



# 收获，不止Oracle

梁敬彬 梁敬弘 著

打开惊喜之窗 翱翔意识天空

兄弟出品



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# 收获，不止 Oracle

梁敬彬 梁敬弘 著



电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

在这本书里读者将会跟随作者一同对 Oracle 数据库的相关知识进行梳理, 最终共同提炼出必须最先掌握的那部分知识, 无论你是数据库开发、管理、优化、设计人员, 还是从事 Java、C 的开发人员。接下来作者再将这部分知识中最实用的内容进一步提炼, 浓缩出最精华的部分, 分享给大家。这是二八现象的一次经典应用。

这部分知识就是 Oracle 的物理体系结构、逻辑体系结构、表、索引以及表连接五大部分。通过阅读这些章节, 读者将会在最短时间内以一种有史以来最轻松的方式, 完成对 Oracle 数据库的整体认识, 不仅能在工作中解决常规问题, 还能具备一定的设计和调优能力。相信通过这些章节的学习, 会给读者的 Oracle 学习带来极大的收获。

然而, 作者最希望看到的是: 让读者的收获, 不止 Oracle。

为达到此目的, 作者精心将全书分成了上下两篇, 刚才所描述的具体知识点体现在全书的上篇, 而在下篇中, 读者将通过各种精彩故事、生动案例, 体会到该如何学习和如何思考, 在意识的天空抛开束缚, 无拘无束、尽情飞翔。

在这里, 读者也许会有疑问, 前面说的有史以来最轻松的方式是一种什么样的方式呢?

还请亲爱的读者自己去揭晓谜底吧。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

### 图书在版编目 (CIP) 数据

收获, 不止 Oracle / 梁敬彬, 梁敬弘著. —北京: 电子工业出版社, 2013.5  
ISBN 978-7-121-20070-0

I. ①收… II. ①梁… ②梁… 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 063004 号

策划编辑: 张月萍

责任编辑: 徐津平

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

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: 787×980 1/16

印张: 31.25

字数: 750 千字

印 次: 2013 年 5 月第 1 次印刷

印 数: 4000 册 定价: 59.00 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 [zltts@phei.com.cn](mailto:zltts@phei.com.cn), 盗版侵权举报请发邮件到 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 序

随着 IT 技术的蓬勃发展，相关技术书籍也大量涌现，这是令许多从事 IT 行业的人士十分高兴的事。但是，大多书籍都是就技术领域来谈技术，读起来难免有些枯燥或感到深奥。而我读这本书的内容时，禁不住有一种冲动，一个声音从心里喊出：原来技术书籍也可以这样让读者喜爱！

从来没有为哪一本书写过序，也从没有想过可以为一本书写序，因为心中明白自己没有这样的能力。但是，当我们公司年轻的数据库专家梁敬彬将他的书稿发给我参阅时，我被这本书深深吸引住了，更被他的一句肺腑之言感动了：“我写完这本书，有一种如释重负的感觉，因为我把自己认为对广大技术人员最有价值的东西分享给大家了，我知道很多人可以因此不再为技术所累！”我为我们公司有这样一位年轻的、优秀的、行业内知名的数据库专家感到十分自豪。他是用自己的行动践行着我们福富公司“敬天爱人、追求卓越”的核心价值观；他把自己对广大技术人员的爱通过这本书传递出去了。游于技艺，然心与道合，心中甚感愉悦。于是，认真学习为其作序。

《收获，不止 Oracle》这本书从思维方式和做事方式的高度来阐述技术，相信这样的书对 IT 从业人员是十分有益的。在这个知识爆炸的年代，技术不仅学不完，而且还在不断更新。可以说，具体某阶段技术知识的学习都是暂时的，而学思想、学做事的方法论才是永恒的。授人以鱼不如授人以渔，学会学习，学会总结，学会思考，才是最重要的。这本书将思想、方法、案例、故事完美地融合到技术中，这不仅需要作者具备精湛的知识、丰富的实战经验，更需有思想的高度提炼、方法论的总结、实际案例的组织 and 故事情节的构思。可以说，这本书是浩瀚技术海洋中的一颗璀璨明珠。翻阅过程中，更是惊喜不断，本书除了站在思想的高度将技术和方法论巧妙结合，弥补了业内技术书籍的遗憾外，其妙趣横生、新颖别致的风格也超越了我的想象。读者阅读全书仿佛置身课堂，别具一格的课堂风格让人耳目一新，饶有趣味的故事构思让人拍案叫绝！对于从事和即将从事 IT 相关工作的工程师们来说，遇到这本书应该是一种幸运、一个善缘！

作为公司层面的技术专家，梁敬彬服务的客户遍布全国各地，他能在十分忙碌的工作之余，把自己宝贵的工作经验撰写成 50 余万字的生动有趣的经典技术著作分享给广大技术人员，这是他的努力不亚于任何人的证明！我很高兴、也很欣慰地看到我们的员工有这样的胸怀和激情！

此外，此书的校验及补充完善是由拥有多项核心专利的知名专家、清华大学计算机系毕业的梁敬弘博士完成的。感慨两位专家联袂奉献之余，更感受到他们兄弟间默契配合所传递的爱的情谊。

岂曰无衣，与子同袍。此书必为 IT 行业带来一股清新之风，打造出 IT 书籍的一段传奇佳话，成就一部永不过时的经典力作！我相信，此书将为广大读者开启惊喜之心扉，让读者飞翔在思想的天空、遨游在方法论的海洋、采摘到技术的累累硕果，相信收获远不止 Oracle！

**福富软件公司副董事长、总经理：杨林**

# 别出心裁 另出蹊径

## ——与梁敬彬先生序

敬彬的新书就要出版，邀我写一点感受，于是就有了这一段文字。

我和敬彬相识是在 2010 年，那时我正在编辑《Oracle DBA 手记》一书，偶然被他发表在 ITPUB 论坛上的一篇文章所吸引，那篇文章的题目是《DBA 小故事之 SQL 诊断》，其内容清晰、行文引人，于是就和他约了那篇稿子加入到书中，期间邮件往来再到北京会面，就此熟识。

从当时的一篇文章到今天的一本书，我能够清晰地看到作者一以贯之的思考和叙述方式，这种积累与坚持也正是作者成长和成功的要素之一。

当时那篇文章的感受和今天这本书是类似的，作者能够用曲折的笔触将自己的经历真实生动地再现出来，并且带领读者一起经历一次思维的探索，这是属于他的独特风格。

作者在书中反复传达的核心观点是：Oracle 数据库看似艰深的原理实际上和生活中的基本常识并无二致。理解了这一层意思，就能够克服对于这门技艺的畏惧之心，此后的学习自然就能够顺风顺水。

诚然如此，我也经常和朋友们说，对于 Oracle 的很多艰深算法，如果由我们去深思熟虑，其结果都必然大致相同。类似 HASH 原理，布隆过滤等算法，理解了你就只觉得巧妙而不觉艰深。

现在梁老师就为我们寻找了一系列源于生活、循序渐进的学习路线，如果你能够细心领会，就会觉得这一门技艺实在是趣味横生。

盖国强 (eygle)  
Oracle ACE 总监  
云和恩墨创始人  
ACOUG 创始人

# 名家推荐

敬彬兄这本书有着与市场上其他 Oracle 书籍与众不同的特点，他通过一个个精彩的小故事，串起 Oracle 的核心知识和优化方法论，并时刻强调学习和工作的意识，如何不被技术束缚，如何跳出技术，意识和方法真的很重要。相信读完本书，你的收获，绝对不止 Oracle！

丁俊 (dingjun123)

ITPUB Oracle 开发版资深版主

《剑破冰山——Oracle 开发艺术》副主编

通读本书，如醍醐灌顶，豁然开朗，本书从实战出发，出发于技术，而超脱于意识，回味无穷，作者拥有多年的 Oracle 应用和体系架构设计的经验，付出了不亚于任何人的努力，总结出众多独到的经验，不失为一本好书，为学习和使用 Oracle 的技术人员带来诸多益处。

傅祥文

福富软件公司运营总监

由梁敬彬、梁敬弘兄弟合作的《收获，不止 Oracle》一书问世了。这对学习和从事数据库相关事务的业内人员来说，是一件幸事。读一本技术方面的书，或修一门课程、听一个讲座，大凡可能有三方面的收获：掌握相关的知识，提高解决问题的能力，激发学习、探讨有关问题的兴趣和热情。这些可能的收获不在一个层面上，后两者更为可贵。梁敬彬梁敬弘兄弟的这本书恰恰给读者提供了这样的机会。当然，要有真收获，还要有真努力。

梁敬弘曾是我的学生，不仅学业专精，围棋也下得很不错，是一个真诚而聪明的小伙子，跟他的哥哥相比，内向一些。梁敬彬与弟弟相比更善于沟通和表达，是一个数据库方面的专家，也

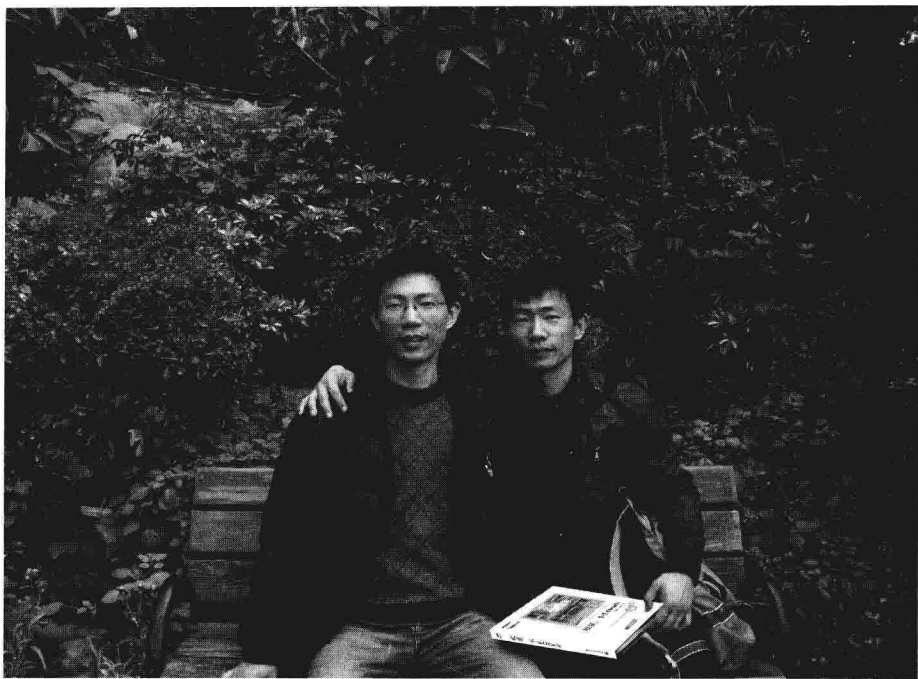
是一个很好的教师。在此，预祝本书的出版获得成功，同时也祝兄弟二人在事业上不断取得新的成就。

黄连生  
清华大学计算机系教授

曾经有 Oracle 的初学者问我，怎么开始学习 Oracle？那时候我的答案很简单：“去下载 Oracle 的在线文档，包括 Database Concept、Administrator’s Guide，然后开始学着做实验。”诚然，对学习技术而言，在线文档是一个不可多得的利器，但是，对于一个刚刚开始接触 Oracle 的人来说，要从枯燥的英文文档中去学习和理解 Oracle 的技术体系，也许有点勉为其难。就算是市面上众多的 Oracle 技术书籍，多数也是堆砌满了技术细节，随时可能吓跑初学者。

好在，现在梁敬彬先生通过自己在日常工作和培训中的磨练，把自己对 Oracle 技术的感悟，通过一个一个的小故事，浅显而又形象地展现了出来。对于初学者来说，可以慢慢地在一个个小故事中去了解 Oracle 数据库。读完这本书，你也许会恍然大悟：“哦，原来 Oracle 是这样子的。”

罗海雄 (rollingpig)  
ITPUB Oracle 管理版资深版主



梁敬彬，网名 wabjtam123，任 ITPUB 版主、ITPUB 社区专家、福建富士通公司数据库专家。参与编写过《剑破冰山——Oracle 开发艺术》、《DBA 手记 2》等技术书籍，多年从事电信相关行业工作，负责系统架构设计、优化、培训等工作，有着丰富的数据库管理、设计、开发、培训经验和电信行业经验。

梁敬弘，清华大学计算机系博士毕业，在计算机领域和金融领域皆有建树，拥有多项计算机相关核心专利技术的同时还拥有金融行业的 CFP 等高级认证。现就职于华夏银行总行。

## 前言

# 迈出崭新一步——本书特点及存在意义

### 0.1 当今时代，既是最好的也是最坏的

近年来，我深刻地体会到，如今这个时代对于绝大部分技术人员而言是幸福的时代，能在这个时代从事 IT 技术相关工作的技术人员应该很开心。因为比起老一辈技术人员，现今的技术人员几乎从来就不缺资料。除了可以在各技术平台的官方网站下载到详尽且准确的学习资料，他们还可以很容易地在 Google、百度上搜索到自己想要的答案，也可以很方便地在各种技术论坛上注册提问从而获取别人的帮助，其中最著名的就是 ITPUB 论坛，我本人也从中受益匪浅。

然而，这个时代对技术人员来说也是痛苦的时代。无论是传统的电信、金融、证券行业，还是新兴的互联网行业，我们都不难发现运维系统涉及的数据量、访问量及并发量都在以惊人的速度不断激增。重压之下，很多 IT 系统运行举步维艰，技术人员压力巨大，甚是痛苦。除了负载压力外，系统的复杂度也随着业务复杂度的增加而成指数级增加，让开发设计人员头昏脑胀，让故障诊断及系统维护的相关技术人员无从下手。

痛苦还不止于此，在这个时代，除了海量负载和高复杂度，还有让技术人员应接不暇而无所适从的各种新技术。且不讨论 IT 所有技术，仅以数据库为例，除了以 Oracle、DB2 等为代表的一系列传统关系型数据库外，还有内存数据库、列式数据库、以 HBase 为代表的分布式数据库等等，让人眼花缭乱。此外，除了各类纷繁技术的选型外，我们还要适应具体各个版本的不断更新，以 Oracle 数据库为例，从早期的 Oracle 5 版本到今天即将正式发布的 Oracle 12c，Oracle 每一次新版本发布都需要我们去学习和适应，投入大量的精力和时间。

这是幸福的时代，也是痛苦的时代。这是最好的时代，也是最坏的时代（如图 0-1 所示）。

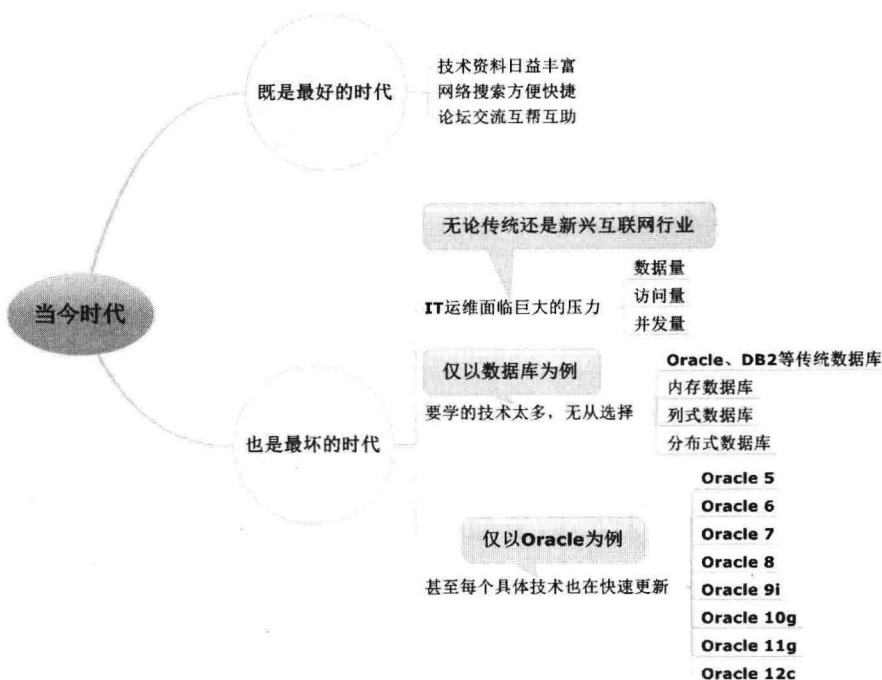


图 0-1 当今时代

## 0.2 技术人员，真正差距其实在意识

在这个特点鲜明的时代，我目睹了一大批在 IT 项目中不能有效适应这个时代而导致摔得鼻青脸肿，碰得一鼻子灰的技术人员，而这些人之中大多还是勤奋上进之人，少数还是技术能手。当然，也有成功的例子，让我来说说他们的故事吧。

### 0.2.1 小白的事情

工作两年多的小白是一个很努力的员工，每天除了用心工作外，还坚持充电学习，两年来他除了把 Oracle 10g 和 Oracle 11g 的官方文档几乎读了个遍，还购买了不少相关的书籍来学习。由于小白的项目组工作强度大，时常需要加班，因此小白工作学习外的时间就只够吃饭和睡觉了。那他的工作表现很出色吗，天道是否一定酬勤？实际情况却是不尽如人意，虽然他了解很多相关知识，可是在实际工作应用中却不得要领，频频出错，领导经常对其提出批评，他自己也相当苦恼。

其实小白只是因为缺少了一种称之为“意识”的东西，这个意识就是，该如何学习。在他的学习过程中，他犯了很多错误。

### 1. 忽略了知识的重点

小白读完了 Oracle 官方文档的几乎所有内容，毅力让人钦佩！遗憾的是，他忽略了二八现象这个事实：20%的知识，解决 80%的问题。小白根本不需要读完 Oracle 所有文档，他所在的项目组是从事数据库开发相关工作，而他却花费了大量的时间阅读完了至今他都尚未开始从事的数据库管理相关知识，比如备份恢复、RAC、DATAGUARD 部署及高级数据同步复制相关的技术，这是完全没有必要的。

另外语法方面的资料小白也根本无须细读、记忆，因为平时用得多的语法，自然记得住，遇到偶尔忘记的场合再去官方文档翻阅即可，现在 Google 等搜索引擎也非常强大，搜索到相关语法易如反掌。

人的精力是有限的，如果我们做到当前做什么事对应学习什么知识，尽量理解原理而不强记语法，将能省下多少宝贵的时间啊。

### 2. 从未考虑知识落地

小白曾经对我说过自己的烦恼，向我说明了他有多努力，看过了多少资料。我试探性地问他是否知道 Oracle 的体系架构是什么？他很自信地做了回答，滔滔不绝，甚至还画了草图给我，回答得让人相当满意。我又问他是否知道索引的结构和原理是什么？再次得到完美的答复。

接下来我再问他，知道这个体系架构对我们工作有啥帮助呢？他一下子回答不上来了。我继续问他，那索引的结构和原理对我们工作中的数据库应用有啥好处呢？再次答不上来。

原来这就是问题的关键所在！

知识要落地，要思考应用的场合，这就是我除了要让他把握学习重点外的第二个建议。他的学习其实就是为了学习而学习，没有思考应用场景的学习，是没有任何意义的。

后来我建议小白来听我在公司开讲的 Oracle 系列讲座，让他明白原来了解体系结构后我们居然可以将一条 SQL 从 42 秒调整到不足 0.01 秒；了解索引结构后，我们居然可以优化我们身边最为常见的 SQL 语句，比如 COUNT(\*）、MAX()等。而在此之前，这些语句根本不会让他对索引产生联想。他终于彻底明白了什么叫知识落地，明白了意识有多么重要！

我最后强调，不只是 Oracle，其实学习任何技术都是一样的，没有思考过你所学的某项技术有什么用，没有想过如何落地，如何应用到实际工作中去，都是毫无意义的学习，纯粹浪费生命。

### 3. 选择技术使用场景

近来小白连续犯错，他学习了并行度这个章节后，知道并行可以有效地利用多个 CPU，就将自己的代码加入了并行度。但他忽略了生产系统不只运行他这一个应用这个事实，结果导致代码上生产后系统资源大量争用。直至最后我们定位到问题在他的代码中，将写法修正，取消了并行度后，系统终于恢复正常。

他为什么会犯这个错误？主要是因为他只专注于技术本身而忽略了应用的场景，如果是凌晨

2 点的某个大任务操作，他的这个设置就很好，因为那个时刻大多其他应用都已经停止运行了，资源不用白不用。

这就是什么时候选择什么技术。小白还有一个有趣的故事，就是在我给他提这第三个建议之后的一周，他又使用了并行，这次他不是在代码中写死，而是选择临时性场合使用，在凌晨 2 点运行系列特定脚本。结果这次运行的比想象的慢得多，还不如之前未用并行的时候。

大家知道是什么原因吗？其实还是因为不知道什么时候选择什么技术。这次他不会影响他人了，因为凌晨静悄悄。但是他影响了自己。因为他的任务的特点是小而多，平均不到 0.01 秒可以完成一条 SQL 执行。他忽略了另外一个细节，并行是需要调度的，调度是需要开销的，调度的开销甚至达到 0.1 秒，那当然是越跑越慢了。所以小任务实际上是不需要考虑并行的。

类似的故事还有很多，在本书中大家还会接触到什么时候选择什么技术的各类场景，这里暂且只说并行。

小白的故事说完了，其实要是早些年，他或许不会有这些苦恼，因为学习资料少，他即便不抓重点也不至于学到筋疲力尽；此外他的诸多失误都和性能有关，而早些年大多 IT 系统压力很小，在基本功能满足后性能问题几乎可以忽略不计，那小白也就没错可犯了。

再次强调，当今的 IT 建设项目，如果不对海量数据和高并发带来的性能问题进行有效的架构设计、部署规划，你的项目基本上就被判了死刑。而 Oracle 新版本为什么对应大量技术文档，很大一部分原因是 Oracle 新版本提供了更多的性能调优产品。为什么现在新技术如此繁多，其实也是源于此，难道内存数据库和列式数据及分布式数据库的产生，不是为了让系统跑的更快一些吗？只是这里要注意，它们不可能替代传统关系型数据库，因为它们都有各自适合的应用场合。小白的故事如图 0-2 所示。

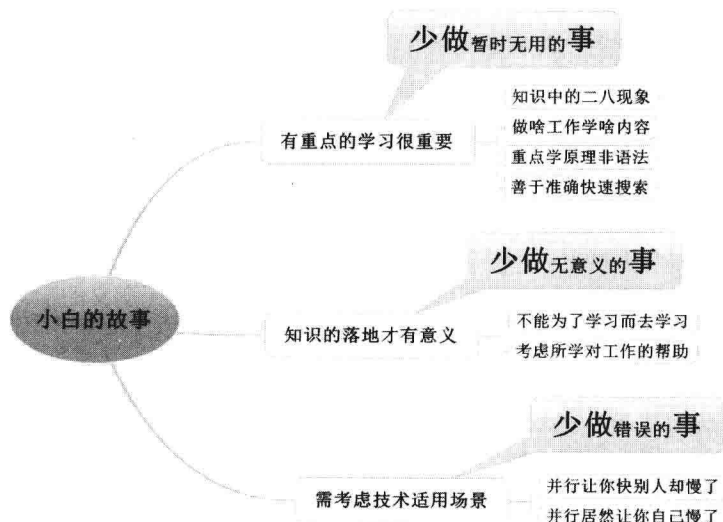


图 0-2 小白的故事

## 0.2.2 小刘的故事

### 1. 该如何请教他人

工作近 5 年的小刘特别勤学好问，工作中一遇到不明白的技术问题就问，有时问周围的同事，有时发帖子在论坛上提问。可是最近他非常郁闷，因为总是没人愿意回答他。是不是大家不够团结友爱？非也！问题出在他自己身上，你们知道为什么吗？

让我们来看看他是怎么问的吧。

“请问，对表的某列建索引如何建？”

“请问，分区表中删除分区后，索引会不会失效？”

“哦，您说我问过同样的问题了，我怎么记不起来了？”

前两个提问有问题吗？不少人都觉得应该一点问题都没有，其实是很有问题的。因为很显然第一个问题可以 Google 搜索到，而第二个问题除了 Google 可以搜索外，你完全可以通过做一个试验来得出结论。

网上可以很方便搜索到的东西去问别人无异于浪费别人的时间，当今这个时代，搜索能力成为了最重要的能力之一。知道关键字的比不知道关键字的搜索能力更强；英文好的比英文糟糕的搜索能力更强。为什么不去锻炼提升自己这方面的能力呢？显然可以动手试验证明结论而不去试验，会让你白白失去了一个宝贵的试验和总结的机会。

第三句对话暗示该同学经常问同样的问题却不自知，这说明了什么？别人告诉你答案时，你思考过吗，记录过吗，总结过吗？

### 2. 问题该怎样描述

某次我在外地出差，下飞机后打开电脑，收到了小刘的一封求救邮件，内容是这样写的：今天早上上班发现宁夏的数据库很慢，请帮忙诊断分析一下，万分感谢！

之前我还收过小刘的另外一封求救邮件，内容是这样写的：紧急求救，湖北数据库崩了，请尽快帮忙处理，谢谢！

从这两封邮件可以看出来，求救者严重缺乏经验，在沟通上存在着巨大的问题，你们知道是什么问题吗？

先说这个慢的邮件，是某个局部慢还是整个系统都很慢，不同情况的处理方式截然不同。是今天开始忽然变慢还是一直以来都慢，直至今天再也无法忍受，不同情况的处理方式显然是截然不同的。再说这个慢字，有一次我对求助者说，这个语句 1 秒就出来了，你怎么会觉得慢得受不了了？他的回答是，原先只需要 0.1 秒左右，这语句一天可是会跑上千万次啊。哦，原来 1 秒也让人觉得超级慢。还有一次有个求助者对我说，这个语句现在要跑 2 个多小时啊，原先超级快，只需要 5 分钟左右。哦，原来 5 分钟也有人觉得超级快。因此，你准确地告诉我现在要执行多长时间，希望执行多长时间，不是很准确吗？

再看第二封邮件，崩了？什么叫崩了，是数据库忽然关闭无法启动，还是系统运行太缓慢，还是数据文件丢失？哦，我来公布答案吧，后来知道这个崩了原来是指系统资源负载太高，系统

运行缓慢。原来这就是崩了，为什么要说这么模糊的字眼呢？

还有，这两封邮件都有个共同的特点，就是既没有提供接口人的联系电话，也没有提供需要处理的机器的 IP 地址、登录用户名和密码。

毫无疑问，接下来我要再确认好多次，才能最终帮上他。

3. 失误不只是粗心

小刘最近非常苦恼，因为近期系统经常遇到并行相关的等待事件，导致系统运行缓慢，后来查出来是诸多表和索引都有默认的并行度，这是因为系统很多大表通过并行的方式完成了创建，小刘创建完毕后忘记将并行度取消了。因为此事影响了生产系统正常运行，小刘被领导批评了，他也承认了自己过于粗心的毛病。

其实我们仔细分析就知道，这种问题绝对不只是因为粗心。显然有个更重要、更本质的东西没有揪出来，这就是流程和规范。

如果小刘在操作前事先准备好操作步骤，而不是即兴发挥现场操作，取消并行度的步骤应该就不会忘记。

即便他忘记了这个步骤，连脚本都没有体现，也没有关系。如果有规范，要求生产的操作准备了脚本，并且必须要他人审核通过方可操作，那也就不会出错了。

即便审核的人也犯错了，如果有个完善的流程，在生产中完成操作后必须检查哪些项目，那这个例行检查也必然会让错误再次避免。

人不是机器，都会失误，即便有的人素质非常高，不容易犯错，你也不能指望所有的人都这样，因此流程和规范才是最重要的。本书中大家会发现我们都有哪些流程和规范是用来避免误操作的。小刘的故事如图 0-3 所示。

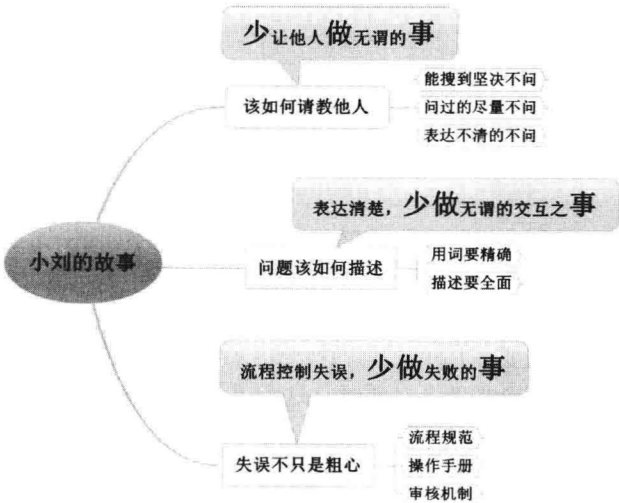


图 0-3 小刘的故事

## 0.2.3 自己的故事

### 1. 湖北神秘调优之旅

某日湖北某工程点向我求救，需要优化他们的生产系统，瓶颈正出在数据库上，为此公司让我去湖北现场进行调优。去之前我心中很没有底，因为据说这个性能问题已经困扰该工程点半年多了，期间该工程点已经请过好几波人马来调优过了，而且请来的人员中还不乏精通数据库的知名高手，这么多人这么长时间都无法把系统性能提升，我可以吗？

结果呢，我赶赴湖北现场忙碌一周后，该工程点生产系统性能最终得以大幅度提升，该工程点兴奋之余还发了一封表扬信到我公司，以此来肯定我的贡献。事后不少同事问我为什么那么多人长时间没解决的问题，而你一个人一周就搞定了，用了什么 Oracle 的高深技巧搞定这事。实际情况是，我并没有用任何 Oracle 技巧，甚至我根本就没有利用到和数据库有关的任何知识，但是我却解决了性能问题，圆满完成了任务。这是千真万确的事。

我到底做了什么事，让我来公布一下答案吧。我到了湖北，和现场技术支持人员以及相关人员对当前业务深入交流了一周，精简改写了大量逻辑被人为复杂化的 SQL，删除了部分多余的 SQL，对大表的保留记录情况根据业务特点进行了合理的规划，让不少大表变成了小表。

然后呢，然后就没有然后了，系统运行飞快，我胜利归来。这是一个很简单的道理，原先不少难以优化的运行缓慢的语句被我直接给咔嚓掉，从系统中消失了，此外不少大表变成了小表，系统变快不是自然而然的事情吗？

### 2. 三句话恢复的故障

某日早上 10 点，我接到项目组的紧急求救电话，被告知当天早上 8 点，某平台的生产系统运行极其缓慢，已经有相关人员在现场进行调查分析了，可惜两个小时过去了，直至现在依然没有解决问题，客户投诉不断，情况非常紧急。恰逢我在外地出差，且不说我在外地有重要的事情要处理，即便是立即飞回来也需要一天的时间，该如何解决这个问题呢？

结果呢？10 分钟后，在我的帮助下，问题解决了，故障恢复了。我做了什么事，这么神奇？

其实，我只在电话里和求助者交流了短短的三句话，求助者心领神会而去，然后问题就解决了。我是让他们调整什么神秘的参数吗？不！其实我用的方法和数据库的知识一点关系都没有，他们要是早点给我打电话就好了，免得相关技术人员折腾了两个多小时，却无法消除故障，实际上这个技术人员的水平并不低，甚至可以说是 Oracle 方面的行家。

到底是哪三句话这么神奇呢？

“系统是从什么时候开始慢的，是一直就慢还是突然变慢？”回答是今早上班忽然感觉缓慢，昨天下班以前都是正常的。

“那昨天你做了什么吗？”回答是昨天晚上升级了补丁。

“补丁允许回退吗，如果允许，你就回退补丁吧。”回答是系统都不能用了，如果回退可以解

决问题，当然可以回退。

接下来十分钟后，电话告知我，回退后系统正常了，问题解决了。

现在不影响生产系统的应用了，剩下下来的事情就是开发人员自己慢慢去检查代码有什么问题，最终问题在于代码有死循环和笛卡儿积，后续更改后投放生产就没问题了。

这其实就是生活，好比你去医院看病，肚子疼医生肯定会问你从什么时候开始疼，如果就是今天早上疼，那接下来必定会问你昨天晚上吃了什么……

### 3. 数据迁移有那么难吗

某日，某生产环境要做数据迁移，方法是通过 EXPDP 的方式将旧环境的数据库导出，然后通过 IMPDP 的方式导入到新环境，要求在凌晨 2 点开始导出，6 点前完成导入，因为之后需要测试，必须保证上班时间 8 点后，新系统可以正常运行。

这里涉及的技能是掌握 Oracle 的 EXPDP/IMPDP 命令，如果今天让我来上课描述这个工具的使用，大致 45 分钟时间可以完成课程的讲述，还包含了动手试验的时间。因此这显然不是一件很复杂的工作，当时现场的操作人员是一位工作多年的，拥有 OCM 证书的技术人员，大家也都觉得很放心。

然而实际情况是，从凌晨 2 点开始操作，直至下午 6 点，才完成了导入工作，整整比预计推迟了 12 个小时！这期间虽然临时做了不少应急补救措施，但是还是严重影响了生产的正常运营。

为什么会这么糟，中间出了什么问题？其实这里的故事太耐人寻味了，需要改进考虑的细节也太多太多了。

接下来揭晓谜底吧，还是通过一段对话展开。

“请问你要导出的库有多大？” 答曰不知道。

“请问你要导出的库最大的对象有多大，能否列举前 20 名的大对象？” 答曰没统计。

“请问是否是所有的对象都需要导出？” 答曰应该是，没确认。

“请问你知道导出和导入机器的 CPU、内存配置情况吗？” 答曰没注意。

OK，这就是原因了，让我来告诉大家我调查的结果吧，全库有 1TB 这么大，其中某记录操作日志的表 T1 就达到 400GB，连同索引，大小合计 500GB 左右，近乎一半的量。而和业务人员确认的结果是，T1 表可以暂且不导出，只需在新环境中建表结构即可。前 20 名的大对象除了巨无霸 T1 表外，还有不少也很庞大，其中有一张单表即有 120GB，只需保留最近三个月即可……经过我的确认，1T 的数据最终只需要导出 300GB 左右。

接下来的调查还发现了更加惊人之处，导出和导入的机器 CPU 个数居然都达到 64 个，强劲得让人惊叹！而我观察操作的脚本，也有写并行度，只是随便写了 PARALLEL=8。当时是凌晨，我们完全可以将导出导入的 PARALLEL 设置为更大，比如 60 个都不为过，这里又将会有 5 倍以上的提速。