
Oracle 设计

Dave Ensor & Ian Stevenson 著

林琪 王宇 译

O'REILLY®

Beijing Cambridge Farnham Köln Paris Sebastopol Taipei Tokyo

O'Reilly & Associates, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目 (CIP) 数据

Oracle 设计 / (英) 恩索尔 (Ensor, D.), 史蒂文森 (Stevenson, I.) 著 ; 林琪 , 王宇译 . - 北京 : 中国电力出版社 , 2002.9

书名原文 : Oracle Design

ISBN 7-5083-1112-4

I. O... II. 恩... 史... 林... 王... III. 关系数据库 - 数据库管理系统 , Oracle - 设计 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2002) 第 053058 号

北京市版权局著作权合同登记

图字 : 01-2002-1203 号

©1997 by O'Reilly & Associates, Inc.

Simplified Chinese Edition, jointly published by O'Reilly & Associates, Inc. and China Electric Power Press, 2002. Authorized translation of the English edition, 1997 O'Reilly & Associates, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly & Associates, Inc. 出版 1997。

简体中文版由中国电力出版社出版 2002。英文原版的翻译得到 O'Reilly & Associates, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者 —— O'Reilly & Associates, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

书 名 / Oracle 设计

书 号 / ISBN 7-5083-1112-4

责任编辑 / 宋宏

封面设计 / Edie Freedman , 张健

出版发行 / 中国电力出版社 (www.infopower.com.cn)

地 址 / 北京三里河路 6 号 (邮政编码 100044)

经 销 / 全国新华书店

印 刷 / 北京市地矿印刷厂

开 本 / 787 毫米 × 1092 毫米 16 开本 33.5 印张 493 千字

版 次 / 2002 年 9 月第一版 2002 年 9 月第一次印刷

印 数 / 0001-5000 册

定 价 / 59.00 元 (册)

作者简介

Dave Ensor 是BMC软件公司的Worldwide Solutions, PATROL R&D部门的管理人员,他的工作是在顾客使用BMC的PATROL产品和Oracle服务器时提供帮助,并将他在这一领域所做的工作结果反馈到产品规划中。他在IT领域已经有30多年的丰富经验,自1987年以来即开始研究有关Oracle的设计和性能问题。他曾在英国的Oracle全球性能研究小组担任多年的领导工作,并为顾客以及性能需求苛刻的项目提供咨询支持。Dave作为一个性能管理和设计的发言人享有盛誉,他不仅在用户大会上提交论文,还为一日性的研讨会撰写和发表论文。他住在英国的伦敦郊区,但常年都奔波于各个用户点和会议之间。在闲遐之余,他也去旅游,不过与平常有所不同,这时他所带的不是笔记本电脑,而是携妻子同行。可以通过 dave_ensor@compuserve.com 与他联系。

Ian Stevenson 是一个专注于数据库设计和开发的自由顾问。他最初从层次结构开始,到目前已经在数据技术领域里做了19年的研究工作。他为Oracle英国公司的售后支持和人力资源开发项目工作过两年。在这里他与Dave Ensor建立了友谊。Ian曾在南安普敦大学以一等生获得了数学学位,而且是英国计算机协会的会员。他与Brenda结婚并有两个孩子,Todd和Tara。他还是南安普敦足球俱乐部的狂热支持者。可以通过 ian@westmail.demon.co.uk 与他联系。

封面介绍

本书封面上的小昆虫是一只蜻蜓 (dragonfly) 和一只蜻蛉 (damselfly)。这两只昆虫均为蜻蜓目昆虫。蜻蜓和蜻蛉非常相似，都属食肉昆虫，幼年时为水栖动物，而变为成虫后则是非常不错的飞行能手。它们的飞行时速能够达到 35 英里，而且还能完成很多空中特技。这两种昆虫之间有一个主要的区别，即在休息时停放翅膀的方式有所不同。蜻蜓在休息时会将翅膀张开，而蜻蛉则会把翅膀收起来放在背上。

小蜻蜓有一种独一无二的能力 —— “ 喷气式推进 ”。在它腹部后面可以看到有腮，利用它们的划水，蜻蜓可以迅速向前从而推动自己在水面上疾驶，在出现紧急情况时它就会藉此技术逃生。

成年的蜻蜓和蜻蛉在飞行时捕食猎物。它们飞行时长长的毛腿蜷缩于身体前面，就像一个篮子，而其他昆虫就装在这里。蜻蜓是一种很受人类喜爱的昆虫，这源于它的美丽和优雅，此外还因为它所捕食的主要是蚊子和苍蝇。

O'Reilly & Associates 公司介绍

为了满足读者对网络和软件技术知识的迫切需求,世界著名计算机图书出版机构 O'Reilly & Associates 公司授权中国电力出版社,翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly & Associates 公司是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司,同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为二十世纪最重要的 50 本书之一)到 GNN(最早的 Internet 门户和商业网站),再到 WebSite(第一个桌面 PC 的 Web 服务器软件),O'Reilly & Associates 一直处于 Internet 发展的最前沿。

许多书店的反馈表明,O'Reilly & Associates 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比,O'Reilly & Associates 公司具有深厚的计算机专业背景,这使得 O'Reilly & Associates 形成了一个非常不同于其他出版商的出版方针。O'Reilly & Associates 所有的编辑人员以前都是程序员,或者是顶尖级的技术专家。O'Reilly & Associates 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家,而现在编写著作,O'Reilly & Associates 依靠他们及时地推出图书。因为 O'Reilly & Associates 紧密地与计算机业界联系着,所以 O'Reilly & Associates 知道市场上真正需要什么图书。

前言

为什么要写这本书呢？因为我们强烈地感到大家低估了 Oracle 设计的重要性，况且目前也迫切需要一本深入探讨这方面内容的书。除了这些想法，写这本书时我们还有其他一些考虑。

Ian 在参与了 Oracle6 和 7 的几个项目的设计阶段之后，萌生了写这本书的念头。他注意到许多同样的问题和困难经常会一次次地出现。尽管这些问题很类似，但是其解决方法却会因为项目的不同而有所差异。显然，对于具体的解决方法不太可能简单地断言孰是孰非。事实上，就特定的项目而言，每个解决方法都可能是上策。所以他力图建立一个信息库，借此将做出特定设计决策的理由记录下来。这样，如果将来他（或者其他人）碰到类似的问题时，就能够从信息库中获得帮助。不难相信，这本书将成为 Oracle 设计领域的一个绝好媒介，大家可以由此在 Oracle 社区共享一些难得的设计经验。

Dave 多年以来一直任 Oracle 全球性能研究组（Oracle's Worldwide Performance Studies Group）的负责人。在实践中，他不只一次地发现 Oracle 性能问题总是源自相似的根本原因。它们往往并非是由于构造技术上存在着缺陷，也不是由 Oracle 性能调整造成的。真正的原因通常是项目小组的设计上存在着错误。不过也不能否认“传统”因素确实也会对性能造成影响。但是很明显，正是由于设计决策有问题，许多项目根本无法提供较好的性能。基于这样的背景，Dave 在一个研讨会上为所有开发人员准备了一篇题为“Designing for Oracle7”的发言稿，尽管它很长，但是仍

不足以提供完全可靠且面面俱到的设计方案。和 Ian 一样, Dave 在撰写发言稿的时候, 也发现如果要评判设计方案, 就必须根据项目的实际需求和其所处的具体技术环境来决定。

假如没有固定的规则可依, 又为什么要写一本这方面的书呢? 还有为什么目前市面上关于设计方面的书这么少呢? 首先回答第二个问题, 我们认为, 由于无法给问题提供直截了当的答案 (即使是最简单的问题, 比如“每个表都必须要有个主键吗?”), 便使得许多作者和出版商退缩了, 因为他们认为读者只愿意接受对所有问题都能给予绝对答案的技术资料。其实事实并非如此。

我们写这本书有两个绝好的理由:

1. 在参与的所有项目中, 对那些解决起来希望很渺茫的问题, 我们都曾付出大量时间探索其解决方案, 但是无论我们如何千方百计地努力, 也永远无法解决这些问题。如果我们知道这些, 就可以在一开始时避免这些问题, 以节省宝贵的时间。
2. 此外我们发现, 随着经验的逐步积累, 即使能从以前的项目设计技术中汲取一些影响系统性能的粗略方法, 也是难能可贵的。那些经验丰富的设计人员, 有时也难免会忽略某个特殊问题的解决方法。因此, 倘若有些其他类型的、即使并非是绝对的参考资料, 对于探讨不同的设计方案也可能会是一种价值不菲的工具。

写此书的初衷就是要给大家 (那些和我们一样的人) 提供一个日常使用的工具。我们非常乐意与你分享以前的经验, 虽然有时很困难, 但它们确实是有益的经验。

关系数据库的设计问题非常棘手。即使看起来相对简单的业务需求, 说不定也会引出复杂的设计问题。当你仔细斟酌各种满足业务需求的设计方案时, 常常转了一圈又会回到最初的选择, 此时你可能已经记不清这个选择有何缺陷。但却切实感到问题没有取得任何进展, 于是有点儿灰心丧气。在这里我们要告诉你, 即使有本书作为参考, 上述情况也在所难免。从事设计工作的人都曾有过同样的经历, 你一定要相信自己最终能够冲破黑暗, 走向光明。

本书的目的是帮助你工作得更潇洒、更轻松。若能把这些经验真正应用到你的设计中, 必将收获丰硕的果实。

为什么针对 Oracle ？

出版商所问的第一个问题就是：“为什么是一本关于 Oracle 设计的手册？为什么不针对所有的关系数据库？Oracle 有什么不同吗？”这的确是一个很重要的问题，我们也很愿意回答。

包含计算机应用程序在内的所有设计，都要充分利用所提供的技术和资源。当选择某个设计方案时，必须充分考虑到 Oracle7 所具有的许多特殊的、惟一的技术特点。本书用很大的篇幅来讨论那些特点。在多数情况下，我们建议你应该首先在自己的环境中进行测试，然后再选择某个特殊的设计方案。

我们不敢妄断这本书会使你一夜之间成为一名成功的 Oracle 设计人员。因为没有什么可以代替设计经验。那些出色的设计人员总是在不断地学习进取。在本书中，我们只给出了一个更加结构化、更容易控制的设计环境框架。针对主要的设计决策，每一章都给出了相应的替代方案，而且还强调了该设计的优点和缺陷。当然，最终的决定权在于你，我们只希望你能够三思而后行。

目前 Oracle 已经是一个成熟的产品。从最初发布这个产品开始，Oracle 公司就开始从应用程序（或用户界面）中提取业务规则的智能因素，将其纳入数据库（或服务器）中。这一变革意味着在同样的情况下，采用 Oracle7 的项目比采用 Oracle6 或 3GL（第三代程序语言）开发的项目需要花更多的时间进行数据库设计。如果是一个优秀的设计方案，创建应用程序的任务会更加简单。一旦应用程序的建立阶段已经开始，就很难再修改设计阶段做出的任何错误决定或者疏漏，而且会为此付出高昂的代价。

一个差的设计方案会令开发周期在长时间内困难重重，最终系统的质量也很低劣。事实上，如果方案设计得太差，就很难形成一个最终的系统，即使能够成系统，也是一个相当拙劣的产品。设计一个系统时，总要考虑它的性能和可用性。如果你被告知响应时间无关紧要或者“以后再修正响应时间”，那我们建议此时一定要进行争辩。在产品系统中，修正一个设计缺陷或错误所付出的代价非常大。应当抱着“极其苛刻”的态度解决性能设计问题。在我们的印象里，即便是一个低端系统，其修正也无法得到准确无误的保证。

还要记住的是，最终用户对任何应用程序的感觉，通常决定于他们第一周内使用该系统的印象。新的计算机系统的性能、用户应答时间、易用性和健壮性，直接决定了用户最终能否接受这个系统。

总而言之，我们在任何设计过程中都要达到以下目标，这些内容也是本书力求帮你做到的。贯穿系统运行的整个过程中，要求的性能指标包括：

功能性和适应性

吞吐量

响应时间

可用性

易用性

安全性

本书的组织结构

本书所述内容共分五部分。虽然我们希望你能通篇逐字地阅读，但我们也知道，有些读者不想从前往后一字不差地翻一遍；即使要看大部分，也不愿意按照顺序进行。为了满足大家的需要，我们尽量使每章内容相对独立，减少交叉引用的次数。那些经验丰富的设计人员、具有良好的项目生命周期管理经验的管理人员，可以跳过前面几章（主要是第一章和第二章）。使用过 Oracle 最新版本的人也可以跳过第二章。许多描述特殊技术的章节，也只是在具体设计时才需要细细研读。分布式数据库（第十二章）和数据仓库（第十三章）专门介绍了这些特殊的技术。

第一部分：开始 Oracle 设计

第一章，“简介”，介绍项目开发的各个阶段，特别是策略制定阶段、分析阶段、设计阶段和建立阶段；本章采用了一个真实项目的简化例子，来解释各个项目阶段及其需要交付的内容。

第二章，“Oracle 设计的重要性”，针对 Oracle，描述了特殊的设计问题，并逐条说明 Oracle7 的重要功能；对 Oracle8 及以上版本做了一个简单介绍。

第三章，“数据建模”，详细考查数据建模；定义典型的关系数据库术语，如实体、关系和第三范式(3NF)；逐条说明项目分析人员应当向设计人员提交哪些内容，才能将概念数据模型转化为逻辑数据模型。

第二部分：数据库设计

第四章，“决定何时反向规范化”，为提高性能，针对数据库数据，介绍一些特殊的反向规范化技术。

第五章，“选择数据类型和 null”，介绍各种 Oracle 数据类型，描述如何为数据库中的对象选择最恰当的数据类型，并探讨那些最令人头痛的问题，包括什么是 null，如何对待 null。

第六章，“选择键和索引”，讨论如何为特定的数据库选择最合适的键。

第七章，“时间数据的处理”，探讨一个与 Oracle 以及其他关系数据库都有关系的特殊问题——关系数据库没有充分支持时间序列（时间）数据。本章提供了大量特定于 Oracle 的技术，可以使用它们来克服时间数据所造成的限制。

第八章，“加载和卸载数据”，研究了从外部数据源生成 Oracle 数据库的各种方法。还讨论了如何从 Oracle7 中提取数据。

第九章，“如何放置和存储对象”，了解一些重要的数据库物理设计方法，比如数据库大小和文件放置。

第十章，“保护数据”，涉及备份、存档、审核和安全等主题。

第三部分：特殊体系结构的设计

第十一章，“客户/服务器的设计”，采用 Oracle 设计方法设计客户/服务器模型。分析了各种优化性能和可用性的分布式处理技术。

第十二章，“分布式数据库的设计”，进一步探讨数据分布模型，介绍为什么以及如何使用分布式数据库。分析了 Oracle 提供的各种可用选项，以及如何在不同的方案中使用它们。

第十三章，“数据仓库的设计”，掌握如何建立和维护数据仓库。如何进行空间建模，如何将数据保存到数据仓库中，以及如何再将它们提取出来。

第十四章,“并行处理的设计”,描述并行处理的概念,Oracle的Parallel Query Option(并行查询选项)和Oracle Parallel Server(Oracle并行服务器)在实际应用程序中的应用,探讨磁盘分段处理和RAID等技术。

第四部分:代码模块的设计

第十五章,“代码设计介绍”,描述代码模块设计的基本概念。

第十六章,“如何划分处理逻辑”,讲述一种划分应用程序逻辑的新方法。

第十七章,“度量、原型和规范”,描述代码设计的正规方法;特别要集中探讨如何确保模块满足需求。

第十八章,“锁定”,包含一些有用信息,能够帮助你最小化应用程序中的争议。

第十九章,“选择工具集”,比较各种支持Oracle后端的前端产品的优缺点。

第二十章,“屏幕、报表和批处理程序等的设计”,解决有关屏幕、报表、批处理程序、错误处理、导航和在线帮助等特殊的设计问题。

第五部分:附录

附录一,“第三方提供的开发包”,它比较了购买一个打包的解决方案与完全从头开发应用程序的各自优点。

附录二,“设计的窍门”,集中讨论两个特殊的设计技术。我们提出一个方法来避免恼人的Oracle触发器“变异表”问题;简要介绍了扩展的SQL。

本书的读者

本书认为你已经具备Oracle方面的知识,但是对于那些需要转到Oracle,并具有其他关系数据库管理系统(RDBMS)使用经验的读者,本书也会给予他们很多帮助。无论你目前具备的是开发经验还是数据库管理经验都没关系,应该先熟悉本书中提到的相关术语和概念。我们还假设你具有项目开发生命期的基本知识。

本书怎样帮你成为一名设计人员呢?它并没有回答所有方面的问题,只是提供一个设计框架和一张常用的设计问题检查表。希望你在进行某项特殊设计任务时,能求

助于本书。它会帮助你澄清大量的、无法考虑周到的问题；我们还希望本书能指导你解决设计问题，激发更深入的思考和更热烈的讨论，为你解决重要问题提供参考，让你的设计更加完美。

针对哪种 Oracle ？

本书涉及的问题适用于 Oracle 的所有发行版本。有些极少数的例外情况，我们将明确指出所适用的版本。希望 Oracle 的新版本能为本书中所讨论的一些问题提供解决方法。无论如何，我们相信书中所提供的大部分内容会继续适用于后继的 Oracle 版本。

排版约定

下面是本书英文所使用的约定：

斜体字 (*Italic*)

用来表示文件、变量和函数名。有时也用来强调或突出第一次提及的关键术语。

黑体字 (**Bold**)

用来表示标题。

等宽字符 (Constant width)

用于代码示例。

大写

在代码示例中，表示 Oracle 关键字。

小写

在代码示例中，表示用户定义的条目，如变量和参数。

标点符号

在代码示例中，实际输入的与显示的相同。

缩排

在代码示例中，帮助显示代码的结构，但不是要求的。

-

在代码示例和相关讨论中，“点”将对象名从组件名中区分开来，起限定的作用。

固定的 < 可变的 >

在大量示例和相关讨论中，表示一部分字符串或数据项是固定的，另一部分是可变的。

实体模型

采用信息工程方法学（IEM）的约定进行绘制。

关于示例

我们原打算为本书附带一张磁盘或光盘，最后放弃了，因为本书所举的示例代码都相当短而且是自包含的。但是，如果你想获得本书有关程序的电子阅读版本，可以访问 O'Reilly 的 Web 站点，通过匿名 FTP 下载实例。

建议与评论

本书的内容都经过测试，尽管我们做了最大的努力，但错误和疏忽仍然是在所难免的。如果你发现有什么错误，或者是对将来的版本有什么建议，请通过下面的地址告诉我们：

美国：

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472

中国：

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室
奥莱理软件（北京）有限公司

询问技术问题或对本书的评论，请发电子邮件到：

info@mail.oreilly.com.cn

最后，你可以在 WWW 上找到我们：

http://www.oreilly.com

http://www.oreilly.com.cn

致谢

有许多人为本书做出了贡献。首先，我们要感谢 O'Reilly 公司的鼓励，感谢他们提出了专业的、友好的建议。特别要对我们的编辑 Debby Russell 表示谢意。Mike Sierra 是 O'Reilly 公司的工具专家，他为我们解决了大量的转换和制作问题；John Files 和 David Futato 是本书的责任编辑；Danny Marcus 是校对人员；Chris Reilley 配制了本书所有的插图；Edie Freedman 设计了封面；Nancy Priest 创建了内部格式；Seth Maislin 负责建立本书的索引。

还非常感谢 Mark Daynes、Barry Goodsell、Graham Wood、Martin Cantwell、Steven Feuerstein 和 Mark Gurry 对本书提出的建设性意见和评论。

我们尊敬的同事们也为本书做了许多无私的奉献。书中的很多经验，源自与他们友好的交谈，而且常常是在咖啡售货机旁。Ian 还要特别感谢 Andrew Mackley 对数据仓库一章的贡献。

Ian 要感谢他的妻子 Brenda 和孩子 Todd 以及 Tara，感谢他们的支持与鼓励，特别是对他长时间在办公室加班所给予的谅解。

Dave 要感谢 Ian，是 Ian 最初提出写本书的想法并邀请他参与写作。他还要感谢他的妻子 Mefus 对本书的支持（包括过去 27 年对他的一贯支持）。

目录

前言	1
----------	---

第一部分 开始 Oracle 设计

第一章 简介	13
--------------	----

什么是设计?	13
一个案例研究	14
寻找替代的方法和途径	25
制定设计阶段的计划	29
设计的任务	30

第二章 Oracle 设计的重要性	46
-------------------------	----

特殊体系结构的设计	46
性能设计	51
其他设计考虑	53
Oracle 的版本演变及其对设计的影响	55
小结	72

第三章 数据建模	73
模型的类型	74
什么是数据建模?	76
实体关系图	78
对概念信息模型的质量保证	82
实体生命历史和数据流图	106
数据驱动的设计和元模型	110
第二部分 数据库设计	
第四章 决定何时反向规范化	115
反向规范化：是什么，为什么，什么时候?	115
实现反向规范化的方法	119
其他类型的反向规范化	121
第五章 选择数据类型和 null	127
可用的数据类型	128
数字数据	129
日期和时间(时间)数据	132
字符串数据	134
非结构化的数据和 BLOB	137
其他数据类型	138
null 值	140
第六章 选择键和索引	146
主键	147
其他键	156
索引的概述	158
索引类型和建立索引的技术	162

第七章 时间数据的处理	179
时间数据的问题	179
时间数据库的设计问题	183
时间数据：小结	204
第八章 加载和卸载数据	205
与外部系统的交互	205
数据兼容性问题	208
数据迁移步骤	215
数据转换	217
文件格式	221
排序、恢复和提交的频率	223
使用 SQL*Loader	223
从 Oracle 到 Oracle：一个特殊的案例	226
引出数据	227
第九章 如何放置和存储对象	230
指定对象的放置位置	230
规划大小	233
指定存储参数	235
创建脚本	238
实现计划	238
第十章 保护数据	240
存档	241
审核	248
安全	259
备份	270