

<<SQL应用重构>>

图书基本信息

书名：<<SQL应用重构>>

13位ISBN编号：9787111263586

10位ISBN编号：7111263588

出版时间：2010-1

出版时间：机械工业出版社

作者：（美）法拉特 等著，苏敬凯 等译

页数：286

译者：苏敬凯

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<SQL应用重构>>

前言

在翻译本书时，正流行着几本名为“啥啥之美”的书。

本书也可以说讲的就是“如何将SQL写得更美”。

SQL之美就在于用清楚合理的方式告诉DBMS你要做什么，让DBMS总能以最优的方式高效地完成任

务。SQL是一种描述式语言，在描述中要表达清楚能够影响优化器决策的要点，同时避免过程化思维的那些细节。

而要做到这一点，理解优化器的工作方式和SQL的理念至关重要，正如某条广告语讲的那样“知其道，用其妙”。

本书作者从事数据库咨询工作多年，在本书中，不仅举出了很多具体生动的例子，还讲述了很多重构SQL应用的思想。

无论你是DBA还是程序员，本书都能帮你写出或改写（也就是重构）出优美的SQL语句、清楚明了的处理过程，从而交付出高效的系统，赢得众人的赞许，就像书中的很多故事那样。

参加本书翻译的还有赵龙刚、金振林、周志强、杨宁等。

本书翻译力求忠于原著，但由于时间仓促，译者水平有限，翻译错误和不妥之处在所难免，欢迎广大读者批评指正。

<<SQL应用重构>>

内容概要

当数据库的性能达不到预期时，该怎么办呢？

在用昂贵的硬件升级的力、法来解决这一问题之前，请拿起这本书。

本书将教你如何发现和评估需要重构的代码，理解重构和性能之间至关重要的关系。

如果你的应用陷入了困境，那么本书将能帮你使它重新加快速度。

在本书中你将学习到：

- 判断你是否(以及在哪儿)可以得到性能的提升。

- 应用快速修复的方法，例如在存储函数和过程中限制对数据库的调用。
- 改写SQL语句以提高数据访问的效率。
- 重构任务，例如用存储过程代替应用代码，用全面的SQL语句代替重复的过程化语句。
- 增加并行以重构流程。
- 使用模式扩展、常规视图、物化视图、分区等来重构设计。

<<SQL应用重构>>

作者简介

作者：(美国)Stephane Farout(法拉特) (美国)Pascal L Hermite 译者：苏敬凯 等Stephane Faroult，从1983年开始接触关系数据库和SQL语言，他从事数据库咨询工作已经20多年了。O'Reilly的《The Art of SQL》也是他的作品。

<<SQL应用重构>>

书籍目录

前言第1章 评估 一个简单的例子 评估可能的收益 第2章 健全检查 统计信息与数据失真 检查索引
解析与绑定变量 大数据量操作 事务管理 第3章 用户函数和视图 用户自定义函数 视图 第4章 测试框
架 生成测试数据 比较备选版本 第5章 语句重构 执行计划和优化器指示 分析缓慢查询 重构查询核
心 重新构建最初的查询第6章 任务重构 SQL的理念 更改代码结构第7章 重构流程和数据库 重组处理
过程 撼动基础第8章 实践中的重构 你能看到数据库吗 失败的查询 速度很快的查询 并非显然完全错
误的查询 结束语附录A 脚本及样例程序附录B 工具

<<SQL应用重构>>

章节摘录

插图：如果一个索引不能达到我们所追求的效果，那么有时一个更好的索引却能提供更好的性能。有个问题，为什么要在accountid上单独建立一个索引？

从本质上说，索引是键值的一个已排序列表（在树中排序），这些键值关联着和它们匹配的记录的物理地址，就像本书后面的索引一样——它是一个关键字及相关页码的已排序的列表。

如果我们搜索的是两个字段的值而索引只有其中的一个字段，那么我们就不得不读出所有符合所搜索的键值的记录，然后再丢掉那些在另一个字段上值不匹配的记录。

如果基于这两个字段建立索引，那就能直接得到真正想要的记录了。

我们可以创建一个基于（accountid，txdate）的索引，因为交易日期是查询的另一个条件。

创建了这个基于两个字段的复合索引之后，我们就能确保SQL引擎在索引上能够进行有效的绑定搜索（这叫做范围扫描）。

在我的测试数据中，单字段索引使MySQL的性能提升了3.1倍，而双字段的索引则使速度提高了3.4倍，现在运行这一程序只需要花3.5分钟。

而坏消息是：在Oracle和SQLServer上，即便使用了双字段索引，也没有取得比之前单字段索引时更进一步的性能提升（参见表1.4）。

与我在现实生活中遇到过的很多处理过程一样，这个例子的明显特征是嵌套循环。

并且我们还发现：在循环的深处，返回的每一条记录都要调用工具函数AboveThreshold（）。

我前面提到过，transactions表包含了200万条记录，而且其中1 / 7的记录属于我们要检查的area。

这样，我们就调用了AboveThreshold（）函数很多很多次。

如果一个函数被调用了极多次，每一个非常小的单一修改都会因巨大的杠杆效应而获益颇丰。

比如假设我们把一次调用的执行时间从0.005秒降到了0.004秒。

<<SQL应用重构>>

媒体关注与评论

“有很多讲述程序重构的书，但一直缺少讲述数据库代码重构的书，直到本书出版为止。在StephaneFaroult的这本新书中有很多高级的SQL技术，我一直在自己的工作中使用这些技术。我热情地向大家推荐这本书。

” ——MichaelBlaha，咨询师，OMTAssociatesInc。“终于有了这样一本书，它强调TSQL编写者在数据库总体性能上的作用，以及怎么来改进这一情形。

对于任何一位数据库专业人士来说，只要你想要提升自己的查询编写能力，或者想要改进别人写的查询，那么本书就是你的必读之书。

” ——DwayneKing，总裁，KRIDANConsulting“本书装满了宝贝。在放下本书之前，你一定会体验到惊喜。

在本书中，Faroult先生慷慨地分享了他的那些丰富的经历和清晰的思维。

” ——RoyOwens，数据库开发人员，CBORDGroup，Inc

<<SQL应用重构>>

编辑推荐

《SQL应用重构》由机械工业出版社出版。

<<SQL应用重构>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>