



Amazon全五星评价畅销书，由拥有20余年工作经验的资深数据库专家撰写，权威性毋庸置疑

详尽阐述了100余条SQL语句优化的技巧和最佳实践，以及编写高性能SQL语句的标准和原则，包含大量案例，为解决各种复杂的DB2性能问题提供了解决方案

IBM
PRESS

DB2 SQL

Tuning Tips for z/OS Developers

DB2 SQL性能调优秘笈

(美) Tony Andrews 著
陈勇 杨健康 译



机械工业出版社
China Machine Press

DB2 SQL Tuning Tips for z/OS Developers

DB2 SQL性能调优秘笈

(美) Tony Andrews 著
陈勇 杨健康 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

DB2 SQL性能调优秘笈 / (美) 安德鲁 (Andrews, T.) 著; 陈勇, 杨健康译. —北京: 机械工业出版社, 2013.5

(华章程序员书库)

书名原文: DB2 SQL Tuning Tips for z/OS Developers

ISBN 978-7-111-42502-1

I. D… II. ①安… ②陈… ③杨… III. 关系数据库系统 IV. TP311.138

中国版本图书馆CIP数据核字 (2013) 第101045号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2012-8329

Authorized translation from the English language edition, entitled DB2 SQL Tuning Tips for z/OS Developers, 9780133038460 by Tony Andrews, published by Pearson Education, Inc., Copyright © 2013 by International Business Machines Corporation.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese Simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2013.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和中国香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

这是一本不可多得的 DB2 数据库性能调优秘笈, 由拥有 20 余年 DB2 工作经验的资深数据库专家撰写, Amazon 全五星评价畅销书。本书不仅详尽阐述了 100 余条 SQL 语句优化的技巧和最佳实践、编写高性能 SQL 语句的标准和原则, 以及 DB2 数据库性能优化的“15 步法”, 而且还包含大量案例, 为解决各种复杂的 DB2 性能问题提供了解决方案。

全书共 7 章: 第 1 章总结了 116 条优化 SQL 语句的技巧和最佳实践; 第 2 章讲解了 SQL 语句中提示的编写方法和技巧; 第 3 章讲解了编写高质量 SQL 语句需要遵守的 SQL 标准和原则; 第 4 章介绍了 SQL 程序走查; 第 5 章用 2 个实例介绍了如何检查存在性; 第 6 章介绍了 Runstats, 用来得到一些统计信息; 第 7 章讲解了 DB2 性能优化的“15 步调优法”, 被誉为 DB2 性能调优领域的无价之宝, 适用于各种情况下的性能调优问题。本书最后还有 2 个附录, 分别介绍了谓词的重写示例和 DB2 SQL 中的术语。

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 关 敏

北京京师印务有限公司印刷

2013 年 6 月第 1 版第 1 次印刷

145mm × 210mm · 5.375 印张

标准书号: ISBN 978-7-111-42502-1

定 价: 39.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 8837964 68995259

读者信箱: hzsjj@hzbook.com

译者序

拿到这本书，你可能会有这样的疑惑：这么一本薄薄的小书能告诉我们什么？真的能像作者所说的那样是一本不可多见的调优技巧秘笈吗？要回答这些疑问，最好的办法就是翻开书，自己研读，自己给出答案。

也许你在为应用的调优一筹莫展，也许你在惊异于调优专家的妙手回春，也许你在四处探求应用调优的秘诀，也许你有志于调优却不知从何做起……你手上的这本书会帮你解除疑惑，带你步入调优世界。

这本书的作者是著名的“Tony the Tuner”——Tony Andrews，他在 DB2 领域奋战了 20 多年，现在将他积累的经验凝结成书中的 100 多个 SQL 优化技巧，利用这些技巧，我们可以用最佳的方式构建和优化 DB2 应用。通过大量示例，他为我们展示了 DB2 SQL 调优中的种种问题和解决方案，强调了通常被忽略但意义重大的诸多方面，比如建立标准和原则，实现代码走查、重写查询谓词等等。相信这本书能帮你拨开谜团，享受成功调优的乐趣。

本书由陈勇、杨健康主译，程芳、吴忠望、张练达、陈峰、江健、姚勇、张莹参与了全书的修改整理，并完善了关键部分的翻译。全体人员共同完成了本书的翻译工作。不过由于水平有限，译文肯定有不当之处，敬请批评指正。

前 言

大多数关系型数据库调优专家都认为，对于众多访问关系型数据库的应用来说，性能问题主要是因为程序编写不当，或者 SQL 编写有问题。业内专家还指出，80% 的响应时间问题都应由性能差的 SQL 负责。我个人也非常赞同这种观点。我曾为很多 IT 工作室提供过性能和调优咨询服务，在所有这些工作室中，大多数性能问题都与修改应用和 SQL 代码直接相关，或者与增加和修改索引有关。正是因为这个原因，我一直在努力让开发人员了解各种 SQL 编程方法以及相关的性能问题。我认为应该让更多的开发人员了解如何阅读和分析 DB2 Explain 输出。另外，我相信对于每一个涉及 RDMS 的大型开发项目而言，项目组里都应该配备一个 SQL 技术专家。IT 行业中的 SQL 开发人员并不少，不过从我的经验来看，我发现其中只有不到 10% 的人真正了解 SQL 编程中存在的性能问题，或者真正知道该如何修正这些问题。如果有一位 SQL 技术专家坐镇，很多性能和逻辑问题就能在迁移到生产版本之前及早发现。

本书的出发点是为需要对 SQL 语句或程序调优的开发人员提供一个参考指南。一旦发现程序或应用运行很慢，大多数开发人员都会仓促地归咎于网络、数据库、系统或者过大的事务量等，把它们当作罪魁祸首。不过，很多情况下程序运行速度慢的直接原因其实在于他们的代码。希望本书能为这些开发人员提供帮助，使他们在迫不及待地给 DBA 或其他人打电话求助之前，能够先自己尝试来解决或改善遇到的性能问题。

还有另外一种情况：尽管 SQL（以及 SQL Explain）看起来还不错，但实际效率很差。针对这些情况，本书还提供了一些“调优

技巧”，这些技巧会采用与优化工具选用方法不同的方式对 SQL 进行调整优化。业界很多专家就经常使用这些技巧采用不同方式对原先性能表现很差的 SQL 语句进行优化，使它能更快地执行。有些情况下，修正性能问题的关键在于运行时间，这些调优技巧对于这种情况尤其适用。

在本书中，我总结了我在众多 IT 工作室所采用的一组 SQL 标准和原则。如果你的 IT 工作室还没有建立任何 SQL 标准，可以把提供的这些标准作为不错的起点。很多工作室都会针对他们的应用另外增加更多原则。

有了标准和原则还不够，还需要确保遵循这些原则。我见过很多这样的 IT 工作室，开发人员向我展示了他们的 SQL 编程、COBOL 编程、Java 编程等等标准，但是并没有建立质量保证机制来确保这些标准确实得到实施。所有将进入生产阶段的程序都要经过某种代码走查或审查，以确保遵循标准，另外要保证所写的 SQL 确实高效。审查可以确保程序和 SQL 逻辑正确，还可以确保程序的设计符合需求。本书中专门有一章列出了完成 SQL 编程代码走查时需要询问和检查的各个问题。代码走查需要花一点时间，不过与可能在生产阶段暴露的性能或逻辑问题相比，代码走查绝对是值得的。

很多情况下，开发人员希望对 SQL 代码调优，但是不知道从哪里做起。我会告诉他们，首先要查看各个查询中的所有谓词，尽可能编写得更高效。本书提供了一个附录，其中列出了性能不好的 SQL 谓词，以及如何更高效地重写这些谓词。开发人员要知道一个谓词是可索引谓词还是不可索引谓词，另外要知道它是 Stage 1 谓词还是 Stage 2 谓词，这很重要。这些问题将在本书中详细讨论。

与其说性能是一个 DB2 问题，还不如说它是一个关系型数据库问题。开发人员必须注意如何建立查询结构，以及如何基于这些查询设计应用代码。数据库分析人员和数据库建模人员必须注意如

何设计一个数据库应用。他们需要花一些时间做充分的分析。性能取决于环境、应用和需求。另外，对于性能来说，没有最好，只有更好。尽管大多数调优都在操作系统级和数据库级进行，但是实际上对应用代码调优才能获得最大的性能收益。如果应用需要使用 SQL 从数据库中获取数据，这一类应用就非常适合进行调优。由于很少的 SQL 语句可能会占用大多数资源，所以 SQL 调优通常能得到显著的性能提升。不过，SQL 调优有时可能相当复杂。本书会提供一个起点，力求简单，使开发人员能轻松上手，让 SQL 驱动的程序和应用更高效地执行。

免责声明

本书中给出的调优技巧和建议是我个人的观点，这些都是根据我多年来完成 DB2 应用设计、编程和调优总结得出的。其中一些调优技巧可能并不完全反映 IBM（以及相关组织或这个领域的专家）的看法和观点。这些调优技巧基于我个人的经验积累而成，在我参与的各个应用中频繁使用，使应用得到更好的性能。这 100 多个技巧中，每一个技巧我都曾在对 DB2 SQL 应用调优时使用过，力图使查询和程序更高效地执行。

谈到一种方法好用或不好用时，很多人都会说“看情况”，相信每个人对此都不陌生。一定要谨记这一点。不要机械地照搬本书中的各个技巧，想当然地认为性能会立即改善。这些技巧是为了给开发人员指明方向，提供一些思路，以便他们改进查询或程序。所有人都要自己完成独立测试，验证本书中一些说法的正确性，只有在验证了这些说法确实正确的前提下，才能依此做出判断和决策。

致谢

我要向这些年来与我共事过的广大开发人员和 DBA 致以深深

的谢意。我是程序设计走查和程序代码走查的忠实拥护者。就我个人而言，多年来与其他同事坐在一起讨论、采纳他们关于程序改善的建议，期望改进得到最好、最高效的代码而顺利进入生产阶段，这些让我学到了很多很多。多年前，我们在一起开发一些庞大的应用时，他们就催促我把经验总结下来。后来我不断在会议上或者通过邮件为我们的开发小组提供关于编码技术的技巧，以改善他们的查询和程序性能。最后，终于有一天他们都劝说我应把这些技巧汇集成一本书。

我要特别感谢完成技术编辑的 Chuck Kosin 和 David Simpson。他们是我认识的 DB2 领域最棒的技术人员，他们能加入这个项目真是我的荣幸。非常感谢他们给出的大量评论和建议，正是这些意见让这本书更出色。我认识 David 已经有 3 年多了，我们曾在 Themis Inc 共事过。说心里话，他是我在这个行业见过的知识最渊博、最有经验的 DB2 专家之一。Chuck 有非常深厚的技术背景，之前就曾做过技术编辑，他与 Craig Mullins 共同编辑了《DB2 Developer's Guide》。

另外，非常感谢 IBM 出版社和我一同完成这本书的工作人员，感谢他们的理解和耐心。特别要感谢 Mary Beth Ray、Steven Stansel、Christopher Cleveland 以及参与这本书第一版的所有编辑和制作人员。要按照规定的所有格式和风格汇集成这样一本书，对我来说真不是一件容易的事。我发自内心地认为确实需要更高的 Word 处理水平才能胜任这个工作。

要感谢我任职的 Themis 公司（位于新泽西州的韦斯特菲尔德），感谢大家支持我完成这个项目。在我多年的培训和咨询生涯中，这是我遇到和共事过的水平最高、沟通最顺畅的一群技术人员。有了他们的协助，紧跟最新的版本与时俱进也成了一件乐事，出现问题时他们的第一手经验真是无价之宝。

另外，最重要的是，谢谢我亲爱的妻子 Jan。是她帮助我建立信心、消除恐惧，鼓励我把在这么多项目中学到和实现过的方法公

开出版，与大家分享。

如果你对本书有任何问题或评论，可以与我联系 (tandrews@themisinc.com)。也可以写信给出版社转交给我。真诚期待你们的宝贵建议，实际上早在多年前做程序走查时我就已经感受到他人意见的重要性，从那时起就不敢再自以为是，衷心欢迎你的意见和反馈。本书的目的就是帮助广大开发人员了解各种高效 DB2 SQL 编程的方法。

目 录

译者序

前言

第 1 章 SQL 优化技巧宝典 100+	1
1. 去除在谓词列上编写的任何标量函数	3
2. 去除在谓词列上编写的任何数学运算	4
3. SQL 语句的 Select 部分只写必要的列	4
4. 尽可能不用 Distinct	5
5. 尽量将 In 子查询重写为 Exists 子查询	7
6. 确保宿主变量定义为与列数据类型匹配	7
7. 由于优化工具处理“或”逻辑可能有问题，所以尽量 采用其他方式重写	8
8. 确保所处理的表中数据分布和其他统计信息正确 并反映当前状况	9
9. 尽可能用 UNION ALL 取代 UNION	11
10. 考虑使用硬编码还是使用宿主变量	12
11. 尽可能减少 DB2 的 SQL 请求	13
12. 尽量将区间谓词重写为 Between 谓词	15
13. 考虑使用全局临时表	16
14. 优先使用 Stage 1 谓词而不是 Stage 2 谓词	18
15. 记住（某些）谓词的顺序很重要	20
16. 多个子查询排序	21
17. 索引关联子查询	22
18. 了解 DB2 Explain 工具	23
19. 使用工具进行监控	24
20. 采用提交和重启策略	24

21. 实现优良的索引设计	25
22. 避免与非列表表达式不一致	26
23. 所有筛选逻辑放在应用代码之外	27
24. 确保涉及 Min 和 Max 的子查询谓词要处理可能 返回 Null 的情况	28
25. 如果查询只选择数据，一定要把游标处理写为 For Fetch Only（只获取）或 For Read Only（只读）	29
26. 避免只是为了帮助确定代码逻辑应当执行更新还是 插入而从表中选择一行	30
27. 避免只是为了得到更新值而从表中选择一行	31
28. 利用动态 SQL 语句缓存	31
29. 避免使用 Select *	32
30. 当心可以为 Null 的列，还要当心 SQL 语句可能从 数据库管理器返回 Null	33
31. 尽量减少执行打开和关闭游标的次数	34
32. SQL 中要避免非逻辑	34
33. 使用关联 ID 来保证更好的可读性	35
34. 保证表和索引文件合法而且有组织	36
35. 充分利用 Update Where Current of Cursor 和 Delete Where Current of Cursor	36
36. 使用游标时，利用多行获取、多行更新和多行 插入来使用 ROWSET 定位和获取	37
37. 了解锁定隔离级别	38
38. 了解 Null 处理	40
39. 编程时要考虑性能	42
40. 让 SQL 来处理	42
41. 使用 Lock Table	43
42. 考虑 OLTP 前端处理	44
43. 考虑使用动态可滚动游标	45
44. 利用物化查询表改善响应时间（只适用动态 SQL）	47
45. 结合 Select 的 Insert	49

46. 充分利用多行获取	50
47. 充分利用多行插入	52
48. 充分利用多行更新	53
49. 充分利用多行删除	55
50. 在 Select 子句中使用标量全选	55
51. 在动态 SQL 中充分利用 REOPT ONCE 和 REOPT AUTO, 在静态 SQL 中充分利用 REOPT VARS 和 REOPT ALWAYS	57
52. 标识易失表	58
53. 使用 ON COMMIT DROP 改进	59
54. 使用多个 Distinct	60
55. 充分利用反向索引扫描	60
56. 当心 Like 语句	61
57. 正确地设置聚簇索引	61
58. 必要时使用 Group By 表达式	63
59. 当心表空间扫描	64
60. 不要问你已经知道的信息	64
61. 注意查询中的表顺序	65
62. 使用左外联接而不是右外联接	66
63. 检查不存在的行	67
64. 使用存储过程	68
65. 不要只是为了排序而选择某一列	70
66. 尽可能限制结果集	70
67. 批量删除时充分利用 DB2 V8 的改进 DISCARD 功能	70
68. 充分利用 DB2 LOAD 工具完成批量插入	71
69. 注意视图、嵌套表表达式和公共表表达式的物化	72
70. 考虑压缩数据	74
71. 考虑并行性	75
72. 让 STDDEV、STDDEV_SAMP、VAR 和 VAR_SAMP 函数远离其他函数	76
73. 考虑使用 ROWID 数据类型 (V8) 或 RID 函数 (V9)	

直接访问行	77
74. 用真实统计和一定的数据测试查询以反映性能问题	78
75. 在 WHERE 子句中指定前导索引列	80
76. 尽可能使用 WHERE 而不是 HAVING 完成筛选	81
77. 尽可能考虑 Index Only 处理	82
78. DB2 V9 中表达式上的索引	83
79. 考虑 DB2 V9 Truncate 语句	84
80. 在子查询中使用 DB2 V9 Fetch First 和 Order by	85
81. 利用 DB2 V9 乐观锁定	85
82. 使用 DB2 V9 MERGE 语句	87
83. 了解 DB2 NOFOR 预编译选项	89
84. 考虑 Select Into 中使用 Order By	89
85. 尽量编写布尔项谓词	90
86. 编写传递闭包	90
87. 避免用 Order By 排序	91
88. 尽可能使用联接而不是子查询	92
89. 当心 Case 逻辑	92
90. 在 Order By 子句中充分利用函数	93
91. 了解你的 DB2 版本	93
92. 了解日期运算	94
93. 了解大容量插入选择	95
94. 了解 Skip Locked Data (V9) 避免锁定	96
95. 对输入流排序	97
96. 如果需要真正的唯一性, 可以使用 V8 Generate_Unique 函数	98
97. 了解声明临时表的新选项	98
98. 执行 Get Diagnostics 时需要注意	99
99. 适当地对 In 列表排序	100
100. 结合 Select 的 Update 和 Delete (V9)	100
101. 只在必要时执行 SQL 语句	101
102. 充分利用内存中的表	101

103. 避开通用型 SQL 语句	102
104. 避免不必要的排序	102
105. 了解表达式和列函数	103
106. 结合使用谓词时要注意	103
107. 为搜索查询增加冗余谓词	103
108. 充分利用改进的动态缓存 (V10)	104
109. 尝试当前提交来避免锁 (V10)	106
110. 尝试使用系统时态表获取历史数据 (V10)	107
111. 尝试使用业务时态表获取历史数据 (V10)	109
112. 了解分级函数 (V10)	110
113. 充分利用扩展指示符 (V10)	112
114. 得到更大的时间戳精度 (V10)	113
115. 尝试 Index Includes (V10)	114
116. 使用 With Return to Client (V10)	114
第 2 章 DB2 SQL 提示	116
1. 在 SQL 语句的最后使用 Optimize for 1 Row 语句	117
2. 为 SQL 查询增加 A.PKEY = A.PKEY 谓词, 这里 PKEY 等于表的主键列	118
3. 更换索引选择	119
4. 改变表处理顺序	121
5. 使用分布式动态 SQL	122
第 3 章 SQL 标准和原则	125
面向 COBOL 开发人员	125
面向所有 SQL 开发人员	129
第 4 章 SQL 程序走查	135
第 5 章 检查存在性	139
例 1	139
例 2	141

第 6 章	Runstats	143
第 7 章	查询初始调优步骤	146
附录 A	谓词重写示例	150
附录 B	DB2 SQL 术语	153

第 1 章

SQL 优化技巧宝典 100+

开发人员编写 SQL 代码时，为了确保得到更好的性能，需要采取一些措施，下面会列出其中最有效的一些做法。在关系型数据库管理系统（relational database management system, RDBMS）中，通常 90% 的运行时问题源于 10% 的应用问题。与以往相比，如今的 DB2[®] 优化工具在选择正确的访问路径来满足 SQL 需求方面已经更胜一筹，不过仍不能保证百分之百正确。目前的所有调优工作都与 CPU 使用、I/O 和并发直接关联。本章的大部分内容都会针对 SQL 编码，不过所有内容都与性能和调优有关。

本书中给出的技巧和提示是为开发人员提供的一般最佳实践，适用于大多数 RDBMS。在这些数据库系统中，可能有一些基于成本的优化工具，它们会查看所有可用的选项，构建一个 SQL 查询集，并尽其所能给出最高效的访问路径。对于不同的优化工具，可用的选项可能或多或少，不过一般而言，本书中的技巧可以应用于任何数据库管理系统。

程序编写不当或者 SQL 语句编写不正确，这些通常就是导致性能低下的罪魁祸首。看起来学习和掌握 SQL 语言很容易，不过在性能方面，DB2 优化工具处理 SQL 语言时存在很多低效问题。开发人员、测试人员、DBA 和业务分析师等都能从本书中获益，因为本书特别针对如何编写合适的 SQL 来提高效率提供了很多技巧。

本书的重点是增强开发人员对调优方法的认识，更多地了解 IBM[®]DB2 关系型数据库环境中如何改善性能和调优。本书条理清

楚、通俗易懂，提供了大量开发技巧、建议和 SQL 编码示例，所有这些的最终目的都是为了得到更好的性能。

这里的 SQL 编码示例会用到一些 DB2 示例表，这些示例表在各个 DB2 网站中都能找到，另外在所有 IBM DB2 手册中都有提到。

导致应用、程序或查询性能低下最常见的原因是什么？

- 数据库设计不当
- 应用设计不当
- SQL 查询结构不合理
- 编目统计不当 / 不充分
- 系统类资源（如缓冲区池、RID 池和记录日志）分配不当

以下是解决未能合理优化的一些方法：

- 用不同的方法重写查询
- 采用不同方式重写查询中的谓词
- 改变数据库对象的设计
- 改变数据库分区的设计
- 为已经存在的表增加一个分区策略
- 管理、修改或增强编目统计，以涵盖非标准统计
- 修改系统资源

要记住，DB2 优化工具只能做到：

- 接受 SQL 查询
- 验证语法
- 确认 DB2 编目有效性
- 检查查询中对象的编目统计
- 对所有不同的访问路径计算成本
- 选择成本最低的访问路径
- 计算分区值，确定满足 SQL 数据请求所需的目标分区

作为开发人员，你应该掌握这个领域的大部分内容，这很重要，这样才能在分析性能问题时对调优有所认识。优化工具只能处