

数据库 索引设计与优化

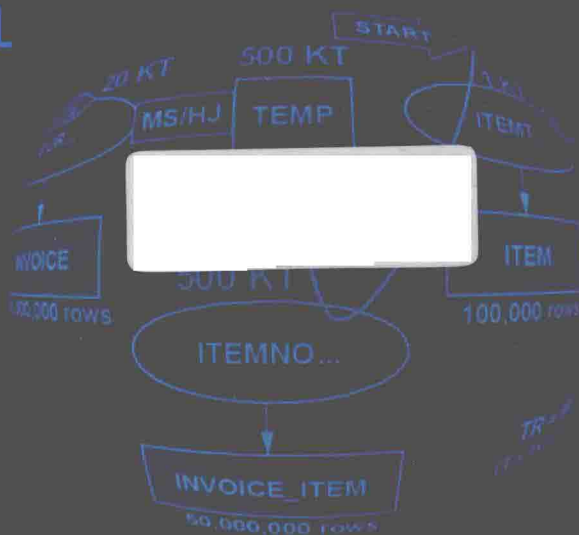
Relational Database
Index Design and the Optimizers

[美] Tapio Lahdenmäki 著
Michael Leach

曹怡倩 赵建伟 译

MySQL

Oracle



SQL Server

et.al.

数据库索引设计与优化

Relational Database Index Design and the Optimizers

[美] Tapio Lahdenmäki 著
Michael Leach

曹怡倩 赵建伟 译



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书提供了一种简单、高效、通用的关系型数据库索引设计方法。作者通过系统的讲解及大量的案例清晰地阐释了关系型数据库的访问路径选择原理，以及表和索引的扫描方式，详尽地讲解了如何快速地估算 SQL 运行的 CPU 时间及执行时间，帮助读者从原理上理解 SQL、表及索引结构、访问方式等对关系型数据库造成的影响，并能够运用量化的方法进行判断和优化，指导关系型数据库的索引设计。

本书适用于已经具备了 SQL 这一关系型语言相关知识，希望通过理解 SQL 性能相关的内容，或者希望通过了解如何有效地设计表和索引而从中获益的人员。另外，本书也同样适用于希望对新硬件的引入所可能带来的变化做出更好判断的资深人士。

Relational Database Index Design and the Optimizers, 9780471719991, Tapio Lahdenmäki, Michael Leach.
Copyright © 2005 by John Wiley & Sons, Inc.

All rights reserved. This translation published under license.

No part of this book may be reproduced in any form without the written permission of John Wiley & Sons, Inc. Copies of this book sold without a Wiley sticker on the back cover are unauthorized and illegal.

本书简体中文版专有翻译出版版权由美国 John Wiley & Sons, Inc. 公司授予电子工业出版社。未经许可，不得以任何手段和形式复制或抄袭本书内容。本书封底贴有 John Wiley & Sons, Inc. 防伪标签，无标签者不得销售。

版权贸易合同登记号 图字：01-2014-4721

图书在版编目 (CIP) 数据

数据库索引设计与优化 / (美) 拉赫登迈奇, (美) 利奇 (Leach, M.) 著; 曹怡倩, 赵建伟译. —北京: 电子工业出版社, 2015.6

书名原文: Relational Database Index Design and the Optimizers

ISBN 978-7-121-26054-4

I. ①数… II. ①拉… ②利… ③曹… ④赵… III. ①数据库系统—研究 IV. ①TP311.13

中国版本图书馆 CIP 数据核字 (2015) 第 098122 号

策划编辑: 张春雨

责任编辑: 徐津平

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

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: 720×1000 1/16

印张: 20

字数: 380 千字

版 次: 2015 年 6 月第 1 版

印 次: 2015 年 6 月第 1 次印刷

定 价: 79.00 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 zits@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线: (010) 88258888。

推荐序一

2015年，我很欣喜地看到这本书终于被真正的技术粉们翻译成了中文。

这本写作于2005年的著作，是少数能够穿越十年仍然历久弥新的经典著作之一。译者曹怡倩和赵建伟是支付宝的资深数据库专家，也是童家旺的得意门生，他们的经验和理解为本书增色良多。我在学习过程中也曾经从本书的英文版中获益，所以当旺旺委托我写一段文字时，我欣然受命，但是其实他才是这本书的知音，我曾经在他的多次公开演讲中听到他郑重推荐此书，并引用书中的方法达成实践。今天，两位作者的真知灼见跨越时间和空间，在中文世界里找到喜爱的读者们，这是作者们的欣慰，更是读者们的幸运。

在RDBMS的世界里，直到今天，索引仍然是无与伦比的重要环节——走过千山万水，优化仍然要回归到和索引纠缠不清的本质上来。作者在第5章提出“前瞻性的索引设计”，其中的观点之一是“设计索引，使所有查询语句都运行得足够快”，这让我想到2011年我们翻译的韩国著作《海量数据库解决方案》一书，李华植先生在书中同样提出了“战略性索引设计方案”，实在是具有异曲同工之妙，大家都着眼于如何使用更合理的索引去满足尽可能多的SQL访问需求。而为了解决开发环节中的不良SQL性能问题，云和恩墨甚至开发了一款专门的软件用于SQL审核，其中的多个维度指标仍然是考量索引的。

大道至简，从最基本的起点出发，才有可能从最根本解决我们面对的复杂终极问题。当然，今天我们对于索引的关注已经不仅仅是有无，而已扩展到索引的各种访问方式，索引全扫描、索引跳跃式扫描等，细微之处更见功夫。

技术的另外一大迷人之处在于其无穷的细节变化和无限的可能。针对作者在第1章中提到的种种误区，我们在阅读时都可以细细地思量并引发探讨。比如作者提到的“误区2 单表的索引数不要超过6个”，在很长一段时间内，我对此并没有深刻的认识和经验，直到2010年在一个客户的环境中遇到了非常棘手的问题，我的确史无前例地分析了索引个数对于Oracle SQL解析性能的影响。要知道，在硬解析的过程中，CBO需要对所有可能发挥作用的索引进行非常详尽的成本计算和评估，这在Oracle中涉及了ROW

CACHE 的算法管理，在串行 Latch 的守护下，如果索引的数量增多，无疑会降低解析的速度。我还意外地发现在 `dc_object_ids` 的使用上，效率不佳——你可能也注意到了，在 Oracle 11g 中，这个资源消失不见了。就是这么好玩，你总会发现例外。而且这一切发生在 SQL 开始执行之前，我们更多时候关注的是索引好不好、快不快，可能我们忽略了在 SQL 开始执行之前的解析环节中就可能存在无法规避的问题。

在 2008 年 Oracle 推出的 Exadata 一体机上，Oracle 更加喜欢在全扫描的过程中通过智能扫描（Smart Scan）进行数据卸载，而存储索引更是一个让人目瞪口呆的设计，你可能觉得这个基于内存的设计根本就不是索引！再然后，在 2014 年 Oracle 推出了 IMO 之后，Larry 居然宣布，有了这个特性赶紧去干掉一堆索引吧，这会让 OLTP 更流畅。

各种好玩啊！技术世界就是如此变化多端，曲曲折折！

你也必须觉得好玩，才能在技术的世界里找到快乐！

在这本书中，你一定能够找到点燃你技术热情的火花！

——盖国强（Oracle ACE 总监，云和恩墨创始人）

2015 年 4 月于北京

推荐序二

经过多年的发展，关系型数据库已经成为一个企业 IT 架构的核心，诞生了包含 Oracle、MySQL、DB2、Microsoft SQL Server、Sybase、PostgreSQL 这样的数据库。上述这些传统的数据库都强烈地依赖于索引的设计与规划。不夸张地说，一个应用是否合格的关键就在于数据库索引的设计与优化。

在工程领域，很多数据库工程师们都积累了对于索引设计与优化的经验，可惜很少有书籍系统地介绍这部分内容，这导致网上存在很多错误的观念。可喜的是本书详细介绍了一种简单、高效的数据库索引设计方法，能让读者对数据库的索引设计快速进阶。此外，本书通过大量的实例分析了关系型数据库的访问路径选择原理，对实践有着实际的指导意义。最为关键的是，这些原理与知识对于所有关系型数据库都是通用的，甚至对于目前出现的 NoSQL 数据库也是具有指导意义的。

本书对于想要理解数据库索引设计方法，以及数据库优化器访问路径选择原理的读者都具有现实的指导意义。希望将本书推荐给所有从事数据库相关工作的人员，它的内容不局限于具体的数据库类型，是一本修炼内功的好书，看完之后定能解决在工程中遇到的各种索引设计与优化问题。

姜承尧

网易杭州研究院数据库技术组负责人

推荐序三

一直以来，我们都将数据库行业鼻祖萨师焯老师主编的《数据库系统概论（第三版）》作为葵花宝典，每年都要对其核心章节读上 1~2 遍，也曾读过他人翻译的《数据库系统面向应用的方法》。总体而言这两本书在关系型数据库的基础知识和技术原理等方面基本一致，也有互补的技术内容，但关于索引设计与优化方面的内容尚有不足，国内外也未发现这方面较好的书籍。好友童家旺找我为其徒弟曹怡倩和赵建伟翻译的书籍作序，我看到书名《数据库索引设计与优化》时便欣然答应了，因为此书能填补国内的空白，为众多技术人员带去福音。

在阅读书籍全文内容的过程中，我深刻地体会到国际友人写书风格非常值得学习，全篇使用示例与公式一步一步分析、计算和验证，不仅得出经得起推敲的方法论，而且深入浅出地讲述了相关的技术原理，既体现出作者的严谨风格，又便于读者学习、记忆和理解。阅读过程中能发现本书的专业术语翻译得非常精湛，对原理、算法、方法论、技术常识等内容的描述，更让人读起来有读原著作品的口感，这体现了译者深厚的数据库理论功底，以及对本书内容的深刻理解，甚至已有超越原著的水准。

虽然近二十年中数据库理论知识未有任何质的变化，但计算机软硬件技术的飞速发展促进了多种数据库管理系统实现算法的革新。随着软硬件技术算法实现的变革，早期关于索引创建与优化的部分准则已不适宜当下，故本书籍从索引相关的基础知识和早期建立准则误区出发，介绍了硬盘处理器内存与索引的关系、其中的技术原理和索引原理，以及不同数据库产品之间的异同、SQL 处理过程与索引的关联性、如何创建理想和前瞻性的索引、索引设计需要关注和平衡的点、不同应用场景下的索引设计技巧与方法论、如何规避索引的弊端、数据库管理系统的索引限制、如何让优化器运行更佳，总结了当下设计优秀索引的准则。

本书按照由浅入深的方式安排内容，清楚阐释了原著作者想要解释的两个问题：如何量化索引和优化器的成本，如何从 CPU 和执行时间的角度量化运行过程。

书中使用了 Oracle、DB2 和 SQL Server 这三种数据库的大量案例，介绍了它们的异同与各自的特性，绝大部分内容也同样适合于 MySQL 和 PostgreSQL 数据库产品。个人认为本书同萨老师主编的书籍一样可奉为葵花宝典，是值得 DBA 们人人收藏的一本书，也适合作为云计算平台研发自动化 SQL 审核与索引创建组件研发工程师的工具书。

金官丁

书于 2015 年 4 月 20 日

推荐序四

本书原著是数据库领域的重要理论大作，几年前童家旺先生就推荐过此书，在《高性能 MySQL》一书中也推荐过它，据说他还要求每位徒弟都熟读此书，可见其重要性。本书花了大量篇幅讲解索引基础的各种理论知识，以及如何综合评估索引使用的代价，而非简单介绍 SQL 如何编写，或者如何进行 SQL 优化等内容。认真通读本书之后，相信各位读者对数据库的索引设计、代价评估等方面就基本可以驾轻就熟了。本书经知含（曹怡倩）等人的悉心翻译，终于有了中文版，是国内广大 DBA 同行的福音，强烈建议购买阅读。

叶金荣

译者序一

记得第一次接触到这本书是在 2011 年的时候，那时我刚毕业参加工作不久，在公司里还处于学习知识和熟悉业务的阶段，当时带我的师傅童家旺先生经常会介绍一些他认为不错的书籍给团队的同事们阅读，这本 *Relational Database Index Design and the Optimizers* 就在他推荐给我们几位入职不久的新人的书单之中。我最初阅读的是书中第 3 章至 5 章的内容，读完后感到非常地痛快，只看一遍，所有的原理就都能清晰地呈现在脑海之中，于是随即将全书的其余章节也通读完成。从这本书中所学到的知识对我之后的工作带来了极大的帮助，使我在应用系统的索引设计及 SQL 调优上节省了大量的精力且取得了不错的效果。之所以能有如此成效，完全得益于 Tapio Lahdenmäki 和 Michael Leach 两位作者的贡献，是他们将宝贵的经验与智慧与广大读者分享。本书从表和索引结构开始，层层深入，通过大量的案例系统地阐释了关系型数据库的访问路径选择原理，以及表和索引的结构、扫描方式、硬件等对性能的影响；详尽地讲解了如何快速地估算 SQL 运行的 CPU 及执行时间的方法，使读者能够运用量化的方法进行判断和优化，指导关系型数据库的索引设计，真正做到知其所以然，进而能够举一反三，灵活运用于各类场景之中。

当得知有幸能够参与本书中文版的翻译工作时，我们感到非常兴奋，迫切地想把它翻译出来与更多的朋友分享。但当真正开始着手翻译时，才体会到这其中的艰难以及身负的责任，以至于有段时间愈翻译愈感到焦虑，深恐译本无法传达作者之原意。对于翻译中遇到的一些疑难问题，有时会反反复复阅读相关内容数遍，查阅各种资料以尽力获得较为准确的译法。加之我和赵建伟工作都较为繁忙，最终的交稿时间远远超出了原定计划。在此特别感谢王中英编辑在此过程中的督促、鼓励以及对翻译疏漏之处的校正。

本书由蚂蚁 DBA 团队的多位同事共同翻译完成，其中第 4、7、10、12、14、16 章由曹怡倩翻译，第 3、9、13、15 章由赵建伟翻译，第 1、2、5、6、8、11 章分别由刘晓军、叶莹佼、曹栖锋、柯翔宇、蔺瑞超、樊振华翻译。全书校稿及前言、术语表部分的翻译由曹怡倩完成。感谢恩师童家旺先生在本书翻译过程中的解惑和帮助。

本书原文通俗晓畅，但因我们个人水平及时间的限制，译文中难免存在

不准确或是不通顺之处,这全是我们译者的责任,与作者及帮忙的朋友无关;如有识者,望不吝赐教。

感谢电子工业出版社编辑张春雨、王中英以及其他所有工作人员在各方面的鼎力支持与几近无限的耐心。感谢在本书翻译过程中给予了我们如此多帮助与支持的朋友、同事和家人们。

感谢阿里巴巴 DBA 二部团队同事和领导给予的支持与帮助。感谢恩师童家旺先生在全书翻译过程中的解惑和帮助。感谢父母一直以来的关心照顾和体谅,使我能够有更多的精力专注于本书的翻译工作。

曹怡倩

译者序二

记得 2010 年 7 月毕业进入支付宝数据库管理团队，担任开发 DBA 一职，阿里系各子公司是国内当时为数不多的在数据库团队划分开发 DBA 和运维 DBA 两种角色的公司，开发 DBA 的主要职责包括评估业务开发人员数据库使用的申请，SQL 审核，上线评估，运行监控，容量规划等，贯穿整个产品的生命周期，其中 SQL 审核尤为烦琐和重要，在业务迅猛发展和小机+存储架构流行的年代，一条烂 SQL 足以搞垮多条业务产品线，修复的代价比较大，周期也比较长。

接触了一段工作一段时间后，我的师傅童家旺先生推荐我阅读本书的英文原版 *Relational Database Index Design and the Optimizers*，当时拿到这本书的时候不以为然，觉得在关系数据库领域遇到 SQL 性能问题时，无非就是添加 B 树索引。但当我真正翻开这本书的时候，真的是爱不释手，一口气把它读完，仍意犹未尽。这本书不仅讲述了如何建立三星索引的方法论，更重要的是给出了基于硬件和软件环境下索引设计的量化评估的方法和实践，掌握了这些方法后，你将能够提前量化业务 SQL 上线运行情况，并指导后期的容量评估。这本书并没有限定具体的商业或者开源的关系型数据库产品，而是讲述通用的理论和方法。所以，强烈建议数据库从业人员，使用关系型数据库的开发人员阅读本书，它会给大家带来体系化的理论和方法。

在此，感谢原支付宝数据库管理团队的领导和同事们合力完成本书的翻译工作，同时要感谢我的家人，是你们不辞辛苦地照顾家庭，我才能够在工作之余专注于本书的翻译。希望大家享受这本书。

赵建伟

2015 年 5 月于杭州

前言

关系型数据库至今已存在了三十多年。在其发展早期，由于硬件资源限制及优化器成熟度的不足，性能问题非常普遍，因此性能成为了人们优先考虑的事项。但现在情况已经不同了，硬件及软件以超出人们想象的速度发展了起来，系统已经能够自己关心自己的性能了，这在之前看来是不可思议的！但比这些资源增长速度更快的是随之产生的大量信息以及这些信息所衍生出的活动。另外，有一个重要的硬件还没有跟上整体的发展速度：虽然磁盘已变得更大且异常廉价，但它们的访问速度仍相对较慢。因此，许多老问题其实并没有消失——它们只是变换了形式。这其中的有些问题可能会造成巨大的影响——那些所谓的应该只需运行不到一秒的“简单”查询实际却运行了几分钟或更久，尽管所有的书中都写了如何正确编写查询、如何组织表，以及应当按照什么规则来决定将哪些列添加至索引上。所以，很明显，我们需要有一本能够突破常规的书，真正开始思考为何现今仍有这么多人还会遇到如此多的问题。

为了满足这一需求，我们认为必须关注两个问题。第一个必须关注的对象是关系型系统中用于确定如何以最高效的方式查询所需数据的部分（我们称其为 SQL 优化器）。第二个必须关注的是索引及表是以何种方式被扫描的。我们试着把自己放在优化器的角度思考问题，也许当我们理解为什么可能存在问题时，我们就能够做出改变。幸运的是，我们需要知道的有关优化器的内容其实非常少，但非常重要。本书与其他同领域的书籍的一个很重要的区别在于，我们不会提供大量的用于指导 SQL 编写以及表和索引设计的规则和语法。这不是一本告诉你在各种场景下应当使用哪一个 SQL WHERE 语句的书，也不是一本告诉你应当使用什么语法的书。如果我们努力遵循一大堆复杂、模糊甚至可能不完整的指导原则，那么我们就是在走前人走过的老路。相反，如果我们能够理解 SQL 请求对关系型系统造成的潜在影响，并知道如何控制这一影响，那么我们就能够理解、控制、最小化甚至避免这些问题。

本书的第二个目的是展示如何使用这些知识从 CPU 和执行时间的角度

量化运行过程。只有这样,我们才能真正判断我们设计的表和索引是否合适,我们需要用真实的数字来展示优化器是如何思考的、扫描将耗费多少时间,以及需要进行哪些改动以提供满意的性能。不过,最重要的是,我们必须能够方便且快速地完成这一评估过程,这就要求我们必须将关注点放在少数几个真正重要的问题上,而不是将关注点放在那些不那么重要的细节上(许多人都被这些细节问题困扰过)。所以,关键就是要关注少数核心领域,并能够说出这需要花费多少时间或成本。

同样是由于我们专注于核心问题,所以我们还能提供另一个优势。对于那些可能使用多个关系型产品(即便来自相同的供应商)的人,由于我们在本书中所使用的是一种适用于所有关系型产品的通用方法,所以使用者就不需要阅读和掌握多套截然不同的规则和建议。所有“真正的”关系型系统的优化器都有一个相同的任务:它们都必须扫描索引和表。它们都使用异常相似的方式来处理这些操作(虽然他们对其有各自不同的描述方式)。当然,它们之间确实存在着一些差异,但是我们可以毫不费力地处理这些不同。

也正是由于相同的原因,本书的读者对象包括:认为了解 SQL 性能方面的知识或如何有效设计索引的知识能给自己带来益处的人,直接负责索引设计的人,编写 SQL 语句用于查询或作为应用程序一部分的人,以及那些负责维护关系型数据和关系型环境的人。只要你觉得需要对自己所做的事情的性能影响负责,那么你都将不同程度地从本书中受益。

最后,用一句话概括本书目标读者所需具备的背景知识:我们假定读者已经具备了 SQL 这一关系型语言相关的知识。考虑阅读本书的人应该已经具备了对计算机系统的大体理解。除此以外,能帮到读者的最重要的品质也许就是对事物运行原理的好奇和兴趣了,还有想把事情做得更好的渴望。另一方面,在众多拥有几十年的关系型系统经验的人中,有两类人也会从本书受益:第一类是那些根据详细的规则手册良好地管理了系统很多年的人,他们想通过理解这些规则适用的原因来使自己的工作更轻松一些;第二类是那些已经使用了本书中所描述的技术很多年,但对于新硬件所带来的改善并不赞赏的人。

本书中的绝大部分观点及使用的技术都是原创的,因此很少有对外部出版物及其他作者成果的引用。在本书的创作过程中,我们非常感谢给予了我们如此多帮助和鼓励的朋友及同事们。感谢 Matti Ståhl 在全书撰写过程中所给予的详细指点及批判性但极其有用的建议。感谢 Lennart Henäng、Ari Hovi、Marja Kärmeniemi 和 Timo Raitalaakso 的帮助和校对,也感谢 Akira Shibamiya

在关系型性能公式上的原创工作。另外，还要感谢许许多多的学生和数据库顾问们，感谢他们提供的对于实际问题及其解决方案的深入见解。最后，特别感谢 Meta 和 Lyn，没有他们的鼓励与支持，本书不可能完成，Meta 还特别为本书设计了封面，与全书的主旨非常契合。

Tapio Lahdenmäki (斯姆勒尼科, 斯洛文尼亚)

Michael Leach (什鲁斯伯里, 英格兰)

目录

第 1 章 概述	1
关于 SQL 性能的另一本书.....	1
不合适的索引.....	3
误区和误解.....	4
误区 1: 索引层级不要超过 5 层.....	5
误区 2: 单表的索引数不要超过 6 个.....	6
误区 3: 不应该索引不稳定的列.....	6
示例.....	7
磁盘驱动器使用率.....	7
系统化的索引设计.....	8
第 2 章 表和索引结构	10
介绍.....	10
索引页和表页.....	11
索引行.....	11
索引结构.....	12
表行.....	12
缓冲池和磁盘 I/O.....	12
从 DBMS 缓冲池进行的读取.....	13
从磁盘驱动器进行的随机 I/O.....	13
从磁盘服务器缓存进行的读取.....	14
从磁盘驱动器进行的顺序读取.....	15
辅助式随机读.....	15
辅助式顺序读.....	18
同步 I/O 和异步 I/O.....	18
硬件特性.....	19
DBMS 特性.....	20
页.....	20

表聚簇.....	21
索引行.....	21
表行.....	22
索引组织表.....	22
页邻接.....	23
B 树索引的替代品.....	24
聚簇的许多含义.....	25
第 3 章 SQL 处理过程.....	27
简介.....	27
谓词.....	27
评注.....	28
优化器及访问路径.....	28
索引片及匹配列.....	29
索引过滤及过滤列.....	29
访问路径术语.....	31
监控优化器.....	32
帮助优化器（统计信息）.....	32
帮助优化器（FETCH 调用的次数）.....	32
何时确定访问路径.....	33
过滤因子.....	34
组合谓词的过滤因子.....	35
过滤因子对索引设计的影响.....	37
物化结果集.....	39
游标回顾.....	39
方式 1：一次 FETCH 调用物化一条记录.....	40
方式 2：提前物化.....	41
数据库设计人员必须牢记.....	41
练习.....	41
第 4 章 为 SELETE 语句创建理想的索引.....	43
简介.....	43
磁盘及 CPU 时间的基础假设.....	44
不合适的索引.....	44