

TURING

Oracle优化日记

一个金牌DBA的故事

白鳝 编著



- 金牌DBA精彩纷呈的经历
- 86个优化技巧活学活用
- 培养多种能力，保障职场成功

 人民邮电出版社
POSTS & TELECOM PRESS

TURING

Oracle优化日记

一个金牌DBA的故事

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Oracle优化日记：一个金牌DBA的故事 / 白鳧编著.
—北京：人民邮电出版社，2010.7
ISBN 978-7-115-23071-3

I. ①O… II. ①白… III. ①关系数据库—数据库管理系统, Oracle IV. ①TP311.138

中国版本图书馆CIP数据核字(2010)第096986号

内 容 提 要

本书是一本介绍 Oracle 数据库优化方法的书，以一个实际的大型优化项目为原型，用日记的形式记录了一个优化小组的 DBA 如何从纷繁的头绪中找到突破口，进而完成了一个看似不可能完成的任务的历程。在日记之间，作者还穿插了优化小技巧、每日点评等，总结了数据库优化常用的方法、工具和技巧。另外本书第一次详尽地披露了 Oracle 数据库内部存储结构，并公布了部分代码，对于有兴趣研究数据库内部存储结构或者编写 dul 工具的读者有一定的参考价值。

本书适合 Oracle 数据库管理开发人员阅读。

Oracle优化日记：一个金牌DBA的故事

- ◆ 编 著 白 鳧
责任编辑 王军花
 - ◆ 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 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本：800×1000 1/16
印张：22.75
字数：538千字
印数：1-3 500册
- 2010年7月第1版
2010年7月北京第1次印刷

ISBN 978-7-115-23071-3

定价：55.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223

反盗版热线：(010)67171154

给初学者的建议

为什么写 DBA 日记

我的 QQ 网名叫“老网虫白鳝”，很多网友都问我为什么起了这么土的一个网名。实际上，这个网名应该是 1998 年的时候起的。10 多年前，我是个十分喜欢玩些新鲜东西的人，ADSL 刚刚在中国出现，我就成为中国第一批使用 ADSL 的宽带用户，时间好像是 1998 年。通过宽带上网确实很快，也拓宽了我在网上的视野，对于用惯了拨号网络的我，第一次感受到了飞一样的上网感觉。有一天中午，和几个朋友一起吃饭，说起了 ADSL，朋友很是羡慕，不过对于我每天只是上网看看新闻，在四通利方上打打嘴仗，感到很是不屑。他建议我去试试 OICQ。在注册 OICQ 账号的时候，我想继续使用在四通利方上的网名“水鱼”，不过“水鱼”很容易让人产生不愉快的联想，于是我准备换一个网名，由于双鱼座的我对水产品的热爱，我想起了中午的午餐“白鳝”。“白鳝”也就是我们常说的河鳗，广东人称之为“白鳝”。结果，QQ 上就多了一条“白鳝”。光阴荏苒，突然有一天我发现我的网龄已经接近 10 年了。于是把网名又改成了“老网虫白鳝”。说实在的，没有“老网虫”，就没有作为 DBA 的“白鳝”。是互联网让我从一名仅仅会安装 Oracle 的入门者成长为一名资深的 DBA。

在网上和网友讨论 Oracle 已经有 10 年了，很多网友问我为什么不写本书，其实我也一直想写本关于 Oracle 的书。2002 年，我产生了把来自 Metalink 的学习成果写出来的想法，书名都起好了，叫《Oracle 深度历险》。书编写了一年多，Word 文档算下来也有 1000 多页。在 2004 年我校对这本书时，发现其中的大多数内容都包含在市面上相关图书中，出版价值并不大。虽然第一次写书很失败，不过写一本书的想法一直没有磨灭。但因工作繁忙，很少有大块空闲时间用以写作，因而也难于写出一本连贯性很强的书来。当年看王强的《圈子圈套》时一下子被迷住了，推荐给很多朋友，看过的人都说在里面能够看到自己的影子，我当时看《圈子圈套》的感觉也是如此。说起来和王强还有过一面之缘，根本就没把他和作家联系起来，但是他的作品在 IT 圈子里的人看来，比作家写得更深入。因为这是他在 IT 圈子里摸爬滚打的经验总结。看过《圈子圈套》后，我开始写《IT AND I》。王强是从一个系统集成行业的高层人物的角度去看问题，而《IT AND I》里的莫明是这个圈子里处于底端的工程师。现在《IT AND I》在我的另外一个博客 (blog.oraclefans.cn/baishan2) 里连载，不过最近也已经很长时间没有更新了。我是一个比较懒散的人，也不想给自己太大的压力，只是想把我这些年里做 DBA 的一些经验写出来，共享给大家，所以我决

定写这本书。开始写的时候，我的初衷还是自娱自乐，并没有出书的打算。

直到有一天，我和我的一个同学在北京相聚。他以前在 BEA 公司，在 Oracle 收购 BEA 以后，他成为 Oracle 一个售前部门的总监。我说了一些 DBA 圈子里的事情，他也介绍了 Java 圈子里的一些事情。Oracle 圈子和 Java 圈子全然不同，Java 圈子是一个十分开放的圈子，由于 Java 的开源性，整个 Java 圈子都十分开放，大家都以把自己的工作成果开放出来给大家分享为荣，所以 Java 技术发展十分迅速，技术方面的创新层出不穷。我想，Oracle 圈子只有开放了，才可能像 Java 圈子那样欣欣向荣。从那次开始，把本书公开发表的想法就逐渐形成了。这也是我这一次重新修订《Oracle 优化日记》的一个主要目的。既然目的是正式出版，那么书中的很多内容就不能太随意了，很多观点也要再三推敲，免得误人子弟。

现在大多数的 Oracle 图书都是以技术为主，而且品种十分丰富，但是对于 DBA 来说，要学的不仅仅是技术，还有很多东西不是仅仅通过技术传授所能学到的。作为一名优秀的 DBA，需要具备一些气质、一些性格以及一些处事的方法和原则。这些都不是纯技术的问题，但是往往和我们的 DBA 生涯关系重大。我写这本书的初衷也是为了把我和其他朋友 DBA 生涯中的一些故事介绍给正在学习或者使用 Oracle 的 DBA 们看。本书不是一部小说，因为里面将会介绍很多 DBA 的知识和技术。但是本书也不是一本纯粹的技术书，因为其中穿插了很多故事情节，除了那些风花雪月的事情，其他情感几乎都会在日记里体现。DBA 从事的是一种职业，在职业生涯里也会有喜怒哀乐。除了技术以外，我想我也应该把这些喜怒哀乐传递给大家。

为了叙事方便，我会把发生在很多人身上的事情集中在“我”一个人的身上，“我”不仅仅代表了白鳍，更代表了一批奔四的老 DBA。为了避免一些法律问题，部分客户会使用化名或者故事情节会有所演绎。本书总的来说还是一本技术性图书，并不针对某个人或者某个企业，因此请 DBA 或业界朋友不要对号入座。

另外，十分感谢储学荣、熊文华、罗时峰、邓智国等为本书提供的技术资料 and 精彩案例，感谢毕建波、薛军和吴刚为本书提供的实验数据，也十分感谢彭国秋、段小真、段小成对本文中文字的校对。当然也要感谢老方、老干、老肖、老熊和小齐，我至今还很怀念几个人在一起喝酒谈论 Oracle 的日子，没有数年前我们的那一次合作，就没有今天的这本书。另外，远在澳大利亚的 Ben 和美国的 John 也给了我极大的支持。在我写《Oracle 深度历险》的时候 Ben 也给予了我极大的鼓励，并且在 2002 年很诚恳地告诉我这本书已经落伍了。Ben 希望我写一些更深入的更有价值的东西。我终于完成了对 Ben 的承诺。

谁适合看这本日记

本书最早是在我的博客上连载的。连载期间，有些朋友产生了共鸣，有些朋友说看不太懂，有些朋友说深受启发。那些产生共鸣的人大概也是资深 DBA 了。本书是比较真实的，虽然也有一些艺术夸大的成分，但也都是有现实依据的。

那些看不懂的朋友，还是没有理解我的意思，实际上本书就像流水账一样记录了一些 DBA 日

常的工作。如果你觉得看不懂，可能是你还没有碰到过那种情况，你不需要理解其中的每个技术细节，看不懂的地方完全可以跳过，这并不妨碍你看其他章节。（当然书中对有些问题的处理过程写得过于简单，读者不容易看懂，因此在为出书而修订时，我已经将这些案例的处理过程细化。另外，对于刚刚入门的 DBA 来说，本书从一个优化案例入手，可能确实会感觉有点不容易摸到头脑。不过不要紧，你完全可以把这本书当做一本写得比较通俗的小说来看，跳过那些生涩的技术描述，提前体会一下 DBA 优化项目时会遇到些什么问题，以及如何面对和处理这些问题。只要你理解了 DBA 分析问题的思路与方法，这本书对你来说就值了。

那些感觉深受启发的人，应该是刚刚进入 DBA 行业的新手。这本书中的很多技术都是这些 DBA 目前正在接触和使用的，而他们又往往缺乏接触大型优化项目的机会。前几天有个网友问我，他正准备接手一个优化项目，能不能给他介绍一下优化项目该怎么做。我推荐他到我的博客上去看看本书。也许和你以前看到的 Oracle 书有点不同，不过书就是书，而不是至高无上的“葵花宝典”。如果你认为看过一本书就能成为高手，那你就错了。

一名老 DBA 的成长之路

结缘 Oracle

从 1993 年第一次帮客户安装 Oracle 算起，我和 Oracle 已有了 16 年的亲密接触。说实在的，第一次接触 Oracle，我对它的印象十分差。在此之前，我只接触过一个大型数据库——DEC 公司的 RDB（现在它已归入 Oracle 名下了）。当时，国内使用的最广泛的小型机平台是 DEC 公司的 VAX，操作系统是 20 年前大名鼎鼎的 OpenVMS。20 世纪 80 年代后出生的人耳熟能详的是 Unix 和 Linux。倒退十几年，20 世纪 90 年代初或者更早的计算机操作系统课程中，很多算法都来自 OpenVMS。那时，在国内使用最广泛的 Oracle 版本是 5.1，而且当时大家的版权意识都比较薄弱，不愿意花几十万元购买正版的数据库，所以当时有一种职业很吃香，即能够帮客户破解和安装系统的工程师。

那时候，我是一名在 OpenVMS 上开发应用的软件工程师，由于工作关系，接触了较多的 VAX 系统。也因当时懂 VMS 和 Oracle 的人十分稀缺，因此常有集成商找我利用周末帮助安装系统。我第一次和 Oracle 的接触就是这样开始的。软件已经破解好，当时有一位老师水平很高，居然写出了一个生成 Oracle 许可证文件的程序，花上 2 000 元就可以买到一个和机器码绑定的许可证，安装介质就是那种 20 年前十分流行的正方形磁带。复制安装介质，编译链接，然后创建数据库，以前的 Oracle 安装十分烦琐，连数据文件都要手工创建后添加到表空间里。当我手忙脚乱地忙活了一天，终于替客户成功安装了一套 Oracle 5.1，并拿到 2 000 元钱的时候，心里别提多高兴了，因为那时候我的月工资不过 1 000 元。虽然钱挣得不算困难，但是 Oracle 给我留下的恶劣印象使我很长时间不愿意接触它。和 RDB 比起来，Oracle 简直太烦琐了，其性能和功能也无法和 RDB 相比。基于这个认识，我在 1994 年帮助泉州电信开发计费系统时，还是全力推荐客户使用 RDB。这个计费系统十分成功，获得了省级的科技进步三等奖。

在这段时间里，我在每个项目中都会碰到大型数据库，要么选择 Oracle，要么选择 RDB，但我更愿意选择 RDB。这期间，Oracle 在不断进步，而 RDB 随着 OpenVMS 在商业上的失败却日薄西山，几年后终于被 Oracle 收购。1995 年，我为一个政府部门设计一套电子单据处理系统时，客户坚持使用开放的 Unix 系统，而拒绝使用 OpenVMS。在 Unix 平台上，Oracle 成了我唯一的选择，当时正是 Oracle 7.1 大行其道的时候。Oracle 7 有了太大的进步，其方便的安装配置以及优异的性能让我感到十分意外。因此，在 1996 年我为泉州电信设计联机实时计费系统时，

Oracle 成了我的首选，正是这个项目让我真正喜欢上了 Oracle。当时使用的服务器是一台有两个 21164 CPU、256MB 内存的 DEC Alpha 2100，虽然今天看来，这台服务器还不如现在的一台普通 PC，但在当时则代表了最先进的 64 位计算的服务器，完成了一个具有上百万市话用户、50 万长话有权用户的大型本地网的联机实时计费系统。在这个项目中，我第一次对 Oracle 进行了优化，通过调优的系统发挥了强大的性能，一个电话在挂机后 5~10 秒就能完成通话话单的结算。后来在这个系统的基础上，福建富士通开发了一个话费回送系统，挂机后几秒钟就可以把通话费回送到客户的来电显示电话上。

从那以后，我和 Oracle 结下了不解之缘。1997 年，我第一次参加了 Oracle Open World，那次北京的盛会除了让我了解了超大内存 (VLM) 和超大数据库 (VLDB)，也让我结识了一批 Oracle 第三方服务的先行者——北京巨龙的朋友，他们在 Oracle 这个产业上获得的成功让我羡慕不已。

真正让我成为 DBA 是在 1999 年后。此前，我虽然和 Oracle 形影不离，不过我的主要身份还是一名系统架构师、一名十分优秀的程序员，数据库安装、维护和优化只是我的副业。从 1999 年开始，由于要为一些客户提供专业的第三方技术支持，我开始认真研究 Oracle 的架构及内部原理。我花了差不多两年的时间，在 Metalink 上阅读了至少 2 000 份技术文档，一个专题一个专题地研究 Oracle 的内部原理，从寥寥可数的文字中解密一些 Oracle 秘不可宣的“隐私”，甚至在一些 bug 报告的晦涩文字中寻找 Oracle 内部原理的合理解释。后来由于工作关系，我接触了不少 Oracle 内部文档。我发现若能早点获得这些文档，那么整个学习过程至少可以缩短一半。我觉得 Oracle 公司应该开放这些文档，让愿意深入研究 Oracle 的人员学习。

这段时间，我经常上技术性网站 ITPUB。刚开始的时候，大家的 Oracle 水平都有限，论坛的学习气氛也十分不错。而随着网站的人气越来越旺，在论坛里大家不再是一起学习和讨论问题了，而是不停地扯皮、争吵，于是 ITPUB 上有一批人转到了 www.oracle.com.cn，我也在这个网站上面混迹了一段时间。DBA 这个圈子保守的气氛使这些网站都很难成为真正高手的园地。在这些 IT 网站上，有价值的内容越来越少，所以我把所有的精力都放到了 Metalink 上。

成长的快乐

经常有 DBA 和我探讨学习过程中的各种迷茫和痛苦。学习 Oracle，在最初的阶段是十分轻松的。Oracle 的技术资料唾手可得，对于初学者来说，这一点十分重要。不过随着学习的深入，很多 DBA 都感觉好像遇到了瓶颈，无论自己多么努力，技术水平想要提高就十分困难了。实际上，DBA 在其成长过程中，所需要学习的不仅仅是技术。现在介绍 Oracle 技术的图书已经相当多了，通过阅读这些图书，DBA 应该能够学到足够的专业知识了。不过大家可能都有这样的感觉，刚开始学 Oracle 时，觉得 OCP 是一道十分高的门槛，总觉得不知道自己需要花上多少时间才能达到那个境界。而事实上，真正想要去考 OCP 认证，花上半年到一年的时间也就足够了。而通过 OCP 认证考试后，自己还是感觉到心里空空的，碰到问题还是找不到解决的方法。

实际上，如果真正认真学习了 OCP 课程，了解了 Oracle 的一些基本原理，并且通过 OCP 考试后，理论上已经具有相当的基础了。只是 Oracle DBA 工作不是考试，OCP 理论也只是一个初

级的理论，如何将这些理论知识融会贯通，实际应用到日常工作去，是十分关键的，这也就是我们常说的工作方法。作为一名 DBA，除了学习理论知识外，工作方法也是十分关键的。因为 DBA 是一个职业，而不是一门课程，理论知识只是基础，只有理论基础是远远不够的。我碰到过很多正在学习 Oracle 的朋友，他们很容易走入两个极端：一种是仅仅注重理论，看了很多书，但总是感觉看书时好像什么都懂了，一放下书就觉得好像什么都没学到，真正碰到问题时还是两眼一抹黑；另外一种觉得读书太枯燥，总是喜欢自己摸索，他们哪怕碰到一点点小问题，都会去寻求其他人的帮助，而不愿意自己去书本里学习。实际上，在一个 DBA 成长的过程中，需要将读书和实践有机结合起来，读书固然重要，不过没有实践操作，就无法巩固书中学到的知识。

不过并不是每个人都有条件能够参加各种实践活动的，特别是刚刚入门的 DBA，他们往往缺少大型项目和大型数据库维护的经验，这对于他们在技术上的提高是很不利的。这时其他人的经验就是很好的教材。本书的目的是把我这许多年在 DBA 工作中碰到的一些典型案例用一种很轻松的方式说出来，让大家在看这些案例时学习到分析问题的方法。如果看本书时，仅仅去关注那些技术性的东西，那就本末倒置了。本书的真正精华是那些像流水账似的东西，所以大家不要把它当做一本技术宝典来使用，其中涉及的技术都是大家在其他地方能够学习到的。大家更应该看到的是我在碰到各种问题时，是如何处理的，是如何把一些很基本的技术运用到这些项目中去的。记得有一次我去一个客户那里做性能分析，分析结束后，给客户介绍的时候，客户说你说的这些技术我都学过，怎么你能说的如此头头是道，而我却无法把这些知识和实际工作结合起来呢？那位客户不久前刚刚通过了 OCM 考试，按理说他的技术水平和实际操作能力应该是足够的，为什么他无法把知识和实际结合起来呢？实际上这就是工作方法问题，也就是本书想要教给大家的。

每个 DBA 的成长之路是完全不同的，我开始接触 Oracle 时，在国内接触 Oracle 的人还很少。当时唯一能够找到的 Oracle 资料除了 Oracle 随机文档外，就只有太极出的那套 VAX 技术书里屈指可数的几本薄薄小册子了。刚开始接触 Oracle 时仅仅是安装 Oracle，最早是 OpenVMS 平台，后来逐渐转移到 Unix 平台：SCO Unix、DIGITAL Unix、IRIX、SUN OS 等。Oracle 5 和 Oracle 6 在性能优化上没有什么可做的，主要是针对 SQL 进行优化，维护管理也较为简单，主要是表空间管理。那时的系统也比较小，一般都在几百兆字节到几个吉字节之间，所以也很少能够碰到 Oracle 的 bug。不过在这段时间里，有较多的机会接触小型机、网络、操作系统和应用开发，这些经历都让我在今后的 DBA 生涯受益匪浅。随着 DBA 工作做得越来越专一，接触数据库以外的工作就越来越少，到目前为止，可能除了数据库以外，其他技术都基本上停留在理论上了。

对于一名初级 DBA，多接触一下其他技术是十分有好处的。随着时间的推移、年龄的增长，学习能力和学习新东西的速度都会有不小的下降。我曾经在赛格计算机公司工作过几年，该公司是国内最早的系统集成商之一，其创始人查树衡多次出现在徐迟老先生的报告文学里，在 20 世纪 80 年代末是类似于当今杨元庆、郭为这样的风云人物。在赛格计算机工作的几年，有大量机会接触系统基础，接触 VAX 小型机、DEC 网络产品和存储产品，甚至还帮助公司“攒”了几台小型机。在很长的一段时间里，这段经历对我都有很大的帮助。

特别值得一提的是，应用软件开发方面的经验对我的 DBA 工作帮助良多。DBA 是在与系统和应用打交道，而不是仅仅与数据库打交道，因此应用软件开发、应用软件体系架构方面的经验和知识是必不可少的。在完全成为一名 DBA 之前，我曾经是一名软件工程师、项目经理、系统架构师。我设计过大量的应用软件，因此在分析一个系统时，我往往能够从开发者的角度去考虑问题，在处理问题时比较能够抓住关键，提出的建议也能够切合实际。我经常看到一些 DBA 给客户提出的优化建议，从 Oracle 数据库的理论上来看这些建议没有问题，不过作为一个系统来说，这些建议的针对性不强，可操作性很低，这种建议哪怕提得再多，再深刻，包含的技术含量再高，也是没有多大价值的。

在刚刚进入 DBA 这个行业，特别是刚刚工作时，应该多接触一些应用开发、系统体系架构、IT 架构和硬件方面的知识。这些知识的学习不能停留在表面上，而应该较为深入地去了解。做过系统工程师或软件工程师的 DBA 往往更容易成功。最理想的状态是在做 DBA 之前做过一两年开发，还从事过个把年的硬件工程师工作。实际上，在 DBA 的工作中，不断地要面对应用软件和系统硬件方面的问题。在实际工作中，也会不断地学习这些方面的知识。如果你并没有像我说的的那样，在从事 DBA 这个职业之前从事过软件开发或者硬件维护的工作，那也没有什么关系。在 DBA 工作中，尽可能多地去学习这方面的知识就行了。

在 1992 年到 1998 年，我逐渐从安装 Oracle 转向在 Oracle 上做各种应用软件，其中也经常对数据库进行简单的性能分析和优化。在那段时间里，Oracle 相关的图书也逐渐多了起来。通过阅读相关图书，我对 Oracle 的一些基础知识有了一个整体的认识。

在工作几年后，我虽然安装过上百套 Oracle 数据库了，不过对 Oracle 的认识还是停留在很低的层次，于是我想系统地学习一下 Oracle 数据库。在这段时间里，有一本书对我帮助很大，那就是机械工业出版社出版的《Oracle 初学者指南》，这本书不厚，不过对于初学者来说，已经很全面了。后来我也向很多初学者推荐过这本书，但现在在书店里很难找到了。另外，赵松涛编写的《Oracle 9i 中文版基础培训教程》对于初学者入门来说也是十分好的。实际上，Oracle 的官方文档里有一些入门性的文档，对于初学者来说也是很有用的。《2 Day DBA》就是一本十分不错的入门图书，很适合初学者阅读。

在工作中，我也经常碰到关于性能方面的问题。早期的时候，对于性能的分析仅仅在各种缓冲区的命中率方面，我用 Delphi 编写了一个小程序，把一些常用的脚本编写成一个性能分析小工具，虽然那个小工具在现在的绝大多数 DBA 眼里只能算是一个“小玩意儿”，不过在 20 世纪 90 年代末，到客户那里拿出一个自己写的分析工具，客户还是能够高看你一眼的。随着为更多的客户优化系统，在性能分析方面，我学会了使用 BSTAT/ESTAT 工具，这个工具就是现在著名的 STATSPACK 工具的前身。在 Oracle 7 上，可以使用这个工具来进行 OWI 的分析。不过那段时间里，对于 Oracle 的学习还不是很系统，主要是在工作中遇到什么问题，就去学习什么知识。1999 年，一个偶然的机会有我读了一下 *Oracle Concepts*，感觉这本书对我的帮助很大，很多以前工作中碰到的疑点都在这本书里找到了答案。所以我会给每位 Oracle 入门者推荐这本书。认真读几遍这本书，比学习一些独门秘籍要有用得多。*Oracle Concepts* 这本书适合不同层次的 DBA 阅

读，对于初学者，阅读这本书可以打好基础，不过第一次阅读这本书的人，可能很多地方无法理解，其实这时可以跳过那些看不懂的章节。有了一两年工作经验后，再来阅读这本书时，你会感觉这本书能给你带来很多惊喜，很多工作中碰到的匪夷所思的事情，这里都会给你一些合理的解释。这时阅读这本书，还可以让你对你前一段时间的工作和学习做一个梳理，得到一个很大的提升。实际上工作几年后再来阅读这本书的一些章节，还是感觉会有收获。几年前，我粗粗地浏览 *Oracle 10g Concepts* 时，仍然感觉受益匪浅。

在 1999 年，由于我要给几个客户做一些维护工作，所以对 Oracle 的知识做了一些梳理，期间阅读了一些 Oracle 图书，但最大的发现是 Metalink。由于给客户做相关服务，从客户那里拿到了 Metalink 账号，从那时开始，我发现以前想从图书上获取的知识绝大多数都能够在 Metalink 上获取。通过对 Oracle 概念的阅读，我已经基本上掌握了 Oracle 的基本概念和体系，知道了 SGA、PGA 和 UGA 等基本概念，但是这些概念在我的脑子里还是凌乱的、不成体系的、粗浅的。这些概念对于我做一些复杂的分析帮助不大，我需要更加深入地理解这些概念。在 1999 年到 2000 年，由于客户的水平和维护需求也相对较低，所以虽然我在协助客户进行数据库维护，实际上大多数工作都是较为初级的工作。这段时间里我花了大量的精力在 Metalink 上阅读技术文档，这时的主要学习任务是扩大知识面。Oracle 是十分庞大的体系，其广度很大，如果要掌握 Oracle 的一些基本技术，就必须花费足够的时间。在这之前，我的 Oracle 知识主要集中在和应用软件相关的方面。通过这段时间知识面的扩展，我对一些 Oracle 的主流技术、工具基本上都有了一个初步的认识。这段时间的学习对于从事系统架构设计的我来说，帮助也是十分大的，因为这段时间里我面对的主要系统还是使用 Oracle。由于对 Oracle 数据库了解得深入，在我进行系统设计时，就不自觉地从 Oracle 数据库的角度去考虑应用软件，使应用软件能够更加适合 Oracle，更多地利用 Oracle 的新技术。

突破瓶颈

从 2000 年开始，我突然发现我遇到了瓶颈，无论如何看书学习，自己的技术水平就像停滞一样。随便拿一本书翻翻，我都会觉得书上的知识我都看过，好像没有什么东西能够让我提升能力了。这一点让我十分困惑，因为我知道，对于 Oracle 来说，我刚刚入门。有一天，我和海外的一位朋友在 MSN 上聊天，说起了这个问题，他建议我写本书，把我所学到的 Oracle 知识写出来，这可能对初学者有很大帮助。事后我考虑了几天，于是决定写一本关于 Oracle 数据库的书，书名叫《Oracle 深度历险》，这本书包括第 1 章 Oracle 基础知识，第 2 章 SQL 与 Oracle 数据库编程，第 3 章深入了解 Oracle 数据库，第 4 章 OEM 与其他 Oracle 第三方工具，第 5 章备份恢复与容灾，第 6 章 Oracle 数据库性能优化。为了编写这本书，从 2000 年到 2003 年，我阅读了大量 Oracle 数据库方面的技术资料 and 图书。由于第 3 章的内容是介绍 Oracle 基本原理的，所以我查找了数百篇关于 Oracle 内部原理的技术资料，其中大多数来自 Metalink。在收集资料的过程中，我也得到了美国和澳大利亚 Oracle 公司朋友的大力协助，获得了大量仅限 Oracle 公司内部使用的文档。这些文档对于我理解 Oracle 的内部原理帮助十分大，并且我已经陆续将其发布在 Oracle 粉丝网

(<http://www.Oraclefans.cn>) 上了。《Oracle 深度历险》的编写工作历时 3 年，不过在 2004 年我第二次修改这本书时，还是决定放弃这本书，因为那时候 Oracle 的技术书已经相当丰富了，再出版同样一本书没有任何意义。虽然《Oracle 深度历险》夭折了，不过写书的目标让我在两三年时间里对 Oracle 重新梳理了一遍，对 Oracle 的认识也更加深入了。在写书的过程中，对于每个知识点，如果能够进行实际操作的，我必须亲自操作一遍。这段时间虽然没有写出一本好书，不过写书的经历对我来说是一笔十分宝贵的财富。这几年来，我突然发现 2000 年时我感到的瓶颈在不知不觉中被我突破了。在这之前，我学习了大量零散的知识，虽然利用这些知识可以解决日常碰到的大多数问题，但是我并没有考虑把这些知识串起来，并没有深入地去了解 Oracle 的内部机理，因此我思考问题时往往只局限于某一个知识点。通过这 3 年的写书经历，我习惯于从 Oracle 的内部机理来思考问题，而不仅仅是从某个问题的表象入手。这样我在理解一些 Oracle 原理时，可以更加深入、更加周到。写书带给我的思考使我突破了瓶颈，进入了一个新的境界。

我也经常建议公司的员工，不要光看书，看书时一定要自己亲自做一遍，然后再把做的过程写成文档，书上的东西才能真正变成自己的。其实这个过程包含了几个步骤，第一步是通过读书学到了新的知识，然后通过自己的亲自实践将书中学到的新知识得到一个感性的体验，最后将这个体验用自己的语言描述出来，那么这个知识点就记住了。如果不这么做一下，读书的效果就要打很大的折扣了。

对于 DBA 来说，写博客是一个不错的主意。将自己学习的成果通过博客整理出来，既可以起到整理思路，完善知识点掌握的作用；又可以通过博客和其他 DBA 进行交流，为其他正在学习中的人提供技术资料，最终达到群体学习的目的。我经常建议公司的年轻人群体学习。群体学习说起来很简单，就是如果存在一些知识点，有几个人都想去学习，如果每个人都是独立地去学习可能需要花费一两个月的时间，如果换一种学习方法，就是几个人分分工，每个人学习一个知识点，然后通过互相交流的方式，大家互相传递知识。这种学习方法可能可以缩短一倍的学习时间，而且学习到的知识的深度也会高于一个人自己学习。群体学习的前提条件是，一起学习的人的水平基本接近，而且大家都很开放，都愿意把自己的知识拿出来和大家分享。

大成之道

2000 年的写作使我突破了第一个瓶颈，不过很快我又遇到了另外一个瓶颈，这也是很多 DBA 都会遇到的。在 2003 年到 2004 年，我经常有碰到了天花板的感觉。这段时间里我在经营一家软件公司，最初的时候还自己写一些代码，随着公司越来越大，最后连系统架构都交给了技术总监。从那时候开始，我觉得自己离开发越来越远了，不过在这个阶段我也有更多的时间研究 Oracle 相关的技术。这段时间给客户做优化的项目比较多，在做项目时，总是感觉很难较快地抓住问题的核心。在这段时间里，我阅读了大量 Oracle 内部和优化的相关图书，包括 *Cost Based Oracle Fundamentals*、*Oracle 8i Internal Services for Waits Latches Locks and Memory*、*Inner Look on Oracle Latches*、*Oracle Performance Tuning & Tips*、*Oracle 9i Tuning Guide* (Oracle 官方文档)、*DSI 401E*、*DSI 402E* 等。实际上，每一本书对我理解 Oracle 都有很大帮助，但是哪一本书也没办法解决我

所有的问题，在很多地方涉及内部原理和算法时也往往都是点到为止。DSI 是 Oracle 内部培训教材，也是广大 DBA 追逐的目标。认真学习一下 DSI 对理解 Oracle 内部原理是有帮助的。不过想理解 Oracle 内部原理并不是只有 DSI 一条路，说实在的，DSI 的文档，我手头有一些，不过真正认真去看过的也只有 *DSI 401E* 和 *DSI 402E* 这两个。现在 DSI 文档在互联网上很容易下载到。系统地学习一下 DSI 课程对于中级水平的 DBA 来说是个不错的选择。

除了读书外，还有很多问题是书本上无法回答的，所以我也在互联网上查找更多关于 Oracle 内部原理的文档，通过 Google 和百度去搜索更多的 Oracle 技术网站。<http://ixora.com.au> 是一个相当不错的网站，上面有很多关于 Oracle 内部原理的资料，虽然资料基本上都是基于 Oracle 8 的，但是资料十分权威。ASKTOM 网站也是这段时间我经常访问的网站。最初知道 ASKTOM 是通过 Google 搜索的时候，基本上都有链接指向 ASKTOM，从 ASKTOM 我学到了 Tom 分析问题的思路和方法，这对于我今后自己处理问题是很有帮助的。

至于国内的网站，那时候 ITPUB 和 oracle.com.cn 比较热门，开始的时候也经常在这两个网站上讨论一些技术问题。随着这两个网站上一批 DBA 的水平越来越高，这两个网站上技术交流的质量却越来越低。我也很难在这两个网站上找到所需要的资料，于是把目标转向了英文网站。国外的 Oracle 技术网站很多，不过大多数都是会员制的收费网站，少量免费网站（比如 ITToolbox）上面的技术水平又普遍比较低，所以找了一圈，最后还是觉得学习 Oracle 最好的网站还是 Metalink。由于朋友的帮助，这段时间获得了不少 Oracle 内部的技术资料。通过学习这些资料，我很深入地了解了 Oracle 内部的一些算法。这些学习过程对我突破这个瓶颈起到了至关重要的作用。

在突破这个瓶颈的阶段，我的学习方法是对于某一个知识点，深入地研究下去，并尽可能地把相关知识点融会贯通，特别是两个知识点的结合部。为了了解数据块的结构，我花了相当长的时间去转储数据块，甚至编写了一些小程序去读取数据块中的数据，这时以前搞开发时沉淀下来的深厚的 C 语言功底起到了很好的作用。对于绝大多数 DBA 来说，完成好日常工作并不需要学习那么多内部原理性的东西，不过如果你有兴趣并且有足够的时间，深入研究 Oracle 的内部结构和原理是十分有趣的事情。至少对我来说是这样的，从 bug 报告生涩的文字和少量的内部代码里分析 Oracle 内部实现的原理，和看一部好的小说一样有趣。上大学时和刚毕业时喜欢看一些散文和诗歌，而现在我阅读的对象除了轻松的商业小说外，就只剩下 Metalink 文档了。这是一个爱好问题，可以说已经与 Oracle 技术无关了。

对于绝大多数 DBA 来说，可能他们缺少实践的机会，这对于 DBA 这个职业来说是十分致命的，可能会影响到 DBA 的成长。我第一次做优化项目时，虽然说那时候我已经具备了很深的理论知识，从事 Oracle 维护工作也有几年的时间了，但在项目一开始，还是碰到了很多意想不到的困难，甚至有时候出现了方向上的偏差。这实际上和实际工作经验是息息相关的，哪怕你的理论水平再高，没有真正做过几个优化项目，是很难真正理解什么叫优化的。对于缺乏实际工作经验的 DBA，在 ITPUB、Metalink、OTN 等论坛或者 QQ 群里帮助别人解决问题是一个十分好的办法。你可能会碰到各种各样的案例，它们都是你在目前的工作环境中无法碰到的。通过接触这些实际案例，可以弥补你在实际工作经验上的不足。从 2004 年起，我建立了自己的 Oracle 讨论群

“白鳝的洞穴”，在这个群里，帮助网友解决问题是一件十分有趣的事情。

当你通过了 OCP 甚至 OCM 认证考试后，你的理论知识积累已经达到了一定的程度。这个时候，如果你不能从事相应的工作，在这些工作中把你的理论知识消化并融入你的思维当中，这些理论知识很快就会在你的头脑中消失。一两年后，这些理论知识就会被忘得干干净净。所以说，通过认证考试不是目的，只是一个过程。当然拥有 OCM 证书，会给你的职场生涯带来很多好处。不过认证考试只是一个方面，深入实践是更为重要的。我见过不少拥有 OCM 证书的人，他们有些已经在企业里做管理了，一旦离开 DBA 生产的第一线半年或者一年以后，这些人也和普通的 DBA 没有多少区别了。因此，作为一名 DBA，如果你想一直保持你的技术水平，不要离开生产一线是十分关键的。这当然又是一个悖论了。对于绝大多数人来说，DBA 只是一个职业，一个谋生手段而已，技术只是获得提升的手段，而生活才是永恒的话题。对于我来说，Oracle 是一种爱好，是一种生活方式，所以我最终选择放弃了在大企业做高级技术管理的职位，去从事一个和 Oracle 密切相关的职业。而对于绝大多数人来说，完全没有必要这么执着和痴迷，Oracle 只是生活的一部分，而不是全部。

理论知识的学习是十分枯燥的，很少有人愿意一遍一遍地阅读 *Oracle Concepts*，虽然这么做对你的 Oracle 理论知识的提升帮助很大。和我同龄的人也许还会记得一款很经典的游戏“金庸群侠传”，这个游戏里最牛的武功是什么？不是葵花宝典，也不是独孤剑法，而是主人公刚出生就会的一种拳术“野球拳”，这种拳法的 1~9 级都平淡无奇，不过只要练到 10 级就突然变得无坚不摧了。实际上研究 *Oracle Concepts* 也是一样，如果我们能够通过反复阅读，把 Oracle 的概念融会贯通，一样能够起到“野球拳”的效果。

对于初学者来说，还有一点要注意。那就是 Oracle 的知识面很广，一个人也很难有机会把所有的 Oracle 知识都认真地梳理学习一遍，因此我们在学习时要注意抓住重点，不能胡子眉毛一起抓。在 DBA 学习的初期，尽可能拓宽知识面，了解 Oracle 的方方面面，而一个名资深 DBA 会在某些方面可能有十分深入的研究，但是在大多数知识点上可能只能点到为止，甚至只能简单地了解，直到需要的时候才可能去深入分析。

在这个阶段给别人讲课是学习理论的一个很好的途径，我有一些知识就是通过给别人培训学习的，比如说 Data Guard，对于 Data Guard 的原理，我大体了解，这是从重做日志的结构方面可以推断出来的，但是对具体的一些技术细节以及实现方法就知之甚少。有一次需要给一个客户讲 Data Guard 的课程，我花了差不多两天的时间来整理培训讲义，并且准备了一个学生实际操作的实验包，设计了几个常见的演练场景。通过这个备课过程，对 Data Guard 的原理、配置管理和维护的基本操作就了解得十分清楚了。几年过去了，虽然 Data Guard 的技术在不断演进，我在这几年里也没有做过 Data Guard 的项目，但是通过那次讲课，Data Guard 已经深深融入了我的血液里，再也不会忘记了。随着数据库新版本的推出，Data Guard 技术在不断演进，不过其主体技术是不会发生大的变化的，因此在有了一定基础后，只需要你在使用前再去复习一遍相关知识就足够了。

有些知识点可能平时不会注意，但是如果你要给别人讲课，就需要你认认真真地把这些知识

点一个一个搞清楚，因为没有一位老师愿意在课堂上被学生们问得哑口无言。记得刚刚工作时，公司派我给一个客户讲 RDB 数据库，在这之前我连 RDB 数据库是什么都不知道。那时候正在 DEC 公司，老板是香港人，给我留下一本讲义和一本 RDB 的参考手册，给了我 3 天时间准备。那是一项十分艰苦的任务，我给客户讲了近 10 天的 RDB 课程，每天晚上我通过阅读讲义和参考手册备课，对于讲义上的每个实验都要自己先做一遍，第二天再教给学员们。10 天的课程终于讲完了，没有一名学员看出我也是第一次学习 RDB。通过那次讲课，我对 RDB 数据库的了解比公司里其他员工都要深。10 多年后的一天，在一个客户那里听说他们的一个 RDB 数据库里有一批很重要的数据想搞出来，但没有人懂，所以只能通过应用程序显示在终端上，然后由一人一点一点地抄下来。我听说后，凭着那次讲课对 RDB 留下的印象，很快就帮用户把数据导成文本格式，然后通过 SQL Loader 装载到了 Oracle 数据库里。

最后要重申的是，DBA 是一个工作，一个还算不错的工作，但是并不是每个人都需要把 DBA 当做一种生活。因此，对于技术的追求并不是每个人的生活目标。如果你喜欢 Oracle，那你不妨把阅读枯燥的理论和干涩的 trace 文件当成一种乐趣，否则就把它当成一种工作，一种生活中的点缀吧。

DBA 的性格

不一定任何人都需要从事 DBA 这个工作，也不是每个人都能成为优秀的 DBA。DBA 是一种压力相对比较大的职业，要求从业人员在工作期间不断地学习新技术。Oracle 数据库每 5 年左右会进行大版本升级，这就需要 DBA 不断地学习新知识。记得几年前在做一个项目时，和一位干了七八年的老 DBA 一起聊天，他说本来想好了，Oracle 9i 的技术就不去学习了，就吃 Oracle 8i 的老本了，不过没办法，想要生存，必须去学习。最后他说他的最大愿望是不要再去学 Oracle 10g 的东西了。不过愿望只是愿望，两年后，我看到他出差时带着一本 Oracle 10g 的书，就说起了那次对话。他也只能笑着说，干 DBA 的都是苦命人，不学习是不可能的。DBA 这个职业可以做得很长，国外一些高手和大师都是从事 DBA 工作超过 20 年的。不过对于绝大多数朋友来说，DBA 只是职业生涯中的一个台阶而已，因此在做职业规划时，首先你需要考虑 DBA 是作为一种过渡性的工作呢，还是作为一种生活和爱好。

要回答这个问题，就需要根据自身的性格来考虑了。有几种性格是不适合做 DBA 的。DBA 需要谨慎的态度，如果你的性格比较急躁，那么 DBA 不是适合你的工作。DBA 承担了企业中最为重要的数据库的维护，其工作性质决定了 DBA 是一种压力十分大的职业。在处理日常工作以及突发性问题时，急躁是最为可怕的性格，越是碰到紧急的问题，越需要 DBA 以冷静的心态来面对，否则很容易出现不必要的问题。2004 年美国的一项调查表明，超过 30% 的系统故障是由于维护人员人为失误造成的，因此沉稳的性格是 DBA 减少出现操作失误的一个重要保证。

除了过于急躁外，好奇心太强的人也不适合做 DBA。DBA 在做维护工作时，经常会碰到一些莫名其妙的事情，和自己工作无关的事情尽量不要做，这是铁的纪律。Oracle 公司的工程师到客户现场工作时，一般会拒绝客户提出的和本次任务无关的工作，这也是 Oracle 原厂服务经常被

客户诟病的一点。不过我认为这是一种很职业的态度，我只做和我工作相关的事情。从另外一个角度来说，作为一名负责任的 DBA，从职业的态度来考虑问题，就是做自己技术能力范围内的事情。有些 DBA 无法判断某个操作的风险，在这种情况下，客户让你做某件事情，你到底是做还是不做呢？最好的方式是通过向专家咨询，确认没有问题后再去做。而往往在实际工作中，大多数 DBA 都会尝试干一些自己没有把握的事情。一名好奇心很强的 DBA，可能发现了一个新的脚本，就很急迫地想在自己维护的生产库上尝试一下，可能他根本没有去考虑这个脚本是否存在风险。实际上，在我这 10 多年的 DBA 工作中，也多次出现了由于好奇心强导致做一些自己认为没有风险的事情，结果或多或少地造成了一些问题，甚至有一次我在一个客户的生产库上尝试一个以前没有做过的转储命令，最终碰到了一个 Oracle 的 bug，导致 RAC 的一个节点宕机。从那以后，哪怕再好奇，我也会先充分评估操作的风险，并且尽可能不去做一些和自己工作无关的分析。实际上，作为一名 DBA，是很难经得起诱惑的，因为有很多情况可能你一辈子也碰不上几回。作为一个爱好 Oracle 的人，碰到了某种现场，都可能会被吸引，甚至诱惑。作为 DBA，经得起诱惑，是十分好的性格。从另一方面来说，DBA 需要足够的职业素养，由于 DBA 工作的风险十分高，任何一个违背职业素养的工作习惯都可能演变为工作中的失误，因此做一个真正的职业人是十分关键的。

除了上面讲到的沉稳，DBA 也需要有决断的性格。这似乎有些矛盾，我们虽然强调 DBA 不能胆子太大，但是在某些情况下，DBA 必须决断。有一次客户的数据库出现了严重的问题，导致宕机，启动后没多久再次宕机，客户也十分着急，由于时间十分紧迫，现场工程师和我们在二线做支持的人都没有足够的时间去进行分析，我当时感觉和我以前碰到的一个 bug 十分类似，不过从 Call Stack 来看，还是有些差别。当时现场工程师不敢做这个决定，我说这种时候了，如果这个补丁不起作用，那我们的服务也就做到头了。这种情况下没有别的思路，但是什么都不做肯定是不行的，所以立即打补丁。幸运的是，补丁打上之后，数据库恢复正常了。果断不仅仅是一种性格，这种情况下果断是基于一定的条件的，因为我知道哪怕这个补丁不能解决问题，也是没有副作用的。对风险的理解是果断的基础。所以说这里所说的果断也不是绝对的，不是刚愎自用，不是一意孤行，而是在自己的知识基础上的一种果断态度。因为做出选择不仅仅是依靠勇气，而更多的是依靠自己的底气。

DBA 的责任心是很关键的。一般来说，我面试一名 DBA，首先看到的不是他的技术能力有多强，而是他的工作态度和责任心。一个有责任心的人，哪怕技术水平稍微差一点，也不容易出大问题。而一名缺乏责任心的 DBA，不亚于一颗定时炸弹。能把工作当成自己事情的人是肯定能够成为一名好的 DBA 的。在很多情况下，DBA 的工作都是从纷繁的表象中去发现危险的存在，一个把工作当成苦差事的人是很难做到这一点的。我平时很少和同事发脾气，唯一的一次是因为一件小事，当时客户的一个系统需要我们帮助做一个健康性检查，一共有 10 多套大型数据库，要在两三天内完成巡检工作。当时有 3 个人一起参与巡检，采用的方式是集中采集数据、集中编写报告的方式，这种方式一般来说我们很少采用，因为这种方式可能导致巡检质量下降，但由于时间紧迫，也只能采用这种权宜之计了。在做巡检之前，我就和哥几个说虽然时间紧，但是一定

要认真。虽然哥几个答应得挺好，不过报告出来后，我感觉还是过于粗糙。我只好打回去让他们整改，整改了两三次还是难以让人满意。事后我和哥几个说，如果你把这件事当成一个工作，确实让一个人在这么短的时间里做这么多库的巡检，难免会有些枯燥，质量下降也是难免的，但是如果你是以前的手工艺者，做巡检就是我们的手艺，你拿出的活能不能对得起自己这点手艺呢？大家听后都感触颇深，既然我们吃这碗饭，我们就应该拿出对得起这碗饭的手艺。现代社会比较浮躁，大家都是为了生活而工作，工作已经不是目的而只是手段，这一点我也能够认同。不过人除了物质的东西，总还是需要一些形而上的信仰来支撑自己，否则会失去很多乐趣的。这种信仰就是手艺人赖以生存的基础，失去了这些信仰，把 DBA 工作当成纯粹的谋生手段，那么你还为了解决一个问题而兴奋不已吗？还会为了自己的失误而感到懊悔吗？

每一个准备做 DBA 这个工作的人，无论自己的职场规划如何，作为 DBA 就应该明白自己承担什么样的责任。摆在我们面前会有很多诱惑，你面对的是企业最为宝贵的财富——数据库。可能你干一辈子的收入还不如把其中一小部分数据复制出去卖给别人赚得多，但是你必须守住自己的信念，你必须对得起自己，对得起自己的衣食父母。记得刚刚工作时，我在 DEC 软件中心帮助香港氧气公司移植他们的核心业务系统，我负责的工作就是将香港氧气公司的 TME 数据库里的数据移植到 OpenVMS 的 RMS 数据库中去。我第一次接触数据之前，老板让我签署了一个保密协议，他当时对我说，“这些数据，随便拿出一些，你就可以卖出几十万的价钱，但是我相信你不会这么做。作为职场中的人，这是最起码的道德底线，今后你可能会遇到很多类似的事情，只要你一次触动了底线，那就万劫不复了。”作为 DBA，那根底线是绝对不能突破的，这不仅仅是道德问题，实际上这个底线是对我们最好的保护。

一个人的性格是天生的，不过也是可以改变的，如果一个人想去做一件事情，并且不断地在努力，成功的机会是很大的。连郭靖这种蠢笨如牛的人都可以成为一代宗师，你想成为一名 DBA 又有何难呢。虽然说不是所有的人都适合做 DBA，不过这一切对于一个努力的人来说，都不成问题。性格是可以改变的，习惯是可以改变的，为了自己的目标，可以改变一切的人，还有什么不能实现吗？我们公司有一个小伙子，性格极为内向，和同事在一起上班，可以一天只说一两句话，甚至一句话也不说。有一次去客户现场工作了两个多月，我们给他一个额外的任务，那就是请客户的 DBA 吃一顿饭。就是这么一个很小的任务，他最后都没有完成。按理说，这种性格的人是很难成为一名合格的 DBA 的，因为 DBA 需要和别人沟通。作为 DBA，三分靠技术，七分靠沟通。就是这样一个内向的人，在大家的努力下，通过一年的时间，居然有了很大的改变，首先是和自己同事之间的沟通多了起来，和客户之间的交流也逐渐好了起来。虽然和其他工程师比较，他还是属于沉默寡言的那一类人，但可以看出，他一直很努力地克服自己的瓶颈，而且我们也看到了他的努力所得到的成果。我想再有一两年的时间，他会成功的。这里我举这个例子就是想说明 DBA 的最后一个也是最重要的性格——坚持。大家应该都看过《士兵突击》，许三多不是一个当兵的料，不过他在战友的帮助下，一直坚持着，最后成就了兵王。在这个故事里，有两个重要的要素，一个是许三多的坚持，一个是战友的坚持。钢七连的“不抛弃，不放弃”的信念是成功的关键。对于一个刚刚走入职场想成为一名成功 DBA 的人，这个信念尤为重要。