<<Oracle PL/SQL实战>>

图书基本信息

书名：<<Oracle PL/SQL实战>>

13位ISBN编号：9787115294852

10位ISBN编号：7115294852

出版时间：2012-11

出版时间：人民邮电出版社

作者：[美] John Beresniewicz,[英] Adrian Billington,[瑞士] Martin B ü chi,[美] Melanie Caffrey,[美] Ron Crisco,[美] 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 S

页数：166

字数：650000

译者：卢　涛

前言

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 B ü chi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 GennickApress编辑主任助理

<<Oracle PL/SQL实战>>

内容概要

本书由15位知名技术专家联手打造，每位作者分别用一章的篇幅介绍他们最擅长的PL/SQL相关主题，涵盖了PL/SQL开发的方方面面。
本书作者要么是Oracle社区中坚分子，要么是大名鼎鼎的OakTable成员，而且经常活跃在Oracle技术培训第一线，对PL/SQL均有着深入透彻的理解，对解释复杂问题有着简单独到的方法。
一册在手，众多PL/SQL牛人的真知灼见尽收眼底，你还等什么？

本书着重介绍了PL/SQL最新、最实用的特性，从什么该做和什么不该做、怎么做对，以及怎么做更有效率、效果更好等三个部分全面阐释了PL/SQL相关的各个主题。
而且，每一章都配有非常贴切的示例代码、跟踪图以及输出结果，辅以深入浅出的讲解，令人在恍然大悟之后不禁拍案叫绝。
各章内容均涵盖了PL/SQL实际开发中的最佳实践，反映了作者多年积累的经验和教训，其价值非同一般。

本书适合具有一定PL/SQL经验的读者学习参考。

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 Bü chi（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/EBS11i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。

译者简介：
卢涛
1995年参加工作，高级程序员、系统分析师、高级工程师。
2004年起接触Oracle数据库，获得Oracle数据库管理9i至11g、PL/SQL开发、性能优化、RAC管理、数据仓库等多个OCP、OCE、OCS认证。

ITPUB社区Oracle开发版版主，《剑破冰山——Oracle开发艺术》一书合著者。
做过需求分析、系统分析、架构设计、数据库和应用程序性能优化等工作。
参与数次全国性普查数据处理系统的设计、开发和运维。

校审
苏旭晖
在IT行业摸爬滚打20多年，1997年至今一直致力于Oracle开发，擅长用SQL、PL/SQL。
ITPUB社区Oracle开发版版主，以及《剑破冰山——Oracle开发艺术》一书合著者。
现居加拿大多伦多，从事数据库应用系统的设计与开发工作。
出国前任职于厦门巨龙软件公司。
多年来致力于金融、公安、医疗、社保等行业软件的研发。

李颖
2006年毕业于渤海大学英语系，主修英语教育专业。
英语爱好者

贾书民
河北省统计局数据管理中心调研员，高级程序员、高级工程师。
有多年Unix系统管理、数据库系统管理与设计、数据处理软件开发经验。
1993年起开始使用Oracle开发了《河北省综合数据库系统》等多个大型应用系统。
参与过《第五次全国人口普查数据处理系统》等多个国家级项目的设计和开发。
擅长使用SQL、PL/SQL进行高效数据分析处理。
所开发项目获第四届全国统计科学技术进步一等奖。
此外，他还参与编写《剑破冰山——Oracle开发艺术》一书。

<<Oracle PL/SQL实战>>

书籍目录

章节摘录

 插图： 如果计算机可以自动计算出将问题划分成更小部分的方法，并以那些更小的部分为基础并行地运行程序，事情就会很美好。
关于并行的自动识别已经有大量的研究，但事实证明，计算机真的很难理解在哪里有机会使用并行。
虽然已经有人尝试编写新的编译器，使它可以读取高层次的编程语言，并为CPU产生显式并行指令流，但没有太多在这方面很成功的例子，结果是我们仍需要手动设计我们自己的并行处理策略。
好消息是，正确实施并行处理技术，除简单加速之外，还有以下几个好处。
更好的系统利用率，在多处理器系统中，所有处理器都可以应用于任务。
消除一个全有或全无（all—or—nothing）的过程，允许部分完成以防一些子任务失败。
能够在一个固定的时间处理越来越大的数据集，而不是简单地加快速度，以便在一个更短的时间内完成一个固定的数据集。
结果证明，虽然设计和实现并行程序会产生一些开销并需要辛勤的工作，但这些付出同它带来的好处相比往往是值得的。
3.2影响并行处理的定律 用计算机解决问题的技术（或算法）是由计算机科学家研究和编纂的，往往导致所谓的计算定律。
这些定律与计算机体系结构的趋势和过去几年的数据增长互相作用，影响了我们对并行处理的思考。
摩尔定律（Moore's Law，1965年提出）指出，集成电路上可以容纳的晶体管数量每两年增加一倍，并暗示计算机微处理器的性能和速度也每两年增加一倍。
从20世纪80年代中期到大约2004年，由于处理器频率的同步提升，这种加速一直与这个定律吻合。
然而，从那以后，功耗开始成为问题，导致多核处理器的出现，它需要并行处理才能充分利用这条定律。
阿姆达尔定律（Amdahl's Law，1967年提出）指出，使用多个处理器并行计算对程序进行加速受限于程序中的串行片段所需的执行时间。
阿姆达尔定律中存在一个并行处理的寒蝉效应，因为它指出，程序的执行速度不可能比它们的串行组成部分的速度更快。
这意味着利用并行处理，你的程序的速度增加量有一个限制，往往使得并行程序最多只能比与它等价的串行程序快10～20倍。
古斯塔夫森法则（Gustafson's Law，1988年提出）指出，具有非常大的重复数据集的问题可以有效地并行化，几乎是高度并行化（因此，拥有非常小的串行部分并很容易分解为小型独立单位的问题，被正式命名为高度并行）。
这些法则之间的矛盾关系符合21世纪前十年另一个行业的发展趋势，即所谓“ 大数据 ”的崛起。

媒体关注与评论

“ 本书作者阵容强大，内容全面实用，期待更多这种形式的技术书。
”——读者评论

<<Oracle PL/SQL实战>>

编辑推荐

汇集Oracle专家集体智慧的结晶，包罗PL/SQL的方方面面介绍语法的同时展示良好的编程实践内容贴近实战、讲解透彻精彩，每章一个作者，独立成篇，易于理解结合PL/SQL语法特性与良好的开发实践，DBA修炼内力必读秘籍

版权说明