

原创经典

感悟

核心技术

Oracle

资深技术顾问罗敏
继《品悟性能优化》之后，又一力作

罗敏 著



Oracle
资深技术顾问

罗敏

继《品悟性能优化》之后，又一力作！

生动的案例实景 严谨的技术描述 鲜活的人生感悟 轻松诙谐的语言

设计和优化篇

11g新特性篇

故障诊断篇

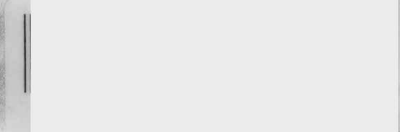
运行维护篇

架构篇

服务篇

包含IT系统全生命周期分析

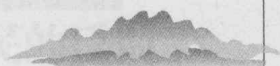
清华大学出版社



感悟

Oracle 核心技术

罗敏 著



清华大学出版社
北京

R311.138
2955

内 容 简 介

作为 Oracle 公司核心技术, Oracle 数据库自 1977 年开始研发并逐步成熟, 已经在全球众多行业得到了广泛而深入的应用, 市场占有率一直雄居数据库市场之首, 从事与 Oracle 数据库技术相关的 IT 人士更是有以计数。

相比 IT 行业其他技术, Oracle 数据库技术是一门门槛并不太高的技术, 刚走出校门的大学生, 也许只需学会几条基本的 SQL 语句, 然后再学学 Java, 会配置 JDBC, 就基本上可以成为数据库应用开发人员了。但以作者从事 IT 行业二十余年的经验, Oracle 数据库其实是一个非常深奥的技术领域。虽然产品历经三十多年发展, 已经非常成熟而稳定, 但是 Oracle 公司为了满足业务发展需要, 依然在老树发新芽般地全力发展这一核心产品, 例如 12c 的数据库云计算等。而且, 即便在这一传统技术领域, 在架构设计、应用开发、运行维护、变更管理等多个领域, 依然让广大客户处处感觉到 Oracle 的博大精深, 真有说不尽、道不完的感慨。

因此, 围绕 Oracle 数据库这一核心产品和技术, 在设计、开发、运维等 IT 系统全生命周期, 作者结合本人在原厂多年从事服务工作的实施案例, 并发出些个性化的感悟, 这就是作者写作此书的目的、内容和特点了。希望这本书能给广大 IT 人士, 特别是从事与 Oracle 数据库相关的同行们带来裨益。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

感悟 Oracle 核心技术 / 罗敏著. —北京: 清华大学出版社, 2015

ISBN 978-7-302-37838-9

I. ①感… II. ①罗… III. ①关系数据库系统 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2014)第 199311 号

责任编辑: 冯志强 薛 阳

封面设计: 吕单单

责任校对: 徐俊伟

责任印制: 宋 林

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社总机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 31.25 字 数: 761 千字

版 次: 2015 年 1 月第 1 版 印 次: 2015 年 1 月第 1 次印刷

印 数: 1~3500

定 价: 69.00 元

自序

“老罗，啥时候出‘月子2’啊？”

自从本人第一本书《品悟性能优化》在2011年5月出版之后，经常会被周围同事、朋友和熟悉的客户这么调侃道：“老罗，啥时候出‘月子2’啊？”我的自嘲之一是：“等村头厕所没手纸了吧。”自嘲之二是：“再等十年吧。”——因为《品悟性能优化》一书被人誉为“十年铸剑”。

时至《品悟性能优化》出版发行一年多之后，发现该书在当当、京东等网站的数据库类图书畅销榜中，偶尔还能排在前列，也听清华大学出版社说该书销售和总体反馈情况还不错。于是，斗胆猜想村头厕所的手纸可能真快没了，到了该补充的时候了。呵呵。

再等十年？IT技术发展这么快，十年之后再写今天的技术，也许又成昨日黄花了。在写作《品悟性能优化》的2010年，Oracle 11g在国内还没有得到广泛应用，自己也缺乏系统的研究和运用，当时主要是以10g为写作对象，网民也发现该书对11g新东西涉猎并不多。因此，以写新书为契机，一边研究和实践新技术，一边总结并付诸笔墨，也不失为一种激励自己不断进取和前行的动力了。

来自网民的反馈

非常感谢《品悟性能优化》一书出版之后，广大读者和客户给予该书的各種中肯的评价。无论是令我耳根发红的褒奖，还是善意的调侃，甚至拍砖，我都能欣然接受，并细细品味。以下就是摘取的部分网民有代表性的评论，以及我在此的回应。

- “很好的一本书，一本不谈技术细节的技术书。呵呵”

谢谢该网民，他（她）看出了我写作该书的初衷之一：“与其让您分享更多的命令、脚本，不如让您在性能优化方法论和思考问题方式方面，同样引起共鸣，包括结合您自己工作中的人生感悟。”

- “挺好的，优化更多的是靠扎实的基础，而不是非常手段。”

是的，我在《品悟性能优化》一书中反复强调20/80规则之一：20%基础技术能解决80%的IT系统问题。许多IT系统问题的确是20%基础技术都没有得到扎实的运用所导致的。例如，很多设计开发人员连复合索引的基本原理都没有完全掌握。

- “没有实际工作经验的人，可能对这本书体会不深，我反正已经看到兴奋地跳到桌子上了。”

谢谢这位老兄（老妹），谢谢这位知己，也谢谢这位性情中人。做技术，特别是做Oracle的，就需要这么激情四射，甚至个性张扬。因为Oracle数据库博大精深，的确有很多让人拍案叫绝的精彩技术。我也曾经这么说过，从事技术工作的，一定要充满激情。你投入了

100%的激情，客户可能被你感染了 70%，只有你 120%，甚至 150%的投入，客户才能 100%地被你所感动。

- “佩服作者的文笔，内容一般，内容没有书名牛。”

呵呵，也非常感谢这位老兄的点评，因为至少你还肯定老罗的文笔尚可。内容的确有很多欠缺，弥补缺憾正是我写作新书很重要的动机。书名嘛，虽然不是我一个人独创，但的确有点故做深沉、假扮高雅了。这次一定起个让大家更能接受、更平和的书名。

- “这本书就是记录作者作为一个 Oracle 咨询顾问如何和客户沟通及忽悠的故事书，可以作为小说来看。”

绝了！佩服该网民锐利的眼光，居然看出老罗在 Oracle 公司从事的工作性质和角色了。所谓的解决方案服务顾问，或者说服务售前顾问，就是不断与客户沟通，了解现有 IT 系统现状、问题和需求，并制定服务方案的工作。说白了，就是忽悠客户买 Oracle 服务。但是，鄙人认为忽悠也是需要扎实技术背景和丰富实践经验的哟，呵呵。

故事嘛、小说嘛，更能吸引读者，让大家感到一种轻松和愉悦。如果只有干巴巴的技术，大家还不如直接看 Oracle 参考手册。有位销售同事说：“老罗，我花了一个晚上就看完了你的书，因为我跳过了所有技术细节，只看故事，很有意思。”呵呵。

- 更多褒贬不一的点评

“必须五星！读过的最有可读性，同时兼具实用性的 Oracle 性能优化的书籍，一口气读下来的，现在刚读到一半就上来发评论！虽然从 DBA 的角度看可能有些角度不够深刻，但对于实际项目的调优，特别是开发人员有很现实的指导意义！”

“里面的内容同我的实际经验一样，没有浪费笔墨在无多大实际意义的内部参数上，而是平实地写怎么优化程序。”

“个人角度出发，举例没有那么细致啦，凑合着看吧。”

“罗老师这本书，写得很风趣，技术图书让人觉得不枯燥，又讲解得比较细致。”

“作者实际调优经验丰富，案例能引人入胜，但涉及的具体技术细节并不多，可能会觉得其中技术实用性不大，但对一般技术人员来说足够了！非常推荐。”

.....

本书的内容、体裁和风格

既然有了再次写作的冲动，特别是得到了那么多读者、网民的善意点评和建议，那么在新书的内容、体裁和风格方面就有了如下的规划了。

本书的内容

与《品悟性能优化》一书不同的是，新书将不再以性能优化为主线，而是划分为设计和优化篇、11g 新特性篇、故障诊断篇、运行维护篇、架构篇、服务篇等，将涉足数据库逻辑设计、物理设计、常见故障诊断、数据库空间管理、版本和补丁管理、数据库安全性、数据库升级、12c 等具体专题，例如在 11g 新特性篇中将介绍数据压缩、SPM、SQL Query Result Cache、SecureFiles 等新技术。最后，还会介绍 Oracle 公司最新的服务体系和服务产品。

本书的体裁

既然涉足那么多领域，不可能每个专题都是那么全面而系统。为避免蜻蜓点水，决定本书以技术杂文形式，针对某个领域的某个技术专题展开深入讨论。根据各专题的不同，内容可能包括该技术原理的介绍、技术运用过程、相关案例，最后都会包括对该技术的感悟和点评。

为弥补《品悟性能优化》一书中对某些技术细节叙述不足的缺陷，本书尽量在每个技术专题中都更加细致，并且更具有实用性，同时介绍相关参考资料及进一步读物。

虽似技术杂文体裁，但希望最终结果是杂而不乱，或者看似杂乱，但求有序而实用。

本书的风格

感谢大部分读者对《品悟性能优化》一书风格的肯定，因此，新书仍然将继承并发扬这种风格，即叙述技术追求严谨，在案例、感悟和点评中则继续保持适度的轻松、调侃和个性化特点。前者是重点，后者是为了提高书的可读性，也是为了让大家在辛苦阅读大量技术细节之后，得到短暂的休憩和放松，甚至报以会心地一笑。

综合上述想法，特别是考虑本书是围绕 Oracle 数据库这一核心技术在展开叙述，因此书名为《感悟 Oracle 核心技术》。

好了，正式开工了！

罗敏

目录

1. 设计、优化篇

第1章 逻辑设计中的那些事	3
1.1 还是从案例开始	3
1.2 什么叫规范化设计?	4
1.3 回到案例	6
1.4 规范化设计的好处	7
1.5 “这系统是你设计的?”	8
1.6 规范化设计工具: Oracle Data Modeler	9
1.7 非规范化设计若干案例	11
1.8 关于数据库逻辑设计的感悟	16
1.9 授人以鱼, 不如授之以渔	19
1.10 本章参考资料及进一步读物	20
第2章 物理设计中的感悟	21
2.1 二十多年前的物理设计	21
2.2 一个项目的物理设计概貌	21
2.3 从总体上考虑物理设计	22
2.4 表和索引设计基本建议	23
2.5 表空间设计的新特性	25
2.6 表空间设计原则和详细设计	27
2.7 数据库初始化参数设计	29
2.8 物理设计中的感和悟	31
2.9 本章参考资料及进一步读物	33
第3章 再谈海量数据库设计、开发和管理	35
3.1 现有系统分区方案分析	35
3.2 大集中系统分区方案	36
3.3 深入探讨全局分区索引	37
3.4 11g 分区新技术	42
3.5 再次品味分区的甜与苦	45
3.6 本章参考资料及进一步读物	48

第 4 章	性能优化：百谈不厌的话题	49
4.1	重温《品悟性能优化》一些重要观点	49
4.2	深入剖析一个测试题	53
4.3	又一次“救火”经历	55
4.4	某综合报表平台的优化	61
4.5	适合于数据仓库的 Bitmap 索引	69
4.6	半小时都没听懂的 Bitmap Join 索引	72
4.7	什么叫精通 Oracle?	74
4.8	一次典型故障引发的思考	75
4.9	本章参考资料及进一步读物	80
 2. 11g 新特性篇		
第 5 章	11g 新技术/新功能使用策略	85
5.1	我被客户“打懵”了	85
5.2	11g 新技术、新特性一瞥	87
5.3	一个纠结的话题	91
5.4	11g 新技术实施总体策略	94
5.5	本章参考资料及进一步读物	95
第 6 章	11g 的数据压缩技术	97
6.1	为什么 Oracle 压缩技术运用不普及?	97
6.2	11g 压缩技术概述	98
6.3	深入剖析 Oracle 压缩算法	102
6.4	一个实际案例的分享	105
6.5	数据压缩相关技术点	106
6.6	其他压缩技术	108
6.7	数据压缩技术运用建议	111
6.8	本章参考资料及进一步读物	112
第 7 章	11g 性能优化新技术：SQL Query Result Cache	114
7.1	Result Cache 原理	114
7.2	Result Cache 使用过程	117
7.3	Result Cache 的管理	118
7.4	Result Cache 相关技术点	121
7.5	客户端 Result Cache 技术	122
7.6	适合 Result Cache 的典型案例	124
7.7	本章参考资料及进一步读物	126

第 8 章	11g 性能管理新工具: SPM	128
8.1	一些典型场景	128
8.2	SPM 原理	129
8.3	SPM 使用过程	130
8.4	SPM 适用场景	132
8.5	SPM 相关技术点	136
8.6	本章参考资料及进一步读物	138
第 9 章	11g 大对象数据新技术	140
9.1	传统 LOB 技术的运用	140
9.2	传统 LOB 技术的不足	143
9.3	新一代大对象处理技术: SecureFiles	143
9.4	SecureFiles 相关技术细节	144
9.5	SecureFiles 迁移方法	145
9.6	SecureFiles 实施案例	147
9.7	本章参考资料及进一步读物	151

3. 故障处理篇

第 10 章	话说故障诊断	155
10.1	通过案例看故障诊断	155
10.2	没完没了的收集数据	162
10.3	数据库常见诊断工具	163
10.4	数据库挂起诊断信息的收集	171
10.5	一些“自主知识产权”脚本	179
10.6	一个项目中的故障诊断	185
10.7	本章参考资料及进一步读物	191
第 11 章	那些常见的 Oracle 错误	193
11.1	ORA-00600: 内部错误	193
11.2	ORA-04030: PGA 内存不够	196
11.3	ORA-04031: Shared Pool 内存不够	198
11.4	空间不够的问题	203
11.5	ORA-00376: 数据库文件不可读	206
11.6	ORA-01555: 快照太旧	208
11.7	ORA-30036: UNDO 表空间无法扩展	211
11.8	日志切换频度过高问题	213
11.9	故障处理的感和悟	218

11.10 本章参考资料及进一步读物	219
第 12 章 数据库坏块处理技术	221
12.1 可怕的数据库坏块	221
12.2 坏块处理的主要流程	223
12.3 坏块处理八卦图	229
12.4 DBMS_REPAIR 包或设置 10231 事件	232
12.5 ROWID 扫描方法	234
12.6 如何从坏块中抢救数据?	236
12.7 坏块抢救的最后招数	237
12.8 本章参考资料及进一步读物	238

4. 运行维护篇

第 13 章 漫谈数据库健康检查	241
13.1 什么是数据库健康检查?	241
13.2 多年前的一次健康检查	242
13.3 最近一次健康检查	246
13.4 关于健康检查的点评	249
13.5 11g 健康检查新特性: Health Monitor	250
13.6 本章参考资料及进一步读物	254
第 14 章 防范人为操作失误的最好技术: Flashback	255
14.1 人为错误的防范	255
14.2 数据库级快速恢复	258
14.3 错误删除表的快速恢复	261
14.4 表级快速恢复	263
14.5 记录级快速恢复	264
14.6 Flashback 在应用测试方面的运用	266
14.7 Flashback 在安全审计方面的运用	268
14.8 Flashback 在容灾方面的运用	269
14.9 Flashback 与传统数据恢复技术的综合运用	271
14.10 11g 新技术: Total Recall	272
14.11 Flashback 技术综合对比	273
14.12 本章参考资料及进一步读物	274
第 15 章 关于数据库碎片管理	276
15.1 数据库空间碎片问题	276
15.2 碎片评估方法	278

15.3	碎片整理方法	285
15.4	实施策略、组合和流程	290
15.5	本章参考资料及进一步读物	293
第 16 章	Oracle 版本、Bug 和补丁	295
16.1	关于 Bug 和补丁的一个典型故事	295
16.2	Oracle 版本和补丁概念	296
16.3	主动安装补丁是防范故障的最有效办法	298
16.4	Oracle 数据库版本和补丁集发布时间表	299
16.5	补丁实施那些事	302
16.6	一位客户的 SR 处理	305
16.7	本章参考资料及进一步读物	309
第 17 章	一个热门话题：数据库安全性	311
17.1	数据库安全性需求及现状	311
17.2	Oracle 数据库安全性解决方案	312
17.3	某银行客户安全性需求和 Oracle 策略	314
17.4	从安全性评估开始	318
17.5	本章参考资料及进一步读物	329

5. 架构篇

第 18 章	再谈 RAC	333
18.1	客户哑口了	333
18.2	RAC 实施方法和实施内容	336
18.3	11g RAC 高可用性方案设计	337
18.4	11g RAC 高可用性测试	341
18.5	某项目的 11g RAC 实施内容	355
18.6	11 件加固 RAC 环境的事情	362
18.7	本章参考资料及进一步读物	365
第 19 章	话说升级	367
19.1	关于数据库升级的疑虑	367
19.2	为什么要升级？	369
19.3	Oracle 升级方法论介绍	371
19.4	现状及升级改造需求分析	374
19.5	升级和迁移技术方案	374
19.6	升级中的性能优化和性能管理	382
19.7	升级项目的实施和组织	384
19.8	升级风险评估控制	385

19.9	有感于某移动公司的升级案例	388
19.10	本章参考资料及进一步读物	389
第 20 章 数据库私有云技术		391
20.1	云计算概述	391
20.2	不同层次的云计算	393
20.3	某案例的基础架构云计算实施	394
20.4	数据库云计算中的典型技术	397
20.5	一次尴尬的拜访经历	400
20.6	数据库云计算案例分享	401
20.7	本章参考资料及进一步读物	406
第 21 章 尝鲜 Oracle 12c		407
21.1	新特性培训课的趣事	407
21.2	12c 架构方面的最大变化	408
21.3	CDB 和 PDB 的创建、启动和关闭	412
21.4	CDB 和 PDB 的日常管理	420
21.5	信息生命周期管理的挑战和 12c 解决方案	424
21.6	Heat Map 和 ADO 详细技术	426
21.7	数据归档新技术	432
21.8	“貌合神离，貌离神合”	436
21.9	12c 实施案例	438
21.10	本章参考资料及进一步读物	441

6. 服 务 篇

第 22 章 再说 Oracle 服务		445
22.1	回顾 Oracle 服务体系	445
22.2	ACS 服务概述	446
22.3	ACS = 医院 + 保险公司	448
22.4	浓墨重彩 SSC 服务	451
22.5	问题、需求和服务解决方案	464
22.6	本章参考资料及进一步读物	467
第 23 章 Oracle 高级客户服务中那些事		468
23.1	如何主动开展 ACS 服务的故事	468
23.2	没有 ACS 服务的窘境	470

23.3 ACS 售前案例：对症下药	472
23.4 ACS 售前案例：机会总有的	474
23.5 ACS 与第三方公司的区别和优势在哪里？	476
23.6 原厂的尴尬和原厂的优势	480
23.7 感悟 ACS 服务	481
23.8 也谈服务	483
23.9 本章参考资料及进一步读物	484
结束语	485

1. 设计、优化篇 中的那些事

数据库设计、应用开发和优化工作是一项对 IT 系统质量影响最大的基础性工作。本人见过太多由于数据库设计不规范、不严谨而导致的严重问题，但此时已是木已成舟，我也只能是望洋兴叹了。我也曾多次扬言要写本书，把数据库设计和应用开发这些基础性内容放在最前面，让年轻的设计开发人员少犯一些低级错误。

于是，新书的第一篇的设计、优化篇就是有关逻辑设计、物理设计、海量数据库设计，以及百谈不厌的性能优化等章节了。

增加一个逻辑设计的话题吧，特别是一些最佳实践经验和常见的设计错误。"可见，设计人员也深深体会到了规范设计的重要性。"

于是，本书就以数据库逻辑设计开篇了。本章首先以若干年前一个典型案例开始，进而介绍关于数据库设计的规范化理论，以及逻辑设计的工具 Oracle Data Modeler，然后介绍非规范化设计的若干最佳实践，最后针对逻辑设计发表若干结语。

1.1 还是从案例开始

十多年前，在一个网站从事数据库 DBA 工作期间，有一天一位开发人员拿来请我和他分析他编写的一个简单 SQL 语句的性能问题，以下是具体情况。

1. 语句和执行计划

该语句再简单不过，即查询"喜欢读书的所有用户"。

SQL 语句如下：

```
SELECT * FROM USER WHERE  
  (SELECT COUNT(*) FROM BOOK WHERE  
    USER_ID = USER_ID AND  
    IS_READ = 'Y') > 0
```

图 1-1

可见对 is_read 表过滤效果很糟糕，当年该网站日访问量已经达到十多万，查询性能可想而知。

2. 问题原因分析

事实上，该开发人员已经在 Oracle9i 中建立了该表的索引，为什么 Oracle 不采用这个索引来加速查询效率呢？答案就在 SQL 语句本身，即该语句的后面一个"%"号，因为 B*树索引是个排序的东西，如果该语句后面没有包含"%"号，那么它的所有内容，即 Oracle 就不需要扫描，没必要使用这个索引了。

该 SQL 语句的完整内容如下：

第 1 章 逻辑设计中的那些事

数据库逻辑设计是从事数据库应用设计、开发、运行维护等各方面工作的一个重要的基础性工作。根据不同的业务和应用需求，确定并遵循数据库逻辑设计原则，例如按照第三范式开展逻辑设计，不仅能满足减少数据冗余、保证数据一致性和完整性、易扩展性和伸缩性等需求，而且也是保障系统高性能的一个重要基础。

有一次，本人与一家公司进行为期 3 天的数据库性能优化培训。为达到良好的培训效果，事前与该公司几位技术骨干就培训大纲进行讨论，其中一位客户就提到：“罗老师，增加一个逻辑设计的专题吧，特别是一些最佳实践经验和常见的设计错误。”可见，该客户也深深体会到了逻辑设计的重要性。

于是，本书就以数据库逻辑设计开篇了。本章首先以若干年前一个典型案例开始，进而介绍关于数据库设计的规范化理论，以及逻辑设计的工具 Oracle Data Modeler，然后介绍非规范化设计的若干最佳实践，最后针对逻辑设计发表若干感悟。

1.1 还是从案例开始

十多年前，在一个网站从事专职 DBA 工作的某一天，一位开发人员过来请我帮忙分析她编写的一个简单 SQL 语句的性能问题。以下是具体情况：

1. 语句和执行计划

语句再简单不过，即查询“喜欢文学的所有客户”。

```
Select id,name,hobby from customer where hobby like '%文学%'
```

执行计划如图 1-1 所示。

Description	Object owner	Object name
SELECT STATEMENT, GOAL = CHOOSE		
TABLE ACCESS FULL	HR	CUSTOMER

图 1-1

可见对 customer 表进行全表扫描。当年该网站客户数就已经达到千万级，查询性能可想而知。

2. 问题原因分析

事实上，该开发人员已经在 HOBBY 字段上建立了索引，为什么 Oracle 不采用这个索引来提高查询效率呢？原因就在于 like "%文学%"操作中的前面一个“%”号。因为 B*树索引是个排序的东西，如果你需要匹配查询包含“文学”二字的所有内容，那 Oracle 就无序可循，没必要使用这个索引了。

进一步观察 customer 表的内容如下：

```
SQL> Select id,name,hobby from customer;
```

ID	NAME	HOBBY
1	张三	文学, 艺术, 电影
2	李四	运动, 文学, 电影
3	王五	登山, 旅游, 文学, 艺术
...		...

哦，原来她是这么设计表的，把客户所有的爱好都塞在一个字段 **HOBBY** 中了，所以查询条件只能写成 like '%文学%'。问题根本原因是该表的逻辑设计不符合规范化设计理论，从而导致了全表扫描。

下面先介绍数据库规范化设计原理，然后再回头介绍该简单案例在数据库设计 and 应用方面的优化。

1.2 什么叫规范化设计？

关于数据库规范化设计定义

以下是维基百科中关于数据库规范化设计的经典定义：

“数据库规范化，又称数据库或资料库正规化、标准化，是数据库设计中的一系列原理和技术，以减少数据库中数据冗余，增进数据的一致性。关系模型的发明者埃德加·科德最早提出这一概念，并于 19 世纪 70 年代初定义了第一范式、第二范式和第三范式的概念，还与 Raymond F. Boyce 于 1974 年共同定义了 BC 范式。”

以下是 Oracle 公司对第一范式、第二范式和第三范式的经典定义：

First normal form (1NF): All attributes must be single-valued.

Second normal form (2NF): An attribute must be dependent upon its entity's unique identifier.

Third normal form (3NF): No non-UID attribute can be dependent on another non-UID attribute.

老罗略加给出的更通俗的解释如下。

- 第一范式（1NF）。所有属性必须是单值的，或者说是原子化（Primitive）的、不可分割的。
- 第二范式（2NF）。所有属性必须依赖于该实体的唯一标识属性，也就是说每个实体应该有唯一标识属性。
- 第三范式（3NF）。没有一个非唯一标识属性依赖于另一个非唯一标识属性，也就是说实体中不存在传递的依赖关系。

不符合规范化设计的若干案例

对抽象概念加深理解的最好办法就是举例。

1. 违反第一范式的例子

显然，上述案例就违反了第一范式。因为 **HOBBY** 字段不是单值的，包含了文学、艺