dbaplus 社群丛书

Data Blockchain AIOps

Broadview

收获，不止Oracle 第2版

梁敬彬 梁敬弘 | 著

非外借

经典力作 2019再版

基于Oracle 12c全面升级

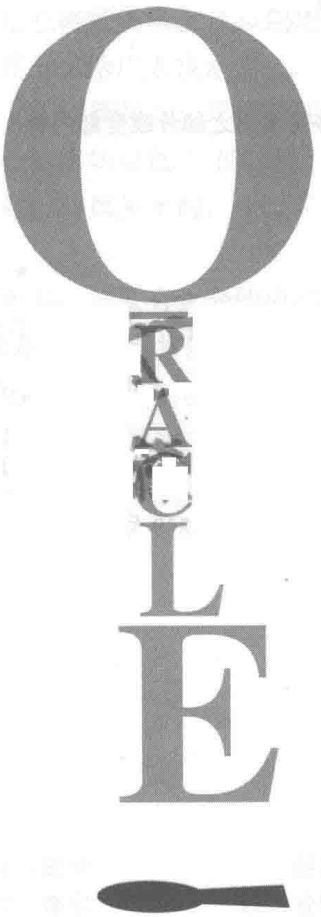
中国工信出版集团

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

dbaplus 社群丛书

收获，不止Oracle 第2版

梁敬彬 梁敬弘 著



電子工業出版社
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。通过阅读本书中的这些章节，读者将会在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，不仅能解决工作中的常规问题，还能具备一定的设计和调优能力。通过对这些章节的学习，读者在 Oracle 的学习中一定会有极大的收获。

然而，作者更希望看到的是：让读者的收获，不止 Oracle。

为达到此目的，作者精心将全书分成上下两篇，刚才所描述的具体知识点体现在全书的上篇中。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。

图书在版编目（CIP）数据

收获，不止 Oracle /梁敬彬，梁敬弘著. —2 版. —北京：电子工业出版社，2019.4
（dbaplus 社群丛书）
ISBN 978-7-121-36092-3

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（2019）第 039839 号

策划编辑：张月萍
责任编辑：刘 舫
印 刷：天津嘉恒印务有限公司
装 订：天津嘉恒印务有限公司
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：787×1092 1/16 印张：29 字数：724 千字
版 次：2019 年 4 月第 1 版
印 次：2019 年 4 月第 1 次印刷
印 数：4000 册 定价：99.00 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）88254888，88258888。

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。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

推荐序一

这是一本非常有趣的技术书，全书以对话穿插故事的形式完成编写，让人在轻松愉快中了解各种知识，构思让人拍案叫绝。也正是由于如此出色的构思设计，本书第 1 版多次印刷，依然售罄。应业界的强烈要求，结合 Oracle 12c 的第 2 版新书终于面世了。

梁敬彬是我们公司的特级专家，他服务的客户遍布全国各地，并取得了良好的口碑。他能在十分忙碌的工作之余，雷打不动地坚持每天早上 4 点起床整理技术知识，把自己宝贵的工作经验撰写成生动有趣的经典技术著作分享给广大技术人员，我非常欣赏技术人的这份激情。

不过，我更欣赏的是梁敬彬身上的创新精神。梁敬彬同时也是我们研究院的副理事长，是研究院许多好点子的主要贡献者之一，比如设置了 EX 币、创办了电子期刊、发布了悬赏令，策划主持了技术比武等，将研究院运作得风生水起，引发了许多电信内部兄弟公司的兴趣，并在电信集团层面进行了分享。

一个历史悠久的公司能够永葆青春，需要有优秀的内部创业文化，每个专家都要有创新精神，敢于否定自己的过去。梁敬彬就是我们公司许多有着创新精神员工的一个代表、一个缩影，希望更多员工能为公司多多建言献策，未来聚焦“两外两 T”业务，在互联网化规模发展、机制创新、文化建设等方面进一步加快步伐，

期待公司出现越来越多的创新人才，以给我们带来更多的惊喜。

吴刚

中电福富公司总经理

推荐序二

经典升级再现，技术传播孜孜不倦

敬彬兄的经典著作《收获，不止 Oracle》要再版了，邀请我写一篇推荐序。因为这本书本身写得很好，推荐序于我而言就不太好写了，相形见绌反而不美，那我就写写与敬彬相识相知的过程吧。俗话说，文如其人，人如其文，真诚的人写的书也定是真诚的。

跟敬彬是在 ITPUB 时代的老友，大家在论坛里就各种问题互相回复，纯然的技术友谊。第一次见面，应该是九华山会议，ITPUB 请了一些版主见面聚会，敬彬给人的感觉就是温文尔雅，彬彬君子。

第二次见面，是在 2013 年的 DTCC 会议上，我当时分享的议题是《大型业务系统 Oracle 数据库 10g 到 11g 的升级实践》，而敬彬分享的则是《赢在起点——谈数据库设计规范》。我有幸聆听敬彬的分享，一个富有经验的老师，将生活中的故事揉进枯燥的技术里，娓娓道来，引人入胜。之后敬彬的《收获，不止 Oracle》和《收获，不止 SQL 优化》两本书相继出版，秉承了他的演讲风格，内容提纲挈领，文字风趣幽默，讲解深入浅出，让读者更易于吸收。

dbaplus 社群成立后，敬彬在繁忙工作之余，多次参与社群的线上分享和线下峰会，用他独特的方式持续传播 Oracle 技术。如今他又投注精力将 Oracle 12c/18c 的新特性及这些年工作生活的新积累更新到《收获，不止 Oracle》（第 2 版）中，实乃读者之福。

杨志洪

dbaplus 社群联合发起人，Oracle ACE，Oracle10g/12c OCM

推荐序三

别出心裁，另辟蹊径

敬彬的新书就要出版了，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》，其内容鲜活、行文引人，于是就和他约了那篇稿子加入书中，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，作者能够用细腻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，就能够消除一些人对于这项技术的畏惧之心，此后的学习自然就能顺风顺水。

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，就会觉得这项技术实在是趣味横生。

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

推荐语

梁先生的技术功底和文字功底同样深厚，更重要的是，他具有作为讲师那种缜密、体系化的思维方式，以及对读者心思的透视力，因此成书脉络清晰，里面还不断穿插许多人生哲理、技术前瞻，让人获益良多。这本书非常适合入行者和在行业里谋求上升的同仁阅读，动人的文笔可以让你一口气读完这本书。这是一本值得向行业推介的优秀技术书籍。

黄志洪（tigerfish）
Dataguru 创始人

到底是什么后天原因导致人和人之间的学习结果发生重大差异呢？其中有一点就是思维方式。于是我尝试在思维方式方面去影响身边的一些人，其中的很多人在各自的工作领域都获得了成功，而在此之前他们却是默默无闻的。

敬彬的这本书就是用诙谐幽默的语言生动地引导大家在意识层面发生改变，然后逐步转化为行动上的改变。按此坚持，几年下来，相信每个人都能迈上新的台阶，这的确早已超越 Oracle 的范畴，对我们学习、生活等诸多方面都有益处。

冯春培（biti_rainy）
支付宝平台数据部资深总监

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经验，付出了超于常人的努力，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
中电福富公司运营总监

敬彬兄的书笔触轻松而生动，在内容表现形式上使用了故事和对话场景，让读者在阅读时有很强的代入感；书中展现了大量的脑图总结，清晰直观，十分便于知识的理解和记忆。此书

无论对于新手还是资深技术专家，都具有极大的借鉴意义。在 Oracle 技术已然成熟的今天，带给 DBA 的挑战和机遇依然有增无减，希望敬彬的这本书能成为你 Oracle 学习生涯的起点。

杨建荣
dbaplus 社群联合发起人，Oracle ACE，《Oracle DBA 工作笔记》作者

由梁敬彬、梁敬弘兄弟合著的《收获，不止 Oracle》一书出第 2 版了。第 1 版取得了很大的成功，成为业界数据库书籍中的经典，堪称不易。该书第 2 版是响应业界的声音而再次出版的，融入了 Oracle 的新特性，预祝新书再次获得成功，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

看过很多 Oracle 的书，让我印象深刻的并不多，梁老师的这本书是我给新员工推荐的必读书。这本书寓教于乐，将做事做人的道理潜移默化地传授给读者，虽然从事的是国产数据库的事业，此书依然能够给我们带来很多启发，收获确实不止 Oracle，你值得拥有！

黄海明
达梦技术总监

敬彬兄这本书有着与市场上其他 Oracle 书籍不同的特点，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

梁敬彬先生通过自己在日常工作和培训中的磨炼，把自己对 Oracle 技术的感悟，通过一个个生动鲜活的小故事，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

前言

迈出崭新一步——本书特点及存在的意义

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学习参考资料。除了可以在各技术平台的官方网站下载到详尽且准确的资料，还可以很容易地在各种搜索引擎中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。

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呈指数级增加，让开发设计人员头昏脑涨，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，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 版本到今天的 Oracle 12c，Oracle 每一次发布新版本都需要我们投入大量的精力和时间去学习和适应。

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。

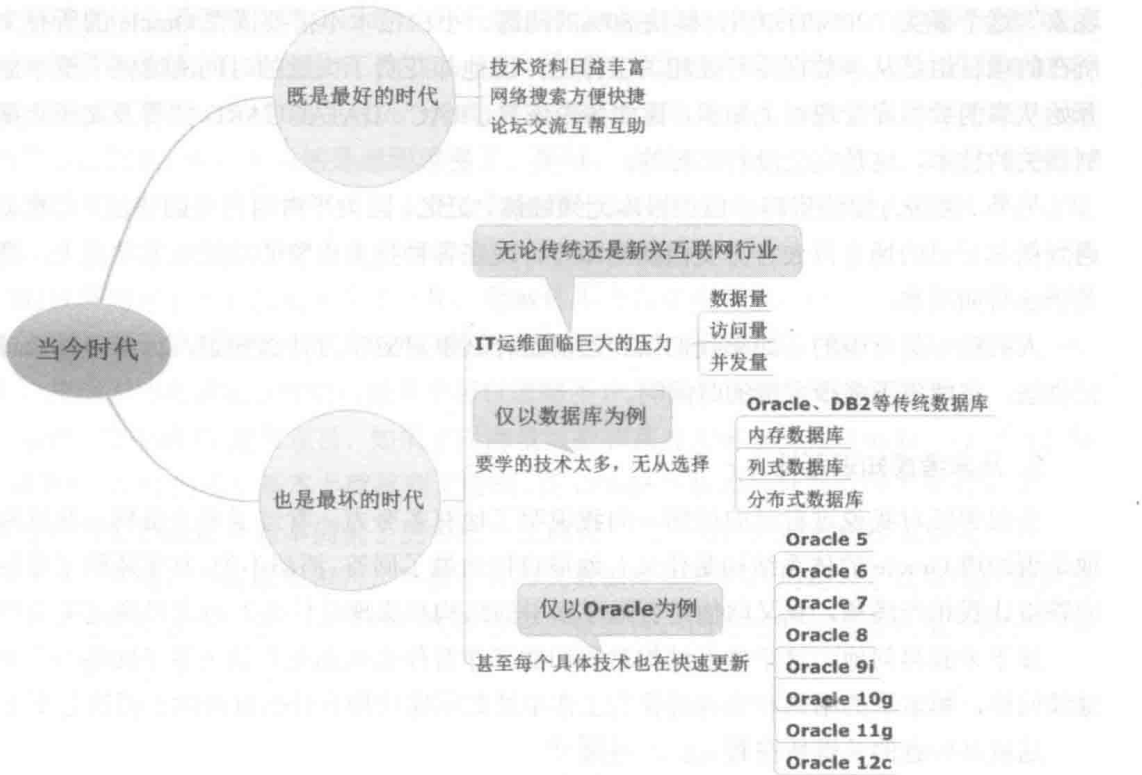


图 0-1　当今时代

0.2　技术人员，真正的差距其实在意识

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　小白的故事

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 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被称为“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 的所有文档，他所在的项目组是从事数据库开发相关工作的，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，现在各种搜索引擎的功能也非常强大，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 的体系结构是什么？他很自信地做了回答，滔滔不绝，甚至还画了草图给我，回答得让我相当满意。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，这个体系结构对我们的工作有什么帮助呢？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什么好处呢？再次答不上来。

这就是问题的关键所在！

知识要落地，要思考应用的场合，这就是我除了让他掌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 指令的执行时间从 42 秒优化到不足 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，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，结果导致代码运行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，因为那个时刻其他大多数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，其实要是在早些年，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功能。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，不是为了让系统跑得更快一些吗？只是这里要注意，它们不可能替代传统关系数据库，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。

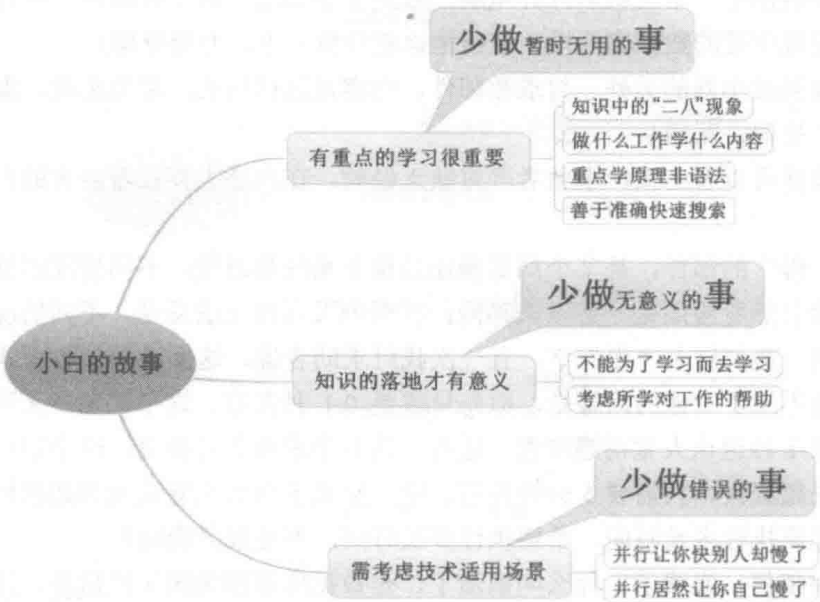


图 0-2　小白的故事

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的问题。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在搜索引擎中搜索到，而第二个问题除了可以搜索外，完全可以通过做一个试验来得出结论。

网上可以很方便地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！

之前我还收到过小刘的另外一封求救邮件，内容是这样写的：紧急求救，湖北数据库崩溃了，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 秒就执行完了，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，这个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，崩溃了？什么叫崩溃了，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“崩溃了”原来是指系统资源负载太高，系统运行缓慢。原来这就是崩溃了，为什么要说这么模糊的字眼呢？

还有，这两封邮件都有一个共同的特点，就是既没有提供接口人的联系电话，也没有提供所需要处理的机器的 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能知道，这种问题绝对不只是因为粗心。显然有一个更重要、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，就不会忘记取消并行度的步骤。

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，如果有一个完善的流程，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。在本书中大家会发现我们都有哪些流程和规范可用来避免误操作。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。去之前我心里很没有把握，因为据说这个性能问题已经困扰该工程点半年多了，在这期间该工程点已经请过好几拨人马来调优过了，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的？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、运行缓慢的语句被直接去掉，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变慢的，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，回退补丁后系统正常了，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的，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 的方式将旧环境中的数据库导出，然后通过 IMPDP 的方式将数据导入到新环境，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 点前，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 分钟的时间可以完成课程的讲述，其中还包含动手试验的时间。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，才完成导入工作，整整比预计推迟了 12 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。

好了，这就是原因了，让我来告诉大家我调查的结果吧。全库有 1TB 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，近乎占一半的量。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 名的大对象中除了巨无霸 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的记录即可……经过我的确认，1TB 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，导出和导入的机器的 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，但只是随便写了 PARALLEL=8。当时是凌晨，我们完全可以将导出导入的并行度设置得更大，比如设成 60 都不为过，这里又将会有 5 倍以上的提速。

后面我告诉他们，其实这次导出导入只需要合计 2 小时就足够了，不需要 4 小时，更不会操作了整整 16 个小时！

4. 充分准备是要领

某次我应邀去安徽某工程点进行数据库调优，周二晚上接到通知，要求周三下班前赶到现场，周四一天之内要把系统性能显著提升，从而应对周五集团的检查工作，当时公司领导也在现场，情况相当紧急。

这是一件相当让人头疼的事，因为系统调优是一项相对复杂和艰巨的任务，要求在一天时间内立竿见影，可能性很小。虽说没有把握，但是也只好硬着头皮去试试看，我在出发前做了充足的准备，这些准备方法大多来自平时的积累，这种积累非常宝贵、非常重要。有了这些重要的准备，我在出发前通过工程点现场同事的配合，获取了大量我想了解的重要信息，在路上和飞机上做了充分的分析和研究，等下午 6 点到工程点现场，我已经对这里的情况了如指掌。

然后，晚上调优到凌晨 2 点，第二天再奋战一天，第三天，系统终于运行平稳了，IDLE 从原先的 0～1%变为 40%～50%，系统性能显著提升。

这个小故事说明了一个道理，积极的准备是很重要的。我准备了一个非常详细的处理及定位问题的流程和思路，从动态到静态，从整体到局部，非常详尽和实用，将会在后续分享给大家。读者可以从中体会到我是如何定位以及分析探索问题的，此外，还可以学习到我所总结的优化思想和方法论。我自己的故事如图 0-4 所示。

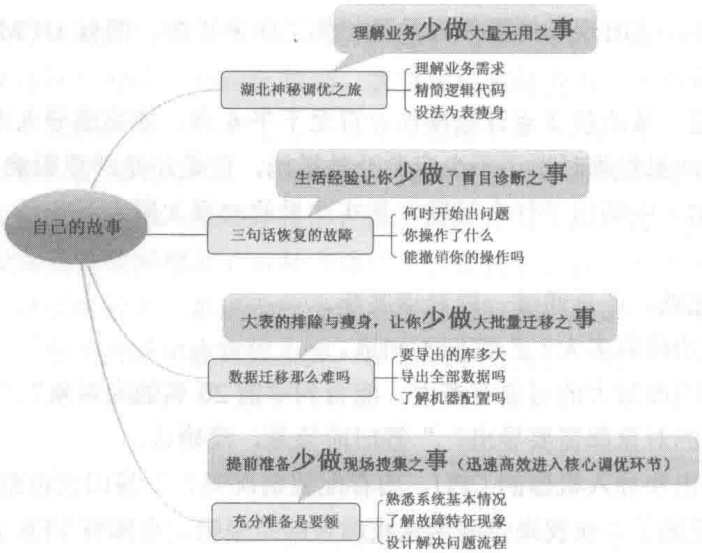


图 0-4　自己的故事

0.2.4　故事的总结

至此我讲完了小白、小刘和我自己的故事。别小看这三个人的故事，其实都是非常典型的。这其中小白的故事告诉了我们学习是很有技巧的，是一门技术活。而小刘的故事告诉我们如何提问、求助以及制定规范是非常重要的。最后我本人的故事告诉大家，很多时候，解决问题不一定完全依靠技术本身，我的 4 个故事中的制胜法宝全部是非技术的，这说明技术虽然重要，但是不能仅仅依赖技术。技术人员的故事如图 0-5 所示。

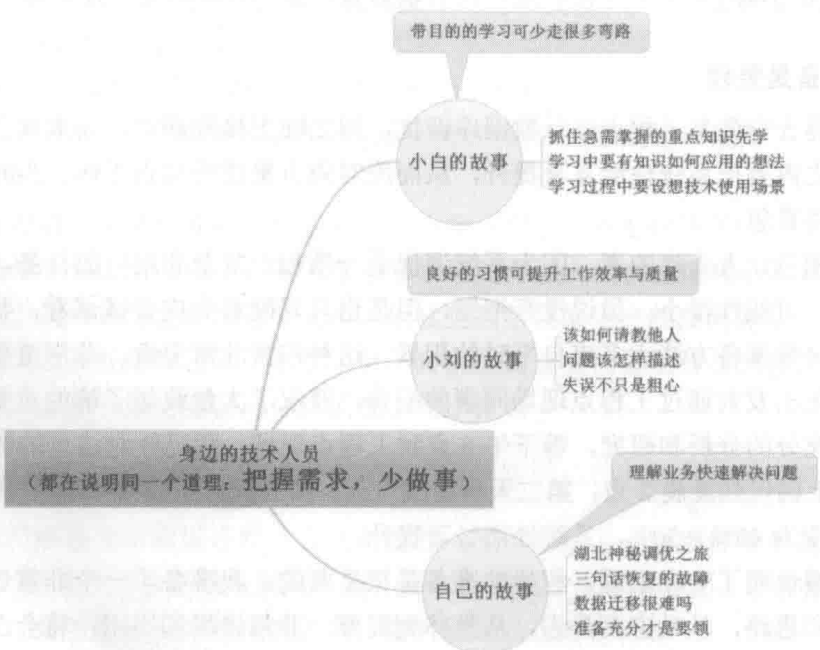


图 0-5　技术人员的故事