DBA的思想天空

——感悟Oracle数据库本质

白 鳝 储学荣⊙编著

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

DBA的思想天空

——感悟Oracle数据库本质

白 鳝 储学荣 编著

人民邮电出版社
北 京

图书在版编目（CIP）数据

DBA的思想天空 : 感悟Oracle数据库本质 / 白鳝, 储学荣编著. -- 北京 : 人民邮电出版社, 2012.10
（图灵原创）
ISBN 978-7-115-29443-2

Ⅰ. ①D… Ⅱ. ①白… ②储… 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220663号

内 容 提 要

本书重在介绍Oracle数据库的性能调优方法及相应的工作思路，但并不拘泥于技术细节。作者结合多年的丰富经验，借助大量真实案例剖析了相关技术原理，阐述了理论知识在实践中的应用方法，并总结出“思路是道，操作方法是技。得道是极大的提升，也是DBA的思想精髓”的精辟论断。

全书分为三个部分，共19章。第一部分介绍了Oracle的基本原理，以及从基本原理衍生而出的一些分析问题的方法和思路。第二部分介绍了DBA应该掌握的常用工具。第三部分介绍了DBA分析问题的主要思路和一些典型案例。

图灵原创

DBA的思想天空

——感悟Oracle数据库本质

◆ 编　　著　白　鳝　储学荣
责任编辑　王军花
执行编辑　赵慧明

◆ 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
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：28
字数：668千字　　2012年10月第1版
印数：1－4 000册　　2012年10月北京第1次印刷

ISBN 978-7-115-29443-2

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

写在前言之前的话

按理说前言之前就不应该说什么了。不过这几天我真的有一种不吐不快的感觉，如果不马上把这些东西写出来和大家共享，就好像会对读者有所愧疚似的。今年 5 月 20 号是老白的母校南京大学的 110 周年庆典，也是我们南大计算机系 88 级同学毕业 20 周年，于是有好事之徒组织了一次同学聚会。遇到这种大碗喝酒、大块吃肉的好事，老白自然是积极响应的。不过既然是南大校友的聚会，当然会有所不同了。我们安排了一个活动，请当年最英俊潇洒、才华横溢的陈道蓄老师为我们再讲一次课，让我们回味一下 20 年前当学生的感觉。

本以为这就是一次活动而已，并没有十分认真。没想到这个活动最终成为“全国 60 名优秀中学生和诺贝尔奖获得者面对面活动”中的一个环节，安排我们和来自全国的 60 名优秀中学生一起聆听陈道蓄教授的讲座。而陈道蓄教授也一如 20 年前的严谨，为我们精心准备了一次 30 分钟的讲座，题目是《软件工程中的“软实力”》。这个和软件工程有关的讲座，以几幅世界名画导入软件工程中的核心问题，让人回味无穷。更令我感到惊喜的是陈老师关于“道”的感悟，他对于形而上的追索，让我有一种醍醐灌顶的感觉。由于音响调试得不好，坐在后排的我只能支棱着耳朵，认真地听他所讲述的每一个字，直到全场响起热烈的掌声，我才从沉醉之中醒来。我想，这 30 分钟，足以让我受益 10 年。而陈老师关于道的感悟，正和老白这本书的宗旨契合，悟道是“Thinking in Oracle”的最高境界。

陈老师谈到做一件事，总是会碰到“道”和“器”。道是形而上的东西，是别人无法教授的，只能靠自己感悟；器是有形的东西，很容易获得，也最容易使人从中受益。由于悟道艰难，因此大多数人都会追求器而放弃悟道。不过道终究是形而上的东西，一旦悟道，那么器就显得那么渺小了。

对于 DBA 也是如此，很多朋友整天都在追求某个独家秘籍，比如一个内部才有（Internal ONLY）的文档，或者一个脚本、工具，而如果你让他去认真读读 *Oracle Concepts*，他就会很不耐烦。前些天，公司的一个小伙子要推荐一个工具给老储，看老储没什么反应，就着急地跑过去和他解释这是个什么东西。老储也是个很直白的人，他说一般我们会根据原理来分析该如何处理，再去查看资料，看看具体的命令，平时没必要去记这些东西。我觉得老储这话说得很好，由道御器是再顺理成章不过的事情了。一旦你能够入道了，那么“形而下”的东西就太好办了。可能我并不知道某个著名的脚本，但是一旦我入了道，那么要做类似的分析，就很容易写出类似的脚本。而如果你并未掌握这个问题的相关原理，那么哪怕你拿着无数精妙的脚本，也可能分析不出什么东西。

很多DBA在学习过程中，总是会不断地追求些解决问题的具体办法，一本好书、一个脚本、一套工具软件，都会让我们感到兴奋。而实际上，这些都属于“器”的范畴，如果把它们作为我们悟道路上的一些工具和手段，那么是无可厚非的，不过你如果把追求这些当成自己追求的终极目标，那就舍本逐末了。我们不应该花太多的精力去追求这些东西，而应该把更多时间用在感悟Oracle本身的“道”上。公司的一个同事总是会发给我一些新奇的脚本或者一些不错的书籍，不过我一般都会拒绝。我会告诉他，你如果有时间就认真阅读一下，我没有时间看这些东西。这些脚本或者书籍都是很不错的，如果是10年前，我会很有兴趣认真地阅读。而现在，我最需要认真阅读的是Oracle的官方文档，从那里，我能够了解到最原汁原味的东西，从中感悟到更多的Oracle的本质。以我的理解，你如果已经掌握了某种方法，那么类似的方法哪怕有几十个上百个，大多数对你来说也是没有多大价值的了。如果你已经习惯于用某种工具去完成一个工作，那么其他的工具对你来说再好也是多余的。如果你总是在不停地学习一些新的工具，一些似是而非的新东西，而你就可能没有足够的时间去独立地思考和感悟。

因此，在DBA成长的过程中，除了如饥似渴地阅读，认真地参与实践之外，多花些时间来思考一些形而上的东西，对于突破瓶颈，尽快“入道”是很有帮助的。当你要去安装一套11*g* RAC的时候，你会怎么做呢？是马上去网上搜索一个攻略呢，还是认真阅读一遍Oracle官方的安装指南呢？我想绝大多数DBA会选择前者。根据攻略，你可以十分快地完成安装工作，而如果要去阅读安装指南，可能要花上一两天来认真思考和体会。这两者的效果区别是十分明显的，一个是别人悟出来的东西，你直接就可以拿来使用，很快，不用费心思；另外一条路是你自己去感悟一些东西，去解决一些问题，因此可能要花费更多的时间和脑力。不过，如果你真的通过认真阅读安装手册，经过多次实践，充分掌握了安装中的一些过程，那么你今后再做11*g* RAC安装的时候，哪怕碰到再多的困难，也可以自行解决，而采用第一种方法的朋友恐怕碰到一点点问题就只能再去谷歌或者别处打听了。

去年，一个网友打电话给我，咨询一个11.2.0.2 GRID的安装问题，我们排除了那个著名的思科交换机问题后，这个问题还是没有解决，后来我建议他跟踪一下root.sh这个脚本，搞清楚到底在什么地方有问题。最后这个问题终于找到了，是由于网卡不稳定导致的。不过通过这次安装，他彻底地分析了root.sh中的每个步骤的细节，他觉得今后再碰到root.sh出现的问题，就十分有信心了。

在学习过程中，不要过于依赖快餐式的攻略，多静下心来，认真思考和感悟一下Oracle的本质，对每个DBA来说都是十分有意义和有价值的。只有通过御器，从而入道，才是最佳的路径。这是陈老师的讲座给我的启示，我也希望我对此的感悟能够传递给每个读者。

白鳍

2012年5月19日于南京

前　言

写完《Oracle优化日记：一个金牌DBA的故事》和《ORACLE RAC日记》后，很多网友问我下面是否会继续写书。我的想法还是和以前一样，先整理整理自己的思路，写一些东西发在Oracle粉丝网上，写过一段时间再根据已经完成的内容决定新书的结构。开始我是想把新书写成DBA日记系列第三部的，不过写作的过程中才发现这种模式写下去有一点千篇一律的感觉了，很多案例从根本上看十分相似。有一次和同事老储聊天的时候，他提到现在很多年轻人不会按照Oracle的内在原理去考虑问题，从而导致经常出现常识性的错误。他的这句话就像火花一样在我脑海中闪现，Thinking in Oracle这几个英文词汇就出现在我的脑中。如果每个人都能以Oracle的基本原理为依据去考虑问题，那不是很好吗？老储的英文名字是John，在我的朋友里有两个John。一个是老外John，我还在玩ICQ的时候认识的网友，一个Oracle技术狂人。不过老外John现在已经成为一家银行的IT技术主管，随着岁月的流逝，当年的技术狂人现在已经成了狂热的人文主义者。去年圣诞新年假期他刚刚完成了10000公里行程的中南美洲自驾游，所到之处全部入住当地最高级的酒店，吃当地最昂贵的美食，还常常邂逅美女，虽然邮件还只是寥寥数字，但是羡慕嫉妒恨已经让我把这个资本主义的老家伙好好骂了几天。另外一个John就是老储，他属于睡在我对面的弟兄，我大学时的室友。自从1995年我把他从unixware汉化小组忽悠到深圳后，我们一直在一起合作。老储是个很低调的人，低调到第一次和客户打交道的时候客户可能会质疑他的能力，不过随着和他交往的深入，你可以发现他的深不可测。

老储的建议让我重新定义了本书的结构，把它分为基础知识、工具、方法和案例四大部分。不过随着写作的深入，我发现这个工作是十分艰巨的。这样一本书结构之大，内容之庞杂，远远超出了我的想象。于是我重新调整结构，将工具这一部分的内容从书中拿掉，准备独立成书，把方法和案例用两章讲完。这样调整后本书的内容更为紧凑，篇幅也可以控制在450页左右。

更大的惊喜也接踵而来，我终于说服了老储，让他参与到本书的写作中来。其实第一次和老储沟通这本书的时候我就邀请过他，不过被他以没空为由拒绝了。老储是一个懒散的人，我每次给他布置写文档的任务，他总是在快到期的时候才开始动手，不过每次提交的东西都让人无可挑剔。能让这么靠谱的人参与本书的写作，确实可以为书增色不少。老储凭着深厚的开发功力，在性能优化方面具有很敏锐的观察力，往往能够在很短很短的时间内找到系统的关键问题，而且他在Oracle文件结构和ASM的结构方面的研究很深，自己也编写了一个类似DUL的工具。有了他的加入，关于ASM原理和数据文件结构这部分内容我就可以推给他了。此前我正为这两节犯愁，考虑是否在书中去掉这两节，补充一些其他内容。老储的加入使这两节保留了下来，对于数

据文件结构和 ASM 文件结构感兴趣的朋友可能会感到庆幸，保留这两节也使得本书的内容更加丰富。

我给本书起的名字是《Thinking in ORACLE 之 DBA 的思想天空》，主编觉得这是一个十分霸气的名字。实际上，透过某个事物的本质去看问题，无论针对什么，都是比较高的境界。对于某些事情，在没有弄清楚其本质之前，我们往往难以找到正确的应对方法，虽然偶尔我们会像瞎猫碰到死耗子一样歪打正着，但是好运气不会总是伴随着你。就像前些年，我不了解“回南天”的成因，因此在每年的 2、3 月份总是十分痛苦。在广东沿海生活过的人都知道每年的春天总是会碰到回南天，时间有长有短，至少也在半个月左右。在回南天里，家里到处都是湿漉漉的，地上、墙上甚至天花板上都会渗出水珠。在这样的环境中生活半个月，绝对是十分恐怖的事情。为了让家里尽快干起来，我的第一反应是门窗大开，同时用风扇拼命扇风。不过这样处理并不能减轻返水的现象，有时候反而水更加多了。后来我请教了一位搞大气海洋研究的人，他告诉我回南天的成因是春天东南季风带来大量的水气，而当气温回升的时候，室外气温高于室内气温，湿热的空气遇到室内较冷的物体时，就会发生冷凝现象，从而就引发了反水的现象。一旦了解了回南天的成因，就很容易找到对付回南天的办法了，只要碰到气温大幅度回升的天气，就门窗紧闭。靠着这个办法，我终于摆脱了回南天的困扰，无论门外的走廊湿成什么样，我家的地上总是干干的。后来每当我看到朋友家里满地积水的时候，就会把这个方法教给他们，他们也逐渐远离了回南天的困扰。

从这个生活案例中，我们也可以看到，一旦了解了问题的本质，就很容易找到正确的解决方法。而没有理解问题本质的时候，我们所采取的应对措施不一定是靠谱的。

这个原则应用到 Oracle 数据库方面，也是一样的。对于每个来应聘 DBA 的人我都会问他们一个问题：“Oracle 到底是什么？”有些人会用数据库基础的理论来回答我：“数据库是数据的集合。”也有些人会感到茫然，不知道我问这个问题是什么意思。实际上很多 Oracle DBA 从来没有思考过这个问题。“Oracle 就是 Oracle，是一个产品，还能有什么意思呢？我不知道 Oracle 到底是什么也没有影响到我做一个合格的 DBA。”很多人都会这么想。

实际上对于 Oracle 我们确实还需要重新去认识认识，每个 DBA 在学习 Oracle 的时候都往往注重于学习如何建库、如何管理、如何编程、如何优化。我们总在不停地去学习一些方法，学习一些秘籍。如果偶尔学到了一些不传之秘，都会感到兴奋异常。也有些人使用这些秘籍解决了一些疑难杂症，成为了大家传说中的高手。

虽然说这也是学习 Oracle 数据库最为常见的一种方法，但是这样学习下去，我们总是在记忆一些枯燥的语法和脚本，虽然经过数年我们积累下了大量的经验，但还是无法真正地理解 Oracle，数据库升级了，系统变化了，我们就必须从头去学习。常年累月，我们总是在一次一次循环往复地重复着同样的事情，直到筋疲力尽，对 Oracle 失去往日的激情，最终 DBA 只是一个职业，Oracle 只是我们谋生的手段。这样学习下去，几年后，很多人就会碰到瓶颈，虽然说自己处理问题的能力和工作经验已经很丰富了，但是技术好像停滞不前了。我周围一些做了五六年 DBA 工作的朋友都遇到过类似的情况，他们咨询我的时候，我告诉他们，这是因为经验的积累已经到了一定程度，需要对 Oracle 基础概念有更深刻的认识。这种情况下，你需要静下心来认真看书，学习 Oracle

的基础概念，只有彻底搞清楚了这些，才能跨过这道坎，达到一个新的境界。绝大多数工作了五六年的朋友已经无法静下心来做这些事情了，因此他们失去了突破的机会。不过也没关系，大多数人选择了新的职业规划，从事管理，或者转向售前、业务专家等职位。

事实上，我们可以换一种方式来学习 Oracle，让 Oracle 的精神融入 DBA 的血液中，让 DBA 像 Oracle 一样思考问题，使 Oracle 成为我们的爱好，作为我们生命的一部分存在。对于大多数 DBA 来说，这也许只是一个乌托邦式的理想，多数 DBA 只是需要有一份工作，需要靠这份工作来生存，娶妻生子，享受生活。并不是所有的人都希望让 Oracle 成为自己生命的一部分，这是很现实的，不过我们虽然可以仅仅把 Oracle 当做是谋生手段，但也还是可以同时尝试了解 Oracle 更多的本质，像 Oracle 一样思考。

对大多数人而言，像 Oracle 一样思考虽然不能带给你更多的生活乐趣，但是通过以这样的方式去学习和思考，会更加精确地了解 Oracle 的精髓，让自己在 DBA 的成长过程中少走弯路。10 多年前我第一次接触 Java 的时候，感到十分头痛。不是自夸，10 多年前，我是一个相当不错的 C 程序员，最高纪录是一天之内编写 500 多行复杂的代码，而且一次性编译通过，一次性测试通过，这样的记录的诞生是基于十分良好的过程思维能力的。不过我这个自封的编程高手第一次接触 Java 的时候，却感到十分吃力。我无法用面向对象的思想去编写程序，所以学起 Java 来十分痛苦，几次学习，最后都放弃了。直到有一天我看到了一本英文的书籍 *Thinking in Java*，通过这本书，我掌握了 Java 和面向对象设计、编程的主要思路。自从看了这本书之后，我再次面对 Java 程序的时候，发现一切都是那么地简单，很快我就掌握了 Java 编程。现在我虽仍然还只是一个三流的 Java 程序员，不过粉丝网的一些修修补补的工作我完全能够胜任了，而且在和一些 Java 开发人员交流的时候，我也能够很快地理解他们的思路。

后来我总结了一下，在看 *Thinking in Java* 这本书之前，我在编写 Java 程序的时候，并没有理解面向对象编程的概念，只能是照猫画虎，拿着一个例子在上面修改。实际上我的编程风格还是面向过程的，因此写出来的代码质量很差。而通过阅读 *Thinking in Java*，我终于学会了面向对象的方法，用 Java 本身的思想去考虑问题，因此能够更加准确地抓住问题的本质。我想，学习 Oracle 数据库也是这样，如果我们通过一个又一个案例去学习 Oracle，那么将永远停留在表层上。有些 DBA 只能重复相同的案例，这样的 DBA，哪怕干上 10 年 20 年，也可能只学到 Oracle 的一些皮毛，碰到一个没有见到的案例，可能就会感到手足无措。而水平高一些的 DBA 往往能够判断出案例的相似性，并通过分析找到类似案例的解决方法，这其实就是因为透过现象看到了问题的本质。

很多 DBA 可能都碰到过我下面所说的一些问题，有时候我们无法评估某项调整可能对系统带来的影响，有时候我们面对一个复杂局面的时候很难快速找到问题的关键，也有时候我们在为解决某种等待事件而感到无从入手。实际上，遇到这些问题，都是因为缺乏对于 Oracle 内部原理的充分认识。在很多情况下，当经验无法为我们提供足够支持的时候，就必须从原理出发进行思考，才有可能真正掌握问题的根源，从而解决问题。

前几天我在一个客户现场做数据拯救工作的时候，他们的备库（也就是现在的主生产库）突然宕机了，当时客户正在做一个删除临时文件的操作，领导就认为他这个操作导致了宕机，而操

作人员也觉得很冤枉，因为这是一个十分常规的操作。我看了看日志文件，从日志上看不出任何由于临时表空间和临时段操作引起的问题，同时又看到了一个好像是人工操作停库的信息，于是推断可能是人为操作所致。后来经过多方面查证，确实是有个 DBA 在家里远程做维护的时候，发现操作 HANG 住了，情急之下，直接重启了数据库。如果你不了解临时段和临时表空间操作的原理，面对这个问题，很可能一上来就把重点放在删除临时文件导致宕机问题的分析上面，这样就偏离了正确的方向，解决问题的效率和成功率就会大大降低。

我们强调理论的重要性，也不是片面强调理论而不重视实践。Oracle 数据库是实践性很强的，没有实践，光学习理论是无法成为真正的高手的。比如说我们学习了很多 OWI 相关的理论，了解了数据库等待事件和一些状态指标的含义，但是我们看到一个库的 AWR 报告的时候，还是无法知道某个指标是否正常。当对大型 OLTP 系统缺乏实践经验的时候，我们就无法知道大型 OLTP 系统的一些技术指标的特性，因此也很难从中找到疑点，进而找到解决问题的方法。

这些年里我接触过大量的 DBA，我一般把他们分为四大类。第一类 DBA 是经验型的，处理问题的主要方式取决于以往的经验，他们往往都有很好的习惯，会把每一个处理过的案例整理出来，今后再碰到这类案例的时候，他们会很快地解决问题。随着工作时间的增长，他们的技术也会相应地提高。第二类 DBA 是理论型的，他们具有很深的理论基础，经常探讨一些“Oracle Internal Only”的高深问题，比如他们能够很清晰地告诉你共享池分配的算法，告诉你 checkpoint 的工作原理，但是这些 DBA 往往缺乏实际的工作经验，他们研究 Oracle 却很少有机会接触大型的数据库系统，因此实际解决问题的能力并不强。另外，由于他们的知识比较片面，在某些方面很深入，而某些方面就是浅尝辄止，这种不均衡导致他们的知识只是以点的形式存在，无法串成整体，因此那些很深入的研究并不能给他们实际工作带来多大的帮助。第三类 DBA 是技巧型的，他们并不注重理论的学习和经验的积累，在处理问题的时候往往能够利用 Metalink 和谷歌、百度之类的工具去搜索解决方案，这类 DBA 最为常见。他们处理问题往往靠运气，而且一些他们自鸣得意的成功案例往往也是经不起推敲的，下回碰到类似案例时，可能还会失败。第四类 DBA 是虚心请教型的，他们无论碰到什么问题，甚至连错误信息都没有看明白，就开始叫“我的系统出问题了”，然后到处去问如何解决。

实际上，这四类 DBA 都是有缺陷的。第一类 DBA 可能经过多年的工作，有十分丰富的经验，处理问题的能力很强，而且分析问题十分敏感，很容易抓到问题的关键，但是由于没有深入理解 Oracle 的理论，碰到一些较为深入的问题的时候，就不容易立刻找到关键。虽然凭借着自身丰富的经验和问题分析排查能力，他们最终也能解决大部分的问题，但是往往问题解决后还是没有真正弄明白为什么会解决问题，下一次碰到类似的问题，可能还是要花很大的代价。

第二类 DBA 在某些方面的理论知识很强，总是喜欢研究一些十分高深的原理性的东西，但是这类 DBA 的主要精力都放在了研究一些 Oracle 内部原理上了，他们没有很多的时间去实践他们学到的理论。这类 DBA 往往知识面较为狭窄，仅精通于自己研究比较深入的领域，在实际工作中也很难发挥出自身的理论研究特长。

第三类 DBA 实际上在我们的现实生活中是最常见的，“万事不明问百度，百度不明就抓瞎”，确实谷歌、百度和 Metalink 能够帮助我们解决不少问题，但是这类 DBA 往往在问题解决后没有

好好思考一下，为什么这个方法能够解决问题，更没有认真总结和归纳一下，下一次碰到类似的问题，还是无法依靠自己的思考去解决问题。于是再 Google 一把，也许这一次运气没有那么好了，Google 出来的资料不是上回的那个了，于是结果可能是很悲惨的。

第四类 DBA 在我们现实生活中也经常出现，网络社会通信十分发达，打个电话或者在 qq 群里、msn 里问问，也许就有人帮忙解决问题。久而久之，这些人放弃了自己的思考，碰到一点点小问题都要找人问。缺乏独立思考问题能力的 DBA，只能称为一个数据库操作员，实际上离真正的 DBA 还有十万八千里呢。

看到这里，大家可能明白了，老白实际上说的不是四类 DBA，而是 DBA 的四种性格，这四种性格可能会集中在某一个人身上。以老白学习 DBA 的经验来看，理论结合实践是十分重要的。在 2000 年前，老白虽然做了很多项目，也是很多人眼里的 Oracle 数据库高手，但那时的老白就是第一类 DBA 的典型，没有经过多少理论学习，几乎所有的 Oracle 数据库的技能都是从实践中获得的。虽然在实践中我总结出大量的经验，甚至有很多客户建议我写一本书，把我对 Oracle 的理解写出来，不过当我自信满满地开始写书的时候，却突然发现，我的一些知识需要进行确认，否则写出来就贻笑大方了。于是我开始大量地学习 Oracle 的一些理论知识，随着写书过程的深入，我越发感到自身理论水平的不足。《Oracle 数据库深度历险》这本书我写了 3 年，实际上 2002 年就彻底放弃了出版这本书的念头，因为我发现自己的理论知识确实还需要进一步的梳理。但是我并没有放弃写作，因为我发现通过写作，我更为系统地将 Oracle 的理论知识梳理了一遍，这次梳理是通过我以前的知识体系、工作经验，与 *Oracle Concepts* 的理论基础进行了一次完整的整合。通过这 3 年的写作，我终于完全疏通了自己的 Oracle 的理论体系，好像一个练武术的人，终于打通了任督二脉，感到无比的畅快。

听老白说了这么一大通，是不是很多人都感觉到手脚发凉，难道成为一个合格的 DBA 有这么难吗？如果我没有打通任督二脉，就不算一个合格的 DBA 吗？实际上 DBA 成长的道路是很多的，并不一定要走老白这一条路，老白仅仅是根据自身的经历，通过这本书来帮助大家梳理 Oracle 的一些基础知识而已。还是那句话，如果 Oracle 是你的爱好，那么你无论花多大代价去研究它都是值得的；如果 Oracle 只是你职场生涯中的一份工作而已，那么只要你认真对待它就可以了，没必要像老白那样执著。

作为一个 DBA，理论学习和实践如何相结合是十分关键的。在初期，一般来说 DBA 都是通过了某种途径接触了 Oracle 数据库，进行了一系列的操作。在工作过程中发现了一些问题，才开始想到需要去看一些 Oracle 的书籍。在这个阶段，Oracle 官方文档的 2days、7 days 系列入门书籍就十分有效。通过这些书籍你可以了解 Oracle 的一些基本的原理和基本的操作，帮你在工作中解决部分问题。这样你在工作中就能够应对一些简单的问题了。不过碰到稍微复杂一些的情况，你可能还是会发懵，这时，*Oracle Concepts* 这本书就十分关键了。从这个阶段开始看这本书是十分必要的，它有助于你在积累经验的过程中不断地完善理论。不过，你可能还无法完全理解 *Oracle Concepts* 中的基本概念，通读这本书是十分必要的，但是不必要把每个问题都搞得十分清楚。因为要达到这一点，你需要花费太多的时间和精力，同时也可能会由于缺乏足够的技术指导而无法真正理解问题的本质。不过在这个阶段，碰到某些问题或者研究各种案例的时候，经常翻翻 *Oracle*

Concepts 这本书是十分有益的，因为在处理问题的时候，你针对这个问题的思考会比较深入，这个时候，认真分析一下相关的理论是十分有效的。对于处理过的每一件事情，都做一个比较详细的记录，是十分好的习惯。记录下某个案例，可以供水平提高后再进行回顾，或者将案例提交给某个专家去评审，或者在网上和大家一起讨论，对于学习 Oracle 的原理都是十分好的方法，可以帮助你在分析案例时提高对 Oracle 数据库原理的认知。

在这本书里，老白会把《Oracle 数据库深度历险》中的一些内容，结合老白的实际工作经验展现给大家。我会剖析原理，并结合案例来说明这些理论知识如何在实践中应用。希望老白的这次写作经历，能够给大家带来一些帮助。

阅读本书的建议

本书还是一本介绍方法的书，并不是一本系统介绍 Oracle 知识的百科全书，因此读者在阅读本书时应该注意方式方法，需要注重对基础概念的理解和对工作思路的理解，而不要拘泥于某个技术细节。本书并没有涉及任何技术细节，关于技术细节，读者可以参考 Oracle 相关的官方文档，比如 *Complete Reference*、*Oracle Concepts*、*Oracle Performance Tuning Guide* 等。实际上，老白认为 Oracle 的官方性能优化手册是我看到过的最好的性能优化方面的书籍，没有之一。如果你认真研读过这本技术手册，就会发现你以前看到过的关于 Oracle 性能调优的书籍的核心内容，只不过是对这本书的另外一种阐述而已，可能会增加一些例子，可能在某些细节上会更加细致，仅此而已。

本书中的一些观点仅仅是老白自己这些年对 Oracle 的理解，并不是金科玉律，可能有一些思路和方法是有一定局限性的，甚至有些或许是错误的，因此读者不能盲从。结合自己的知识体系，重新认识这些思路和概念，使之成为自己的知识体系的一部分，并用于指导你建立自己的问题分析、预案处理体系，才是最为根本的。如果某些案例或者某些观点你不太认同，欢迎大家到 www.oraclefans.cn 上去和老白探讨。有互动的阅读，可能会对你的帮助更大，也有助于老白纠正自己的一些错误。

本书中的一些例子都是比较容易去实践的，老白其实仅仅使用了一套 Oracle 10*g* 的数据库、一个带 sql*plus 的 Oracle 客户端，再加上 profiler 工具，就完成了本书中绝大多数实验。希望有兴趣的朋友可以亲手去做一做这里的实验。如果大家觉得里面的脚本自己敲出来很麻烦，可以到 www.oraclefans.cn 上发帖向老白索取。不过老白觉得，如果你能看懂这些脚本，并自己再写出来，那么这些脚本就真正成为了你自己的工具。而从网站上下载下来的工具，可能很快就会被你丢弃在一边，过几天就忘记了。

目　录

第一部分　基础原理篇

Part 1 第一部分

基础原理篇

实际上对于 Oracle 我们确实还需要重新去认识。很多朋友都会游泳，那么我来问一个简单的问题：什么叫做真正地学会了游泳？每个学游泳的人都有这样的感受，刚刚学习游泳的时候最大的问题是无法浮在水面上；经过努力，发现只要手脚按照一定方式滑动，身体放平就能浮在水面上了，这个时候就感觉从不会游泳变成会游泳了。但是这时候你会面临另外一个问题，就是不会换气，学会换气后再往后学习就容易了很多，不过可能我们游上几十米、百把米就会觉得很累。后来我们发现，原来我们游泳的姿势还不太标准，而如果我们学会了踩水，游多远都不是个问题了，这个时候我们就如鱼得水了。学习 Oracle 其实和学习游泳十分类似，刚刚开始的时候，我们处于入门阶段，要克服对 Oracle 的恐惧心理，只要我们想学好 Oracle，我们就一定能做到，实际上 Oracle 在入门阶段是十分容易的。我们从头学习 Oracle，学会了安装数据库，学会了管理表空间，学会了日常的故障处理。就像我们学游泳的时候刚刚学会换气一样，虽然感觉已经掌握了 Oracle 数据库，但是碰到稍微复杂一些的问题我们还是觉得很难入手。我们能看清楚一些东西，但是感觉还是看不远。

上面这种情况是每个初学者都会碰到的，学 Oracle 和学游泳一样，刚开始的时候我们只是学会了游泳的外部表现，学会了像别人一样划水、换气，但是并没有真正掌握游泳的本质。如果要消除初学期的迷茫，我们一定要真正地掌握 Oracle 数据库的概念，一个 DBA 如果连认真学习 Oracle 的基本概念都不能做到，那么我觉得他也很难成为一个真正的高手。

这些年我在网上和大家一起讨论 Oracle 的问题，也经常有朋友问我，要成为一个优秀的 DBA，应该看些什么书。我给大家推荐的第一本书就是 *Oracle Concepts*，这也是我唯一能够推荐给所有人的书。这本书确实适合任何一个 DBA 阅读，无论你是初学者还是已经工作了很多年的高手。只要你没有认真读过这本书，你就有必要去读一遍。

理解 Oracle 的基本原理有助于你用一种十分理性的思维去考虑问题，在处理问题的时候能够更快地抓住问题的本质。我们大家在从事 DBA 工作的时候，经常会碰到这样的情形，一个问题困扰了我们很长时间，突然猛地一下，你就抓住了问题的关键，然后它就迎刃而解了。而在这之前，我们可能会做过很多尝试，这些尝试有些是有序的，有些是无序的，甚至有些只是瞎蒙。在这种情况下，有时候经验是帮助不了你的，因为所有根据经验的分析都已经做完了，并且被证明是无效的。这时，我们就只能根据自己对问题本质的理解去分析，才可能最终解决问题。

事实上，要想像 Oracle 一样思考，首先就需要了解 Oracle 的本质是什么，它是怎么运作的，在运作过程中，哪些地方可能成为瓶颈。这个时候可能就有朋友感到疑惑了：我们如何了解 Oracle 内部运作的原理呢？这看似一个不可能完成的任务。确实，在处理有些问题的时候，我们可能要了解 Oracle 很深入的算法，但是在绝大多数情况下，我们分析问题只需要了解一些我们在 Oracle 官方文档中提及的原理性的东西，并不需要深入到算法和源代码级别。比如说，如果碰到一个共享池的性能问题，你会马上联系到共享池、库缓存（library cache）、字典缓存（row cache）、CURSOR，以及相关的一些闩锁、参数等，在大多数情况下这些就可以支撑你做分析了。但是必须把这些知识点编织为一张网，而不是一个一个的知识点。对于大多数 DBA 来说，想把这些知识点编织为一张网是比较难的。这需要你花费大量的时间来学习，而很少有一本书能够从这个角度去介绍知识。因此这种学习会比较困难。在本章，老白将尝试以这种方式来介绍相关的知识点，帮助大家从知识点升级为一个立体的知识体系。

另外，把知识点融会贯通不是看书就能达到的，要想达到这个目标，必须进行足够的实践活动。你只有在把学到的知识应用在实践中，才可能突破知识孤岛，达到新的境界。因此，如果你看完第一部分内容后，有些知识点还感觉无法掌握，这也并不要紧，结合自己碰到的案例思考一下，也许会更有收获。

第一篇的大多数章节都是用“理解”开头，因为这部分是以理解 Oracle 基本原理为主要宗旨的章节。通过本篇，老白希望大家能够对以前有一定认识的 Oracle 认识得更为深入，那么你买这本书就不冤枉了。