

《数据库索引设计与优化》

图书基本信息

书名：《数据库索引设计与优化》

13位ISBN编号：9787121260549

出版时间：2015-6

作者：【美】Tapio Lahdenmaki, 【美】Michael Leach

页数：320

译者：曹怡倩,赵建伟

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《数据库索引设计与优化》

内容概要

《数据库索引设计与优化》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。

《数据库索引设计与优化》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，《数据库索引设计与优化》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。

《数据库索引设计与优化》

作者简介

Tapio Lahdenmaki, 数据库性能顾问, 教授通用索引设计课程。他在IBM公司工作了三十多年, 是公司全球课程中有关DB2 (for z/OS)性能相关课程的主要作者。Michael Leach, 关系型数据库顾问, 已从IBM公司退休, 他拥有二十年的应用系统及数据库课程的教授经验。两位作者的文章均被翻译成了多国语言广为传播。他们有关索引设计的方法被成功应用于许多核心系统。

书籍目录

| | |
|-------------------------|----|
| 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 |
| 三星索引——查询语句的理想索引 | 45 |
| 星级是如何给定的 | 46 |
| 范围谓词和三星索引 | 48 |
| 为查询语句设计最佳索引的算法 | 49 |
| 候选A | 50 |
| 候选B | 50 |
| 现今排序速度很快——为什么我们还需要候选B | 51 |
| 需要为所有查询语句都设计理想索引吗 | 52 |
| 完全多余的索引 | 52 |
| 近乎多余的索引 | 53 |
| 可能多余的索引 | 53 |
| 新增一个索引的代价 | 54 |
| 响应时间 | 54 |
| 磁盘负载 | 55 |
| 磁盘空间 | 56 |
| 一些建议... .. | 57 |
| 练习 | 58 |
| 第5章 前瞻性的索引设计 . | 59 |
| 发现不合适的索引 | 59 |
| 基本问题法（BQ） | 59 |
| 注意... .. | 60 |
| 快速上限估算法（QUBE） | 61 |
| 服务时间 | 62 |
| 排队时间 | 62 |
| 基本概念：访问 | 63 |
| 计算访问次数 | 65 |
| FETCH 处理..... | 66 |
| 主要访问路径的QUBE 示例 | 67 |
| 使用满足需求的成本最低的索引还是所能达到的最优索引：示例1 | 72 |
| 该事务的基本问题 | 73 |
| 对该事务上限的快速估算 . | 73 |
| 使用满足需求的成本最低的索引还是所能达到的最优索引 | 74 |
| 该事务的最佳索引 | 75 |
| 半宽索引（最大化索引过滤） | 75 |
| 宽索引（只需访问索引） . | 76 |
| 使用满足需求的成本最低的索引还是所能达到的最优索引：示例2 | 77 |
| 范围事务的BQ 及QUBE .. | 78 |

| | |
|-------------------------------|-----|
| 该事务的最佳索引 | 79 |
| 半宽索引（最大化索引过滤） | 80 |
| 宽索引（只需访问索引） | 81 |
| 何时使用QUBE | 82 |
| 第6章 影响索引设计过程的因素 | 83 |
| I/O 时间估算的验证 | 83 |
| 多个窄索引片 | 84 |
| 简单就是美（和安全） | 86 |
| 困难谓词... | 87 |
| LIKE 谓词 | 87 |
| OR 操作符和布尔谓词 | 88 |
| IN 谓词..... | 89 |
| 过滤因子隐患 | 90 |
| 过滤因子隐患的例子 | 92 |
| 最佳索引 | 95 |
| 半宽索引（最大化索引过滤） | 96 |
| 宽索引（只需访问索引） | 97 |
| 总结... | 97 |
| 练习 | 99 |
| 第7章 被动式索引设计 .. | 100 |
| 简介 | 100 |
| EXPLAIN 描述了所选择的访问路径 | 101 |
| 全表扫描或全索引扫描 ... | 101 |
| 对结果集排序 | 101 |
| 成本估算 | 102 |
| 数据库管理系统特定的EXPLAIN 选项及限制 | 102 |
| 监视揭示现实 | 103 |
| 性能监视器的演进 | 104 |
| LRT 级别的异常监视 | 106 |
| 程序粒度的均值是不够的 | 106 |
| 异常报告举例：每个尖刺一行 | 106 |
| 问题制造者和受害者 | 108 |
| 有优化空间的问题制造者和无优化空间的问题制造者 | 108 |
| 有优化空间的问题制造者 | 109 |
| 调优的潜在空间 | 111 |
| 无优化空间的问题制造者 | 114 |
| 受害者 | 115 |
| 查找慢的SQL 调用 | 117 |
| 调用级别的异常监视 | 118 |
| Oracle 举例 | 121 |
| SQL Server 举例 | 123 |
| 结论 | 125 |
| 数据库管理系统特定的监视问题 | 126 |
| 尖刺报告 | 127 |
| 练习 | 127 |
| 第8章 为表连接设计索引 | 129 |
| 简介 | 129 |
| 两个简单的表连接 | 131 |
| 例8.1：CUST 表作为外层表 | 131 |

| | |
|---------------------------------|-----|
| 例8.2：INVOICE 表作为外层表 | 132 |
| 表访问顺序对索引设计的影响 | 133 |
| 案例研究 | 133 |
| 现有索引 | 136 |
| 理想索引 | 142 |
| 理想索引，每事务物化一屏结果集 | 146 |
| 理想索引，每事务物化一屏结果集且遇到FF 缺陷 | 149 |
| 基本连接的问题（BJQ） | 151 |
| 结论：嵌套循环连接 | 153 |
| 预测表的访问顺序 | 153 |
| 合并扫描连接和哈希连接 | 155 |
| 合并扫描连接 | 155 |
| 例8.3：合并扫描连接 | 155 |
| 哈希连接 | 157 |
| 程序C：由优化器选择MS/HJ（在现有索引条件下） | 158 |
| 理想索引 | 159 |
| 嵌套循环连接VS. MS/HJ 及理想索引 | 161 |
| 嵌套循环连接VS. MS/HJ | 161 |
| 嵌套循环连接VS.理想索引 | 162 |
| 连接两张以上的表 | 163 |
| 为什么连接的性能表现较差 ... | 166 |
| 模糊的索引设计 | 166 |
| 优化器可能选择错误的表访问路径 | 166 |
| 乐观的表设计 | 166 |
| 为子查询设计索引 | 167 |
| 为UNION 语句设计索引 | 167 |
| 对于表设计的思考 | 167 |
| 冗余数据 | 167 |
| 无意识的表设计 | 171 |
| 练习 | 173 |
| 第9章 星型连接 | 175 |
| 介绍 | 175 |
| 维度表的索引设计 | 177 |
| 表访问顺序的影响 | 178 |
| 事实表的索引 | 179 |
| 汇总表 ... | 182 |
| 第10章 多索引访问 | 184 |
| 简介 | 184 |
| 索引与 ... | 184 |
| 与查询表一同使用索引与 | 186 |
| 多索引访问和事实数据表 | 187 |
| 用位图索引进行多索引访问 | 187 |
| 索引或 ... | 188 |
| 索引连接 .. | 189 |
| 练习 | 190 |
| 第11章 索引和索引重组 .. | 191 |
| B 树索引的物理结构 | 191 |
| DBMS 如何查找索引行 | 192 |
| 插入一行时会发生什么 | 193 |

| | |
|---------------------------|-----|
| 叶子页的分裂严重吗 | 194 |
| 什么时候应该对索引进行重组 | 196 |
| 插入模式 | 196 |
| 索引列的稳定性 | 205 |
| 长索引行..... | 207 |
| 举例：对顺序敏感的批处理任务 | 208 |
| 表乱序（存在聚簇索引） | 211 |
| 表乱序（没有以CNO开头的聚簇索引） | 212 |
| 存储在叶子页中的表行 | 212 |
| SQL Server | 212 |
| Oracle | 213 |
| 索引重组的代价 | 214 |
| 分裂的监控 | 215 |
| 总结 | 216 |
| 第12章 数据库管理系统相关的索引限制 | 219 |
| 简介 | 219 |
| 索引列的数量 | 219 |
| 索引列的总长度 | 220 |
| 变长列 | 220 |
| 单表索引数量上限 | 220 |
| 索引大小上限 | 220 |
| 索引锁定 .. | 221 |
| 索引行压缩 | 221 |
| 数据库管理系统索引创建举例 | 222 |
| 第13章 数据库索引选项 .. | 224 |
| 简介 | 224 |
| 索引行压缩 | 224 |
| 索引键以外的其他索引列 | 225 |
| 唯一约束 | 227 |
| 从不同的方向扫描数据库索引 | 227 |
| 索引键截断 | 228 |
| 基于函数的索引 | 228 |
| 索引跳跃式扫描 | 229 |
| 块索引 | 230 |
| 数据分区的二级索引 | 230 |
| 练习 | 231 |
| 第14章 优化器不是完美的 | 232 |
| 简介 | 232 |
| 优化器并不总能看见最佳方案 | 234 |
| 匹配及过滤问题 | 234 |
| 非BT谓词 | 234 |
| 无法避免的排序 | 237 |
| 不必要的表访问 | 238 |
| 优化器的成本估算可能错得离谱 | 239 |
| 使用绑定变量的范围谓词 | 239 |
| 偏斜分布 | 241 |
| 相关列 | 242 |
| 部分索引键的警示故事 ... | 243 |
| 成本估算公式 | 246 |

| | |
|-----------------------------|-----|
| 估算I/O 时间 | 247 |
| 估算CPU 时间 | 248 |
| 协助优化器处理估算相关的问题 | 249 |
| 优化器的问题是否会影响索引设计 | 252 |
| 练习 | 253 |
| 第15 章 其他评估事项 | 254 |
| QUBE 公式背后的假设条件 ... | 254 |
| 内存中的非叶子索引页 | 255 |
| 例子 | 255 |
| 磁盘服务器读缓存的影响 | 256 |
| 缓冲子池 | 258 |
| 长记录 | 259 |
| 慢速顺序读 | 259 |
| 实际的响应时间可能比QUBE 评估值短得多 | 259 |
| 叶子页和表页缓存在缓冲池中 | 260 |
| 识别低成本的随机访问 ... | 262 |
| 辅助式随机读取 | 262 |
| 辅助式顺序读 | 265 |
| 评估CPU 时间 (CQUBE) ... | 265 |
| 单次顺序访问的CPU 时间 | 265 |
| 单次随机访问的CPU 时间 | 267 |
| 单次FETCH 调用的CPU 时间 | 269 |
| 每排序一行的平均CPU 时间..... | 270 |
| CPU 评估举例 | 270 |
| 宽索引还是理想索引 | 270 |
| 嵌套循环 (及反范式化) 还是MS/HJ..... | 271 |
| 合并扫描与哈希连接的比较 | 274 |
| 跳跃式顺序扫描 | 275 |
| CPU 时间仍然不可忽视 .. | 276 |
| 第16 章 组织索引设计过程 | 277 |
| 简介 | 277 |
| 计算机辅助式索引设计 | 278 |
| 设计出索引的9 个步骤 | 280 |
| 参考文献 | 282 |
| 术语表 | 283 |
| 索引 | 291 |

《数据库索引设计与优化》

精彩短评

- 1、花了2个月看完了，赞
- 2、总体来说因为用不上所以收获并没有那么大.....
- 3、内容10年前的，能与时俱进就好了；作为一个MySQL DBA，对一些Oracle、DB2相关内容不感兴趣~
- 4、惊艳
- 5、一个小时翻了一遍；几乎没有收获，只是谈了一些索引创建的常规做法，其实一篇博客就可以讲的更清楚。
里面各种规范化概念推演，然而并没什么用处。使用explain看一看已经能够解决一半以上的问题了。
- 6、是块硬骨头，需要慢慢啃，很有含金量！
- 7、很好的关于索引介绍的书，提供了估计查询执行时间和方法，并解释了索引对于查询效率的影响方式，对实践和指导意义
- 8、书本身不错。基本上索引设计的方方面面都有涉及。互联网时代，大家都在关注大数据，大数据的框架比比皆是，Hadoop，spark等等。越来越少的人去关注单台数据库的性能问题，动不动都横向扩展之类的。这种环境下，我觉得反而应该去关注一些实际落地的技术，研究一下数据库原理。
- 9、高屋建瓴（跳过了后半部分 join 优化和一些高级公式）
- 10、了解三星索引，了解如何评价索引，了解如何设计三星索引，看完这本书的，或者掌握这本书中的内容的朋友，才是优秀的DBA
- 11、正合适我读。
- 12、有点难
- 13、索引设计说起来简单，但DBA，开发人员能够了解清楚，深入的寥寥无几。
这本书更为关键的是有一套量化的方法，读完之后的畅快，看了才懂，强烈推荐。
另外，不要迷信什么nosql，能把存在了几十年的关系型数据库搞懂，索引是必不可少的重要一环。

1、《数据库索引设计与优化》的笔记-三星索引

如果与一个查询相关的索引行是相邻的，或者至少足够靠近的话，那这个索引就可以被标记上第一颗星。这最小化了必须扫描的索引片的宽度。如果索引行的顺序与查询语句的需求一致，则索引可以被标记上第二颗星。这排除了排序操作。如果索引行包含查询语句中的所有列——那么索引就可以被标记上第三颗星。这避免了访问表的操作，仅访问索引就可以了。对于这三颗星，第三颗通常是最重要的。

《数据库索引设计与优化》

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：www.tushu111.com