的问题。从PL/SQL的循环中调用的SQL语句会反复在PL/SQL引擎和SQL引擎之间切换执行，这种两个环境之间的切换称作上下文切换。上下文切换

① 这里的top_sales_customers表需要有6个列。——译者注（以下如未做特殊说明的均为译者注）