

Oracle Core

Essential Internals for DBAs and Developers

Oracle核心技术

- Oracle资深专家Johnathan Lewis最新力作
- DBA和开发者必须了解的Oracle核心机制
- 一本令人震撼、让你大开眼界的必读经典

[英] Jonathan Lewis 著
杨志洪 熊军 耿永辉 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 数据库系列

Oracle Core

Essential Internals for DBAs and Developers

Oracle核心技术

[英] Jonathan Lewis 著
杨志洪 熊军 耿永辉 译

人民邮电出版社

北 京

图书在版编目 (C I P) 数据

Oracle核心技术 / (英) 路易斯 (Lewis, J.) 著 ;
杨志洪, 熊军, 耿永辉译. — 北京 : 人民邮电出版社,
2013. 12

(图灵程序设计丛书)

书名原文: Oracle core:essential internals for
DBAs and developers

ISBN 978-7-115-33564-7

I. ①O… II. ①路… ②杨… ③熊… ④耿… III. ①
关系数据库系统 IV. ①TP311.138

中国版本图书馆CIP数据核字(2013)第262738号

内 容 提 要

本书揭示了 Oracle 数据库内部的核心信息, 这些信息恰好是每个 DBA 排除故障所必需的, 不多不少。内容涉及 Oracle 核心架构基础: 多少开销是合理的? 如何认识到你做得太多? 如何预见故障? 为什么发生这些故障? 等等。

本书适合 DBA、QA、架构师、开发人员等人群阅读。

-
- ◆ 著 [英] Jonathan Lewis
 - 译 杨志洪 熊 军 耿永辉
 - 责任编辑 李 瑛
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 15.25
 - 字数: 369千字 2013年12月第1版
 - 印数: 1-3 500册 2013年12月北京第1次印刷
 - 著作权合同登记号 图字: 01-2011-8035号
-

定价: 59.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

站在巨人的肩上
Standing on Shoulders of Giants



www.ituring.com.cn

版 权 声 明

Original English language edition, entitled *Oracle Core: Essential Internals for DBAs and Developers* by Jonathan Lewis, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2011 by Jonathan Lewis. Simplified Chinese-language edition copyright © 2013 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L. P.授权人民邮电出版社独家出版。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。

专家推荐

“本书中的分析丰富独到，可帮助各个层次的读者生动直观地理解Oracle数据库，帮助Oracle工程师高效地开展性能诊断和分析工作，也可帮助数据库开发人员更好地设计稳定的Oracle数据库应用程序。”

陆进，中国太平洋保险（集团）股份有限公司首席数据库专家

“从入门到成为Oracle DBA，在此过程中我们对Oracle数据库的看法与理解不断改变，也逐渐感知到Oracle数据库内核机制的博大精深。本书揭示的Oracle数据库内核机制，为在通往Oracle数据库大师殿堂路上的DBA们提供了一盏指路明灯。”

林志云，福建移动运行维护室经理、中国第一批OCM（2003年）

“如果你想深入了解Oracle性能问题，而不仅仅是OWI揭示的那点事，那么本书非读不可！本书对那些零碎的知识点进行了系统的整合。我读完本书都惊呆了，完全没料到它阐述得如此全面！”

诸超，唯品会技术总监、著名数据库专家

“本书作者Jonathan Lewis是世界级的Oracle数据库专家，他的每一本书都是经典，毫无疑问，本书也不例外。”

胡昆，中通服网络信息技术有限公司总工程师

“本书是我见过的Oracle数据库内部原理写得最好的一本，没有之一。虽然篇幅不长，但字字珠玑，很多地方都写得极其深入，值得反复回味。相信本书一定能给所有Oracle数据库从业人员带来不一样的体验。”

崔华，Oracle ACE、中航信资深Oracle数据库专家、《基于Oracle的SQL优化》作者

“Jonathan以清晰明了的语言阐述了Oracle这个复杂的话题。除了深入介绍技术细节，本书还介绍了Oracle内核机制的利弊，帮助你可以更好地规划和优化系统。”

Tanel Poder, 世界顶尖Oracle性能专家、Oracle ACE Director、《深入理解Exadata》作者

“本书填补了Oracle官方性能优化指南文档的关键空白:遇到通过OWI诊断仍无法确定的性能问题该怎么办?不会有一招制胜的捷径,终极解决方案是去理解导致Oracle数据库产生此类问题的原因。”

Charles Hooper, 拥有30年DBA经验, OakTable Network荣誉成员、
*Expert Oracle Practices*作者

推荐序一

我们为什么学习Oracle? 也许是出于兴趣, 也许是为了解决工作中的问题, 每个人都有不同的初衷。我是这样开始的: 1997年刚参加工作的时候, 我们几个毫无数据库经验的年轻人被赶鸭子上架, 面对着一堆7.2.3版本的Oracle数据库, 天天枕戈待旦却无处下手。我是个急性子, 有时候看着系统处于各种状态就自行冲上去一阵乱来, 居然还真搞定了几次。没想到, 你越是敢动手, 别人越是愿意给你动手和学习的机会, 我的DBA生涯就是这样开始的。1999年, 一次偶然的机会, 我在毫无准备的情况下参加了OCP考试, 尽管没过但是只差一点点, 这给了我很大的信心, 我对Oracle产生了更加浓厚的兴趣, 从此开始更加努力地学习。有一段时间, 我工作时面对Oracle, 吃饭时思考Oracle, 下班时学习Oracle, 每天只要一有空, 脑子里想的几乎都是Oracle技术问题。2003年, 中国移动组织了全国Oracle DBA大比武, 我很高兴, 也幸运地成为了4个OCM之一。一晃又是10年, 我从Oracle数据库起步, 在工作中逐步接触了主机、网络、中间件和应用, 团队也从一两个人、三四条枪发展到了一个连。一路走来, 感触良多, 主要有以下几点体会与建议。

1. 不怕起点低, 白纸正好做文章。只要你有兴趣、有恒心, 就一定会有收获。
2. 一定要明白技术是为业务服务的。有技术但产生不了业务价值, 基本等同于没有技术, 而这对于技术人员来说是非常不利的。
3. 要想在技术领域不断进步, 就需要不断地学习, 同时借鉴他人的经验。一本好书就像一盏明灯照亮我们前进的道路, 读一本好书就像站在了巨人的肩膀上, 能够看得更远, 走得更快。一本好的技术书中的文章和案例, 起到抛砖引玉的作用, 让读者跟随大师的思路, 去思考如何解决问题, 并形成自己的认识, 取得进步。

《Oracle核心技术》是Jonathan Lewis大师的又一本杰作, 该书介绍了Oracle内部核心运作机制, 详尽分析了Oracle的事务一致性、底层资源竞争、IO及优化、SQL引擎及优化等内容, 将Oracle数据库的深层技术层剖析、抽丝剥茧地展示给读者, 必能加深读者对Oracle技术的认知和理解, 促进他们将这些技术应用到实践中去。

很高兴boypoo是这本书的主要译者。boypoo作为新炬专家团成员, 曾在浙江移动的现场技术支持团队中工作过一段时间, 期间我和他有过多次合作, 经常就技术问题和技术方案进行深入的讨论, 有时工作之余还在真人CS战场上并肩作战。他对工作的热情和对Oracle数据库的深入理解

给我留下了深刻的印象。boypoo准确传神、原汁原味地将大师的杰作翻译成了中文，对于广大Oracle技术的学习者来说，阅读这样的译著无疑是一种享受。

最后，把一位朋友曾经说过的话献给大家：在繁华浮躁的时代，凡能静心著文者必有别样之情怀，我对这些著文传道者满怀敬意，也愿读者能从中获益。

王晓征

浙江移动信息系统部副主任、中国第一批OCM（2003年）

推荐序二

一本不可不读的书

对于数据库爱好者来讲，Oracle是最值得研究的数据库，也是必须研究的数据库。学习Oracle数据库时，一本好书的指引和启发是非常必要的，毫无疑问，《Oracle核心技术》就是这样一本好书，我自己也仔细地阅读过英文版。

在学习Oracle的过程中，有两本书对我的影响很大。第一本是*High Performance Oracle*，大约是10年前阅读的。阅读该书时，我将以前所学的Oracle优化知识全部联系起来了，这就是看大师杰作的好处。第二本是今年年初重新阅读的《Oracle核心技术》的英文版。我之前在支付宝管理了5年支付系统数据库，今年转到了阿里巴巴的OceanBase开发团队。我想为几个主要的开发人员推荐一本Oracle的书，看看能否借鉴一些Oracle数据库的设计思想，比如Oracle细粒度的锁机制、高效的内存管理机制、数据文件及日志文件的格式设计要点、事务管理机制、RAC的基本概念，于是推荐了《Oracle核心技术》，他们真的在用心地阅读。刚好我以前研究过MyDUL，做过Oracle日志格式研究，并从OceanBase那里了解了并发编程代码和内存管理代码，还阅读了部分MySQL源代码，这一次重读《Oracle核心技术》，收获异常丰富，有助于我进一步读懂更多的MySQL源代码。

看这本书也许会比较累。如果你有多年的Oracle数据库开发或管理经验，或者研究过Oracle的文件格式、内存结构，或者正在从事国产数据库的开发，或从事开源数据库MySQL或PostgreSQL的源码阅读的话，那么阅读此书并不会太累，而且阅读此书也是非常有必要的。很多时候我们会觉得Oracle太智能，看完本书后，你会发现Oracle其实还是一段代码或一个程序，这表示你更加清楚Oracle的内核了，也更加理解数据库的内核了。

boypoo、老熊和老耿都是国内数据库专家，他们花了大量的精力翻译了《Oracle核心技术》，这对国内的Oracle爱好者来讲，是一件非常好的事情。不仅仅是大师的书有了中文版，更重要的是，你在阅读的过程中如果发现问题，可以随时联系这三位译者，向他们请教。

楼方鑫

AUL(MyDUL)作者、Oracle ACE、知名DBA

推荐序三

鲜为人知的Oracle内核机制

认识boypoo已有多多年，深知他在Oracle技术方面的功力，听闻他翻译的《Oracle 核心技术》要上市，欣喜过望，从此，Oracle技术丛书中又多了一本技术经典之作。

结识boypoo应该是在2006年，当时我和他都喜欢流窜于ITPUB数据库论坛，平时也经常开展线下技术交流活动。我记得第一次和boypoo见面交流是在上海八佰伴，还有一位叫topas的兄弟，另一位是出差到上海的兄弟，4个人聊了一个晚上的Oracle技术。第一次面对面的交流，让我深刻体会到boypoo对Oracle技术的喜爱和严谨。

后来boypoo担任ITPUB上海教学点的主讲老师，深受众多Oracle爱好者的喜欢。当然，我也有幸聆听了《Oracle管理》及《Oracle性能优化》等课程，并且受益匪浅。生动形象、深入浅出的技术剖析，丰富而真实的经典案例，让学员们忍不住惊叹“哦，原来如此！”，感慨“山穷水复疑无路，柳暗花明又一村”。

常常有爱好数据库的朋友想深入了解Oracle的底层原理及实现机制，我总是喜欢推荐两本书，其中一本就是《Oracle 核心技术》。本书从Oracle的锁机制、Latch机制、事务管理机制、内存管理机制以及各类数据文件等方面，详细阐述了Oracle内部鲜为人知的“秘密”。之前有些朋友一提及此书就感到头痛，一是因为全是英文，二是需要有一定的Oracle数据库知识。现在boypoo等人花了大量心血将此书翻译成了中文，这对Oracle数据库爱好者来说是一个福音。这几天我逐章阅读《Oracle 核心技术》中文版，更清楚地了解了Oracle内核，再加上boypoo翻译功底深厚，用词准确、通俗，读起来简直是一种享受。

最后，再次感谢boypoo努力将《Oracle 核心技术》翻译成中文并与大家分享，我相信本书会给大家耳目一新的感觉，也相信大家一定能受益匪浅，那就让我们好好品味吧。

杨向勇

前程无忧系统总监兼首席架构师

译者序

终于完工了!

从本书英文版于2011年11月出版,着手准备翻译到现在,已经过去整整21个月了。这是我的第一本译作,第一次翻译就选择了Oracle最难的内核著作,还是个业界大师写的,而且是个英国人(我的意思是美国人的用词会更加明快些),这就更增加了翻译的难度。翻译如此艰难,以至于在几次中耕期间,曾一连几天不想去点击那个命名为trans的文件夹,几度暗下决心再也不翻译了。幸好有我的老兄弟老熊和老耿,他们对Oracle内核的理解都远胜于我。我们仨互相搀扶和鼓励,坚持完成了本书的翻译工作,读者也终于能见到本书的中文版了。

尽管中文版的面世比英文版晚了近两年,但作者Lewis的回信给了我足够的信心。虽然12c已经发布,但Oracle核心引擎的机制没有变,仅仅是在本书阐述的思想之上做了些演进,因此读者不必担心本书已经过时了。

正如亚马逊上的书评所说,在Oracle内核领域,本书仅次于已经隐退的Steve Adams的*Oracle Internal Services*。Steve的著作已主导内核领域十余年,也许接下来将让位于Lewis的这本书了。

之前我常常读别人的译作,读文学类的作品大都觉得如此优美,读计算机书籍大都觉得晦涩难懂,因此我们翻译本书的一个目标就是:尽量让读者在阅读过程中,不必翻出英文版对照着去理解。为了将本书翻译得足够地道,我们重温了著名翻译家思果先生的《翻译研究》,并在网络上从阮一峰、余晟及李松峰等老师的翻译心得中汲取营养。译者已经竭尽全力,尽量让本书读起来轻松易懂。

对于本书,我想说几点。第一,虽然本书讲的是Oracle核心技术,但作者匠心独具,用词非常精炼。书中所有细节都是大名鼎鼎的Lewis通过不断的dump研究出来的,它并不像某些脚本书,输出和脚本占据了较大篇幅,因此可谓字字珠玑。古代讲究“增之一分则太长,减之一分则太短”,作者则是“no more no less”。第二,本书虽然以10g和11g为主,但是12c仍然适用。2000年安装Oracle7.3.2 WGS时靠软盘,2013年安装12.0.1EE时用的是互联网下载,安装方式和步骤虽有所不同,但Oracle最核心的进程依然是pmon、smon、ckpt、dbwr、lgwr,最核心的机制依然是redo/undo、事务、锁、缓存、解析,本书主要讲的就是这些核心机理。第三,对于书中绝大部分专有名词,我们都使用了标准译法,对于可能出现多种译法的词,我们会选择用得比较多的那种,比如extent,

目前的译法有“范围”、“区”、“扩展”，我们选择了“扩展”，也有些词我们尽可能不翻译，比如redo和undo，以保持读者的固有阅读习惯。

我翻译了本书的第1、2、7章，老耿翻译了本书的3、4、9章，老熊翻译了本书的5、6、8章，全书由我统稿。尽管得到了很多朋友的帮助，但译文差错仍在所难免。（上大学的时候，看到教材作者在前言里这么写的时候，总觉得这人好虚伪，明明是个大家，还要说自己水平低，等到自己写的时候，才知道他们有多诚恳。）读者们可以在图灵社区提交勘误，也可以用email反馈于我（yangzhihong@shsnc.com）。非常感谢！

感谢三位译者家人的支持，她们在繁忙的工作和学习之余，给予了我们很大的鼓励和支持，让我们尽可能地将本书译好。

感谢上海新炬高级副总经理程永新先生对我当前工作的支持，还要感谢中通服云南公司的徐明熙先生，多谢所有曾经帮助过我的领导、同事、客户和朋友们，是你们的关心和支持让我在IT系统运维和架构这条路上越走越好。

最后要感谢图灵公司傅志红、李瑛和岳新欣等出版界同仁的帮助，感谢各位在翻译和审校等环节的辛勤付出。

杨志洪

2013年8月29日初稿于合肥

2013年10月10日定稿于上海

写给中国读者

在写作本书时，Oracle最新的版本是11.2.0.2。而在写中文版序时，12.1.0.1（12c）已经发布几个星期了。

全世界的Oracle专业人士都在致力于更新软件版本、研究新特性、增强现有功能或者是修复bug，但似乎从来没有时间去学习一些必备知识。

想一个简单的事实，我在本书前言提到：x\$kcbsw结构包含了Oracle处理一个“会话逻辑I/O”所需的所有命名函数。在11.2.0.3版本，我曾指出有1164个命名函数，但看看运行在64位OEL（Oracle企业版Linux操作系统）上的12.1.0.1版本，这个数字是1300。这一变化对本书内容来说有多大区别？对普通Oracle专业人士呢？答案是“没有区别”！

极少的情况下，你可能会遇到某个函数出现ORA-00600或ORA-07445错误，这时你所能做的只是搜集一些能帮助发现并修复这个bug的其他信息；而一旦你掌握了Oracle访问缓冲块的原理，并大致了解了读一致性和块清除之类的话题，可能就不用非得知道某个具体的操作在做什么了。

确实，12c相对于11g有些变化，可能需要看看有哪些重大的改变。这些变化有可能是一些多租户（可插拔）数据库所带来的负面影响；有可能是日志写进程工作方式的改变，允许由“LGWR从属进程组”完成；也有一些变化非常接近数据库引擎的核心，我们可以认为我们今天所了解的这些原则依然有效，只不过未来会有些演进而已。

学习Oracle时要懂得权衡。大部分时候，只要大体知道引擎是如何工作的就足够了，偶尔才需要知道一些世界上只有一小部分人才会知道的精确资料。注意，不要浪费时间去研究不必要的细节，而是要找到折中的办法，使你所掌握的知识足以预判Oracle在你没见过的场景中会怎样做。

本书旨在介绍Oracle的工作机制。我非常荣幸，也非常高兴，有人把它翻译成了中文。希望你喜欢阅读Oracle的故事，也希望它可以帮助提升你的专业技能。

Jonathan Lewis

2013.08.10

引 言

在*Practical Oracle 8i*出版后的第三周，我就收到了询问什么时候出版该书9i版本的邮件——Larry Ellison刚好在这个时候发布了9i版本。过去的12年里，不停有人问我相同的问题，不同的只是版本号。本书基本相当于该书的第二版，仅仅涵盖了第一版的第1章，以及第2章和第3章的少量内容。

有两件事促使我再次提笔写这样一本书。首先，我无数次被问到：Oracle如何用来做某某事？其次，我发现对于这样的问题，很难找到既充分又明确的答案。通常，你只要遍阅Oracle文档，就会找到许多常见问题的答案；只要在网上一搜，就会找到许多介绍Oracle工作原理的文章，却找不到一个全面系统的文档，按正确的顺序进述Oracle的方方面面，让你大体了解Oracle的工作原理以及它为什么只能这样工作。本书要做的就是这件事，即完整讲述Oracle的工作原理。我会系统地讲述，而不是东拼西凑一些细枝末节。

本书目标

本书只有区区两三百页，而11g文档有上万页，所以我似乎不太可能把“整件事”讲清楚，那就事先声明一下。本书讲的是中央数据库引擎的核心机制，这些内容会联动Oracle所有其他内容，主要包括undo、redo、高速缓冲区缓存及共享SQL。即使这样，我仍然不得不舍弃一些细节和有趣的特殊案例，免得本书冗长、臃肿不易阅读。随便举个例子，考虑一个简单的问题：Oracle如何处理一次逻辑I/O？先看看x\$kcbsw结构，它包含了Oracle访问一个块可能调用的所有函数。你会发现，对于11.2.0.2版本，做一次逻辑I/O可以调用的函数有1164个之多。你真的希望详细了解这些函数吗？还是说只要大致了解一下一般要求就足够了？

细节问题在不同的层面重复——关于复杂的问题你了解多少？假如我真的花时间写了些异常复杂的细节，读者会受益多少呢？另外，本书在完整性、精确性和基本的可读性之间做了必要的折中。我认为，我遵循的是Oracle RWP部门的Andrew Holdsworth在2006年OOW大会上所传达的理念。在关于优化器及怎样搜集统计信息的演讲中，他谈到了90/9/1法则，具体说明如下：

- 90%的情况下，默认样本就行了；
- 9%的情况下，需要扩大样本采集范围；
- 剩下1%的情况跟样本大小无关。

这是著名的80/20帕累托法则的增强版，我认为它非常适用于描述理解Oracle核心机制的需求。为便于解释本书，我将该法则重新调整为：90%的情况下你只需了解Oracle如何保持系统运行，1%的情况下你需要搞清楚哪里出了问题，而本书关注的是剩余9%的情况——如果你对Oracle幕后工作有更多的了解，就会更加了解Oracle数据库，会少浪费些宝贵时间。

接下来呢

前段时间，有人问Tanel Pöder（我的技术审阅人）什么时候写一本Oracle核心内幕的书，他这样回答：

“如果要写一本常规的老式纸质书，那么答案是永远都不会写，因为这个领域变化太快了，而且写好这样一本书至少要花一整年的工夫，等到书出版时许多细节必然早已过时。”

这个回答很好。这也再次证明了我的观点，即要避免在那1%上浪费时间，而要坚持关注那些普遍的需求及类似的东西。Tanel没有写纸质书，而是在<http://tech.e2sn.com/oracle>写了一本实时更新的电子书。

但是，纸质书很棒（就算它做成了电子版本），而且我认为，书籍的排版格式会使其内容不同于若干篇互联网文章（就算写得非常好）。纸质书读起来能给人以内容连贯的感觉，整合的文章却不能。在我写这篇引言的时候，我的博客上已经有650篇文章了，文字量比本书要大得多。我可以从中摘录一些放到一起做一个迷你系列专题，而且即使将它们汇编起来，也不会是一本很差的书（哪怕需要我再花些时间写点内容将文章前后贯穿起来）。但是，即使是技术书籍，也需要给读者带来叙述连贯的感觉。

为了解决纸质“非实时”的问题，我在我的博客<http://jonathanlewis.wordpress.com/oracle-core/>上写了许多篇博文。其中每篇对应本书的一章。上面还会给出本书纸质版的勘误或简短的补充。因为是博客的形式，所以读者可以在后面提问或评论。当读者询问我的其他书什么时候再版时，我的回答都是不会再版。但是，根据读者的反馈，我也许会发现书中的一些话题需要进一步解释，我漏掉了一些流行的话题，甚至有些全新的话题需要用一整章或是附录来谈论。

我的开场白说完了。现在轮到你，我的读者，来发表建议了。

致 谢

首先，我要感谢我的妻子Diana，正是由于她的支持，我才顺利完成了本书的写作。没有她对我的爱和耐心，就没有这本书。我记不清她给我端过多少杯茶（经常是茶冷了我都还没喝），也记不清有多少个深夜或是凌晨我将她吵醒，更不知她隐忍了多少抱怨，因为有许多事本该由我来做，却由于我忙着写作而无暇抽身。写这本书需要集中精力，如果不是她全权处理了所有的生活琐事，我根本没法将它完成。

技术书需要有优秀的审阅人，因此我必须感谢我的官方技术审阅人Tanel Pöder（<http://tech.e2sn.com/>和<http://blog.tanelpoder.com/>），他的点评、质疑以及修正对我非常有帮助。

还有两位“非官方”的审阅人，Martin Widlake（<http://mwidlake.wordpress.com>）和 Timur Akhmadeev（<http://timurakhmadeev.wordpress.com/>），他们也为本书的精彩呈现提供了大量的帮助。我请他们告诉我，我有哪些内容遗漏了没有写，哪些内容多余应该删掉，哪些内容写得不够清楚。毫无疑问，他们也帮忙指出了许多技术错误。特别要说的是，如果你喜欢本书中的图表，得感谢Martin，因为他不断跟我说，我讲的有些东西最好用图表来展现。

Andrey Nikolaev（<http://andreynikolaev.wordpress.com/>）专门审阅了本书的第4章。互联网上有大量关于Oracle内部工作原理的信息，但大部分都是这些年已经慢慢过时的材料。关于latch锁定的信息就能说明了上述现象。毫无疑问，这其中部分原因是搞清楚这个课题非常困难。在呈现latch及互斥锁的最新信息方面，Andrey做了很多工作，甚至远远超过了我在互联网上看到的所有其他作者。因此我问他能否帮我审阅第4章的内容。非常感谢他提出的宝贵意见和所做的修正。

尽管这些审稿人都非常优秀，但仍可能在书中发现错误或疏漏。无论是何种错误或疏漏，责任都由我承担。书中有些例子是我故意简化的，有些则可能是我弄错了。书中的大量内容都是关于Oracle内部机制的，而我和几位审稿人都无权访问有关这些机制的内部文档。我们都是普通人，只是使用科学的方法去观察并验证它们。我们会不时地展开讨论，以判断某部分内容究竟是怎样运作的，直到出现新的证据。

我还应该感谢很多人，其中有些我能叫上名字，有些我连名字都不知道。他们都为本书贡献了力量，或提出问题，或给出答案，或请我去帮忙诊断有趣的问题。OTN（<http://forums.oracle.com>）和Oracle-L（www.freelists.org/archive/oracle-l）等论坛上总有许多的问题，一些人描述了在使用Oracle的过程中遇到的问题，一些有知识和经验的人则提供了自己的见解和解决方案。如果不是看到源源不断的新问题，你很容易止步不前，自以为无所不知。因此可以说，在Oracle方面我之所以了解不少知识，都是因为互联网上的那些问题促使我不断进行研究。