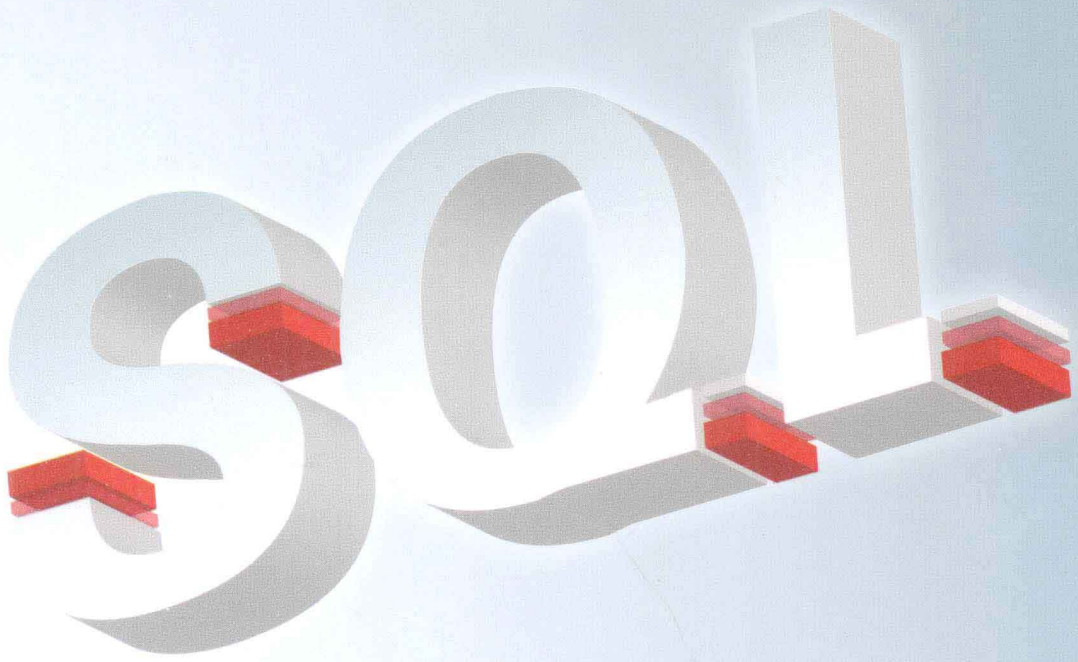
Broadview
www.broadview.com.cn

# 基于Oracle的SQL优化

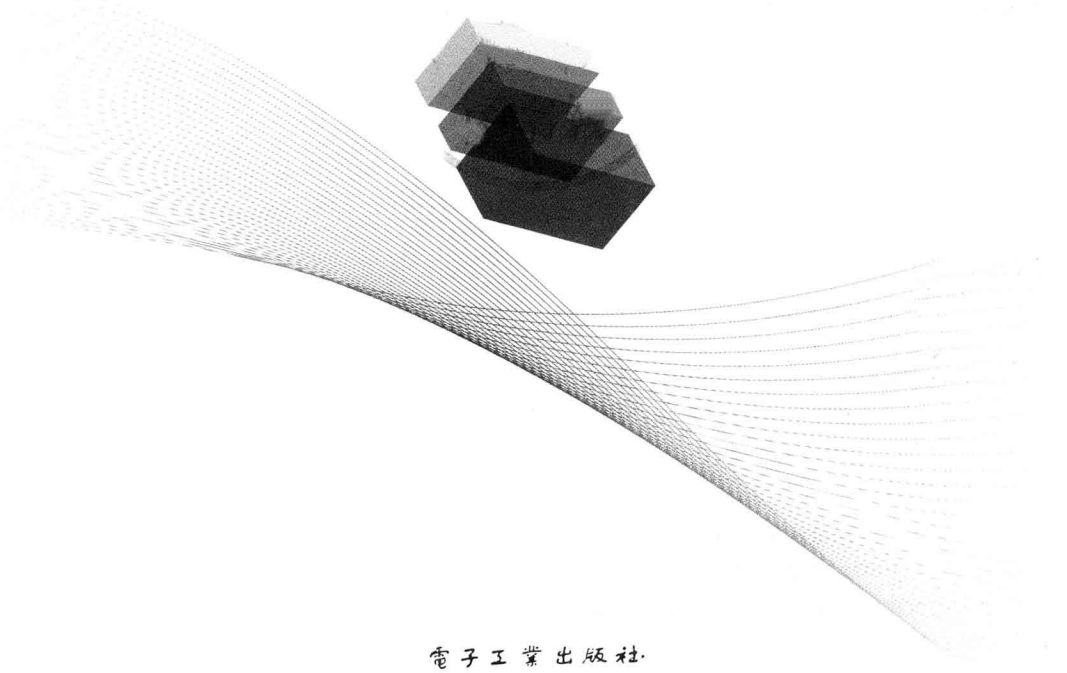
崔华 编著




電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

# 基于Oracle的SQL优化

崔华 编著



電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

## 内 容 简 介

这是一本与众不同的书，它的目的是使读者真正掌握如何在 Oracle 数据库里写出高质量的 SQL 语句，以及如何在 Oracle 数据库里对有性能问题的 SQL 做诊断和调整。

本书从 Oracle 处理 SQL 的本质和原理入手，由浅入深、系统地介绍了 Oracle 数据库里的优化器、执行计划、Cursor 和绑定变量、查询转换、统计信息、Hint 和并行等这些与 SQL 优化息息相关的本质性内容，并辅以大量极具借鉴意义的一线 SQL 优化实例，阐述了作者倡导的“从本质和原理入手，以不变应万变”的优化思路，最后还介绍了作者在实际工作中总结出来的 Oracle 数据库里 SQL 优化的方法论。

本书适用于使用 Oracle 数据库的开发人员、Oracle DBA 和其他对 Oracle 数据库感兴趣的人员，也可以作为各院校相关专业的教学辅导和参考用书，或作为相关培训机构的培训教材。



**图书在版编目（CIP）数据**

基于 Oracle 的 SQL 优化 / 崔华编著. —北京：电子工业出版社，2014.1
ISBN 978-7-121-21758-6

Ⅰ. ①基… Ⅱ. ①崔… 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 257674 号

策划编辑：张春雨
责任编辑：许　艳
印　　刷：北京天宇星印刷厂
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×1092　1/16　印张：53.5　字数：1353.9 千字
印　　次：2014 年 1 月第 2 次印刷
定　　价：128.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

# 推荐序

毫无疑问，崔华写的这本书是一部宏篇巨著，也是国内为数不多的深入讲解 Oracle SQL 性能优化的著作之一。这是他用了一年多的业余时间，特别是用了大量的夜晚，用心写就的一本书。即便是对于我这样看过很多 Oracle 书籍的人来说，这本书也同样带给我新的视角、新的知识。这本书在数据库优化方面的价值，我个人认为怎么评价都不为过。

Oracle RDBMS 是一套巨大而复杂的系统软件，但同时又开放了很多的门让我们去深入探索它。Oracle 的优化器、共享池和游标等，是这套系统内部最为复杂的几个部分。现在，这本书又帮助我们在 Oracle 数据库中打开了另一道门，让我们探索 Oracle 怎样执行 SQL 语句，怎样选择最优的 SQL 执行路径，怎样避免串行争用从而提高并发处理能力。并且在本书的最后一章，作者将 SQL 优化总结上升到了方法论，这对于指导我们进行 SQL 优化的帮助是巨大的。

数据库性能优化是系统生命周期中不可缺少的一环，我们的大多数数据库都会或多或少遇到一些性能问题。特别是随着业务的发展，数据量增加、系统用户数增多，以及系统之间的接口越来越复杂，都会导致系统的性能越来越恶化。怎样去优化一套数据库，我们需要掌握很多的知识，包括存储子系统、操作系统、数据库，甚至还有业务逻辑。很多人面对性能问题不知如何下手，好在数据库的性能问题至少有 80%以上都是由于 SQL 语句性能较差所导致，同时随着硬件性能的提升和硬件价格越来越便宜，硬件所带来的系统瓶颈相对越来越小，所以在大多数情况下，我们优化数据库也就是优化 SQL 语句。

对于任何一项技术的传授来说，都有术与道的区别，也即鱼和渔的区别。如果只有术，则只能是一种经验的积累，不能举一反三，在遇到未知问题时也无法扩展分析；如果只有道，则太过于理论化和学院派，应用到实际中又比较困难。本书很好地把术与道结合在一起，也就是我们常常所说的理论与实践相结合。在书中有性能优化关键的理论知识，也有大量的测试来验证优化理论，还有丰富的实践案例。我相信，通过这样的方式，每一个认真阅读过这本书的读者朋友都会在 Oracle 数据库性能优化上获得大量的知识和技巧。

Oracle 数据库的世界里，有如此之多的与性能优化相关的术语：CBO、优化器、查询转换、执行计划、Hint、并行、游标、绑定变量、统计信息、直方图、索引，等等。在这本书里，你可以找到上述所有这些术语的详细解释和分析。

欢迎来到 Oracle 数据库优化器的世界！

熊军

云和恩墨西区技术总监，Oracle ACE

2013 年 6 月 1 日于成都

# 致谢

感谢黄远邦，你在 SQL 优化方面的技术极为精湛，感谢你抽时间审阅了这本书的部分章节并提出了很好的修改建议。如果没有你亲手做的那 20 多个 SQL 优化的实例，我可能连写这本书的勇气都没有。

感谢熊军，你是我见过的 Oracle 数据库技术最好的人，没有之一。感谢你在百忙之中耐心审阅了这本书的全部章节，对于我认为没有问题的章节你也总是能从中挑出不少错误并给出极好的修改建议，你在审阅这本书的过程中所表现出的严谨和技术功底无数次令我叹为观止。事实上，这么多年以来，你我一直维持亦师亦友的关系，我从你身上学到了太多太多。

感谢黄凯耀，感谢你帮我审阅了这本书的部分章节，你的严谨以及给出的专业意见让我获益匪浅。

感谢我的太太邹乐，原先我一天到晚晃啊晃的时候你总催我写书，而等我真正开始写书时你又总劝我多休息，你是真的对我好。这一年多以来，我把绝大部分应献给你和孩子的业余时间都给了这本书，感谢你的理解和支持。

感谢毕宁，如果没有你当初不断的催促，就不会有这本书的诞生。虽然后来这本书转由策划编辑张春雨接手，但是他和责任编辑许艳同样是极为靠谱之人，感谢你们的专业和高效以及对我不断拖延交稿日期的容忍。

感谢盖国强（eygle），如果没有你当初推荐我参加 2010 年的中国软件技术大会，我就不会认识毕宁，自然也就不会有这本书的产生。感谢你这么多年以来对我的不断关照和提携，你一直是我的榜样和努力的方向。

最后我还要感谢高翔和袁永俊，我在公司和部门内部开设“SQL 优化”这门课程很大程度上是为了完成当初自己在竞聘时对你们所做的承诺，没有“SQL 优化”这门课程，同样也就不会有这本书。感谢这么多年以来你们对我的支持和理解，如果不是你们，我不可能有今天。

# 前言

## 为什么写这本书

写这本书纯属偶然。

2010 年 12 月 11 日，我在中国软件技术大会上做了一个关于 Oracle 数据库备份与恢复机制揭密的主题演讲。可能是因为这个演讲的缘故，电子工业出版社的编辑毕宁随后多次邀请我写一本关于 Oracle 数据库备份恢复方面的书，但均被我以各种理由搪塞、推脱。这一拖就拖了大半年（为什么会推脱？一来是因为我认为备份恢复这个点相对来说较窄，不具备普适性；二来市面上已经有不少关于备份恢复方面的书，这意味着如果我想写出有新意、有不一样的内容的话，会有相当的难度）。

这种推脱一直持续到 2011 年 10 月，那个时候我正好在公司内部主讲一个基于 Oracle 数据库的 SQL 优化的系列课程。我开设这门课程的初衷是因为当时恰逢公司开始研发新一代系统，而我深知对于使用 Oracle 数据库的应用系统而言，SQL 语句的质量会直接影响系统的性能，甚至可以说大部分基于 Oracle 数据库的应用系统的性能问题都是由于开发人员不懂 Oracle 数据库，不懂如何在 Oracle 数据库里写出高质量的 SQL 所致。这样的系统性能问题，单靠高水准的 Oracle DBA 来调整是非常痛苦的，很多时候是事倍功半。如果能把我在 SQL 优化方面的经验分享给大家，告诉大家如何避免在 Oracle 数据库中写出很烂的 SQL，如何在 Oracle 数据库中做 SQL 优化，那么就可以从源头上提升新一代系统在数据库端的性能，这也算是我为公司新一代系统的研发所做的一份贡献。

这门课程一经推出，就取得了很好的反响，同事们纷纷反馈说这门课程很实用，课程里的不少方法和知识点在实际的工作中都能用上，这使我意识到自己是在做一件非常有意义的事情，虽然辛苦，但是确实能帮助到同事。

这门课程大约是30个学时，我才开始讲没多久，毕宁就再次打电话给我，他还是希望我能写一本关于Oracle数据库备份恢复方面的书。我清楚地记得那天下午接到毕宁电话的时候，脑海里突然闪现了一个念头——为什么不把现在讲的这门 SQL 优化的课程写成一本书呢？这样一来可以对毕宁有个交待，二来也可以帮到更多的人，而不仅仅是我的同事。因为只要是用 Oracle 数据库的，只要是构建在 Oracle 数据库上的应用系统就必然会涉及 SQL 优化，也就是说 SQL 优化不同于备份恢复，它是具备普适性的。另外，市面上系统阐述 Oracle 数据库中 SQL 优化的书非常少，这意味着我有很大的发挥空间。如果能写一本系统的、从本质上阐述如何在 Oracle 数据库里做 SQL 优化的书，能够通过这本书教会开发人员如何在 Oracle 数据库里写出高质量的 SQL，以及如何对有性能问题的 SQL 做诊断与调整，那么也许就可以从源头上保证，由这些开发人员所开发出来的基于 Oracle 数据库的应用系统在 SQL 上是没有性能问题的，而那些由于 SQL 撰写不当而导致的各种性能问题也就不复存在了。如果真能做到这一点，那真是一件功德无量的事情。

在和毕宁沟通过几次后，我的上述想法获得了他的支持，于是从 2011 年 10 月份开始，我就正式开始撰写

这本书。只是我万万没有想到，这一写就写了 17 个月。

这本书的撰写过程是极为艰苦的。一来是因为我对自己要求很高，希望写出来的书通俗易懂（普通的使用 Oracle 数据库的开发人员就能看懂），但同时又要具备一定的深度；二来是因为我倡导“从本质和原理入手，以不变应万变”的 SQL 优化思路，必然涉及深入介绍 Oracle 数据库里的优化器，但 Oracle 数据库里的优化器实在是太复杂了。

现在回想起来，我为这本书付出了太多太多。在这一年多的撰写过程中，由于长期熬夜，我能明显感觉到身体越来越差，到了撰写后期更是频繁往医院跑，但无论如何，我还是坚持下来了。

## 本书的主要内容

本书共 8 章。

第 1 章 “Oracle 里的优化器”，详细介绍了 Oracle 数据库中与优化器相关的各个方面的内容，包括优化器的模式、结果集（Row Source）、集的势（Cardinality）、可选择率（Selectivity）、可传递性（Transitivity）、各种数据访问的方法，以及与表连接相关的内容。

第 2 章 “Oracle 里的执行计划”，详细介绍了 Oracle 数据里与执行计划有关的各个方面的内容，包括执行计划的含义，如何查看执行计划，如何得到目标 SQL 真实的执行计划，如何查看执行计划的执行顺序，Oracle 数据库里各种常见的执行计划的含义，以及如何在 Oracle 数据库中稳定执行计划。

第 3 章 “Oracle 里的 Cursor 和绑定变量”，详细介绍了 Oracle 数据库中与 Cursor 和绑定变量相关的各个方面的内容，包括 Shared Cursor、Session Cursor、绑定变量、游标共享、硬解析、软解析、软软解析，以及与它们息息相关的 Oracle 数据库里的四种应用类型。

第 4 章 “Oracle 里的查询转换”，详细介绍了 Oracle 数据库中与查询转换有关的各个方面的内容，包括子查询展开、视图合并、星型转换、连接谓词推入、连接因式分解、表扩展、表移除，以及 Oracle 如何处理 SQL 语句中的 IN。

第 5 章 “Oracle 里的统计信息”，详细介绍了 Oracle 数据库里与统计信息相关的各个方面的内容，包括 Oracle 数据库中各种统计信息的分类、含义、收集和查看方法，以及如何在 Oracle 数据库里正确地收集统计信息。

第 6 章“Oracle 里的 Hint”，详细介绍了 Oracle 数据库中与 Hint 有关的各个方面的内容，包括什么是 Hint，如何用 Hint，Hint 什么情况下会失效，以及 Oracle 数据库中常见的各种 Hint。

第 7 章 “Oracle 里的并行”，详细介绍了 Oracle 数据库里并行的基本概念以及在 Oracle 数据库里如何控制并行，包括在 Oracle 数据库里开启并行、控制并行度等。

第 8 章 “Oracle 里 SQL 优化的方法论”，介绍了在 Oracle 数据库里如何做 SQL 优化，提出了我们总结出来的 Oracle 数据库里 SQL 优化的方法论，并结合实例验证了上述方法论。

## 本书的读者对象

本书适用于使用 Oracle 数据库的开发人员、Oracle DBA 和其他对 Oracle 数据库感兴趣的人员。

本书也可以作为各大中专院校相关专业的教学辅导和参考用书，或作为相关培训机构的培训教材。

## 本书代码下载

本书使用的所有脚本和范例代码均可以通过我网站上的 Books 专栏下载，网址为 http://www.dbsnake.net/books。

## 本书约定

本书介绍的 SQL 优化方法论是通用的方法，并不局限于 Oracle 数据库的某个具体的版本，但本书的实例和测试结果绝大部分是基于 Oracle 11gR2 的（除个别特别注明的实例之外）。而 Oracle 数据库不同的版本之间在某些方面可能会差别很大，所以即便是同样的实例，在 Oracle 数据库不同的版本上的测试结果也有可能不同，在此特别说明，一切以实际情况为准。

由于我的水准和经验所限，书中的错误之处在所难免，在此诚挚期待大家阅读后的指正。可以通过电子邮件与我取得联系（allantreycn@gmail.com），欢迎与我交流任何关于本书的问题。

# 目　录

# 第 1 章

# Oracle 里的优化器

到目前为止，Oracle 数据库是市场占有率最高（接近 50%），使用范围最广的关系型数据库（RDBMS），这意味着有太多太多的系统都是构建在 Oracle 数据库上的。而我们大家都知道，**对于使用关系型数据库的应用系统而言，SQL 语句的好坏会直接影响系统的性能，很多系统性能很差最后发现都是因为 SQL 写得很烂的缘故**。实际上，一条写得很烂的 SQL 语句就能拖垮整个应用，极端情况下，一条写得很烂的 SQL 语句甚至会导致数据库服务器失去响应或者使整个数据库 Hang 住，去 Google 一下吧，这样的例子有很多！

**怎样避免在 Oracle 数据库中写出很烂的 SQL？或者说应该如何在 Oracle 数据库中做 SQL 优化？**这个问题真的很不好回答，且容我慢慢道来。

对所有的关系型数据库而言，优化器无疑是其中最核心的部分，因为优化器负责解析 SQL，而我们又都是通过 SQL 来访问存储在关系型数据库中的数据的，所以优化器的好坏会直接决定该关系型数据库的强弱。从另外一个方面来说，正是因为优化器负责解析 SQL，所以要想做好 SQL 优化就必须了解优化器，而且最好是能全面、深入的了解，这是做好 SQL 优化基础中的基础。

Oracle 数据库里的优化器以其复杂、强悍而闻名于世，本章会详细介绍与 Oracle 数据库里优化器相关的基础知识，目的是希望通过这一章的介绍，让大家对 Oracle 数据库里的优化器有一个全局、概要性的认识，打好基础，为阅读后续章节扫清障碍。

## 1.1 什么是 Oracle 里的优化器

优化器（Optimizer）是 Oracle 数据库中内置的一个核心子系统，你也可以把它理解成是 Oracle 数据库中的一个核心模块或者一个核心功能组件。优化器的目的是按照一定的判断原则来得到它认为的目标 SQL 在当前情形下最高效的执行路径（Access Path），也就是说，优化器的目的就是为了得到目标 SQL 的执行计划（关于执行计划，会在“第 2 章 Oracle 里的执行计划”中详细描述）。

依据选择执行计划时所用的判断原则，Oracle 数据库里的优化器又分为 RBO 和 CBO 这两种类型。RBO 是 Rule-Based Optimizer 的缩写，直译过来就是“基于规则的优化器”；相对应的，CBO 是 Cost-Based Optimizer 的缩写，直译过来就是“基于成本的优化器”。

在得到目标 SQL 的执行计划时，RBO 所用的判断原则为一组内置的规则，这些规则是硬编码在 Oracle 数据库的代码中的，RBO 会根据这些规则从目标 SQL 诸多可能的执行路径中选择一条来作为其执行计划；而

CBO 所用的判断原则为成本，CBO 会从目标 SQL 诸多可能的执行路径中选择成本值最小的一条来作为其执行计划，各个执行路径的成本值是根据目标 SQL 语句所涉及的表、索引、列等相关对象的统计信息计算出来的（关于统计信息，会在“第 5 章 Oracle 里的统计信息”中详细描述）。

Oracle 数据库里 SQL 语句的执行过程可以用图 1-1 来表示。

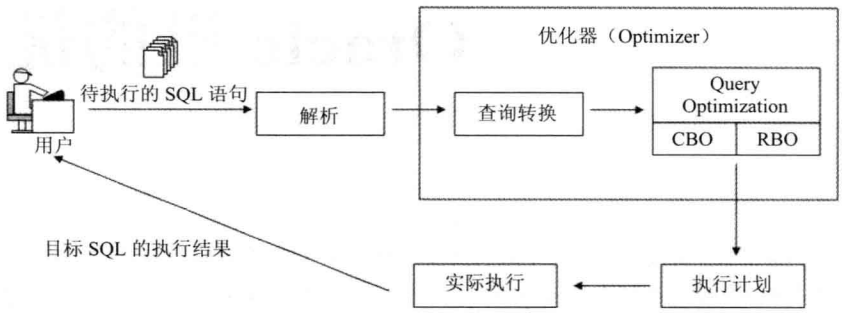



图 1-1 Oracle 数据库里 SQL 语句的执行过程

关于图 1-1，会在“第 4 章 Oracle 里的查询转换”中详细说明，这里只需要知道 Oracle 里优化器的输入是经过解析后（在这个解析过程中，Oracle 会执行对目标 SQL 的语法、语义和权限检查）的目标 SQL，输出是该目标 SQL 的执行计划就好了。

接下来，分别介绍 RBO 和 CBO。

## 1.1.1 基于规则的优化器

之前已经提到，基于规则的优化器（RBO）通过硬编码在 Oracle 数据库代码中的一系列固定的规则，来决定目标 SQL 的执行计划。具体来说就是这样：Oracle 会在代码里事先给各种类型的执行路径定一个等级，一共有 15 个等级，从等级 1 到等级 15。并且 Oracle 会认为等级值低的执行路径的执行效率会比等级值高的执行效率要高，也就是说在 RBO 的眼里，等级 1 所对应的执行路径的执行效率最高，等级 15 所对应的执行路径的执行效率最低。在决定目标 SQL 的执行计划时，如果可能的执行路径不止一条，则 RBO 就会从该 SQL 诸多可能的执行路径中选择一条等级值最低的执行路径来作为其执行计划。

RBO 是一种适用于 OLTP 类型 SQL 语句的优化器，在这样的前提条件下，大家来猜一猜 RBO 的等级 1 和等级 15 所对应的执行路径分别是什么？

在 Oracle 数据库里，对于 OLTP 类型的 SQL 语句而言，显然通过 ROWID 来访问是效率最高的方式，而通过全表扫描来访问则是效率最低的方式。与之相对应的，RBO 内置的等级 1 所对应的执行路径就是“single row by rowid（通过 rowid 来访问单行数据）”，而等级 15 所对应的执行路径则是“full table scan（全表扫描）”。

RBO 在 Oracle 中由来已久，虽然从 Oracle 10g 开始，RBO 已不再被 Oracle 支持，但 RBO 的相关实现代码并没有从 Oracle 数据库的代码中移除，这意味着即使是在 Oracle 11gR2 中，我们依然可以通过修改优化器模式或使用 RULE Hint 来继续使用 RBO。

和 CBO 相比，RBO 是有其明显缺陷的。在使用 RBO 的情况下，执行计划一旦出了问题，很难对其做调整；另外，如果使用了 RBO，则目标 SQL 的写法，甚至是目标 SQL 中所涉及的各个对象在该 SQL 文本中出现的先后顺序，都可能会影响 RBO 对于该 SQL 执行计划的选择。更糟糕的是，Oracle 数据库中很多很好的特