

移动与嵌入式开发技术

WILEY

Inside Symbian SQL: A Mobile Developer's Guide to SQLite

Symbian SQL

编程揭秘



(美) Ivan Litovski
Richard Maynard

等著

杨小冬 译
彭明珍

symbian



清华大学出版社

Symbian SQL 编程揭秘

(美) Ivan Litovski 等著
Richard Maynard
杨小冬 彭明珍 译

清华大学出版社

北 京

Ivan Litovski, Richard Maynard, et al.

Inside Symbian SQL: A Mobile Developer's Guide to SQLite

EISBN: 978-0-470-74402-4

Copyright © 2010 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2010-5496

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Symbian SQL 编程揭秘/(美)里托斯基(Litovski, I.), (美)梅纳德(Maynard, R.) 等著; 杨小冬, 彭明珍 译.

—北京: 清华大学出版社, 2011.3

书名原文: Inside Symbian SQL: A Mobile Developer's Guide to SQLite

ISBN 978-7-302-24871-2

I. S… II. ①里… ②梅… ③杨… ④彭… III. 关系数据库—数据库管理系统—应用—移动通信—携带电话机 IV. ①TP311.138②TN929.53

中国版本图书馆 CIP 数据核字(2011)第 019484 号

责任编辑: 王 军 吴 乐

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京市清华园胶印厂

经 销: 全国新华书店

开 本: 185×230 印 张: 23.75 字 数: 489 千字

版 次: 2011 年 3 月第 1 版 印 次: 2011 年 3 月第 1 次印刷

印 数: 1~3000

定 价: 59.00 元

产品编号: 038098-01

作者简介

Ivan Litovski 于 2002 年加入 Symbian, 最初在 Java 团队工作, 随后陆续参与了系统库、持久性数据服务(Persistent Data Services, PDS)和产品创建工具团队的工作。从 1997 年开始, Ivan 就已经成为一名软件专家。他在 Java、嵌入式软件、信息安全、网络、可扩展服务器和 Internet 协议等方面具有非常丰富的经验。他乐于解决在研发过程中遇到的种种难题, 并享受工作带给他的乐趣。他获得过多项专利, 并在多家同行评审的国际杂志上发表过论文。他拥有塞尔维亚尼什大学(University of Nis)电子与电信专业理学学士学位, 并且是一位经过认证的 Symbian 开发人员。

Richard Maynard 自 2009 年 Nokia 收购 Symbian 以来, 就一直在 Nokia Devices R&D 的 Symbian 设备部门工作, 其职位是持久性数据存储架构师。Richard 于 2002 年加入 Symbian, 在系统库团队工作。在领导该团队将工作转移到 Symbian OS v9 中的平台安全以后, 他转而领导新成立的持久性数据服务团队, 创立该团队的初衷是在平台中引入一种全新的符合标准的 SQL 数据库。在加入 Symbian 之前, Richard 从事 3G 电信和雷达设计领域的相关工作。

James Aley 于 2006 年加入 Symbian, 以完成伦敦大学国王学院(King's College London)计算机科学学位学习计划所要求的为期一年的应用实习。2008 年大学毕业后, 他作为一名软件工程师就职于 Nokia, 一直到 2009 年秋天。他现在已经成为 Symbian Foundation 的一名技术专家, 主要研究生产技术领域。

James 基于 Symbian SQL 开发了一种适用于 Symbian 平台的基于元数据的搜索服务。该服务充分利用了 Symbian SQL 技术, 它可以使对持久性要求非常高的项目对开发人员而言变得非常简单, 而它在这方面的突出表现也成为备受推崇的典范。

Philip Cheung 于 2006 年加入 Symbian, 成为一名初级软件工程师, 他当时加入的是系统库团队。现在, 他已经转到 Nokia Devices R&D 的 Symbian 设备部门, 成为持久性数据服务团队的一名软件工程师。在就职于 Symbian 和 Nokia 的期间, Philip 参与了很多 Symbian SQL 项目的相关工作。Philip 于 2004 年获得曼彻斯特大学理工学院计算机与通信系统工程专业工学硕士学位。

James Clarke 于 2005 年加入 Symbian, 在个人信息管理团队从事联系人数据库引擎相关工作。在此之前, 他以研究员和数据库开发人员的身份在研究院工作了 10 年。现在,

James 已经转到 Nokia Devices R&D 的 Symbian 设备部门, 成为持久性数据服务团队的一名软件工程师。他拥有伦敦大学(University of London)计算机专业理学硕士学位。

Lorraine Martin 出生于格拉斯哥, 并在那里长大, 于 1997 年毕业于格拉斯哥大学, 获得计算机科学与数学专业一级荣誉学士学位。她曾就职于 Cisco Systems。

从 2002 年开始, Lorraine 移居到英格兰的伦敦, 并加入 Quickoffice 成为一名高级软件工程师。Quickoffice 是全球领先的手机办公软件提供商, 其软件适用平台包括 iPhone、Symbian 以及 BlackBerry (黑莓)。Lorraine 之前曾在位于伦敦的 Symbian 办事处工作了三年, 在此期间, 她作为技术主管参与了很多 SQLite 项目的相关工作。

Philip Neal 曾在牛津大学女王学院攻读古典与现代语言学, 并在曼彻斯特大学理工学院、兰斯卡特大学和埃塞克斯大学研究计算语言学。他是一位翻译家, 同时也是技术文档工程师。

Michael Owens 是 CENTURY 21 Mike Bowman, Inc. 公司的信息技术总监。作为一名热衷于开放源代码开发的开发人员, Michael 参与了很多出版物的编写工作, 其中包括 *Linux Journal*、*C++ User's Journal* 以及 *Dr. Dobbs Journal* 等。他曾在 O'Reilly 开源大会(Open Source Convention, OSCON)上做过 SQLite 主题演讲。

Michael 获得田纳西大学(University of Tennessee)化学工程学士学位。他曾先后就职于橡树岭国家实验室(Oak Ridge National Laboratory, ORNL)和 Nova 信息系统公司(Nova Information Systems), 分别从事流程设计工程师和软件开发人员工作。他是最早的 PySqlite 开发人员, PySqlite 是 SQLite 的 Python 扩展。

Martin Platts 于 2007 年加入 Symbian 数据库团队, 在该团队中, 他作为数据库特征鉴定项目的技术主管, 对 DBMS、SQL 以及其他第三方数据库的特征进行比较。在加入 Symbian 之前, Martin 已经拥有四年的 Symbian 工作经验, 在此期间, 他参与了 Nokia 以及 Sony Ericsson 等公司的相关项目。这些项目包括 DBMS 数据库、消息传递以及多媒体等。现在, Martin 是 Nokia 图形团队的一名项目经理。

序 言 一

您是否对在移动设备上使用结构化查询语言(Structured Query Language, SQL)和SQLite 的种种益处心存疑虑? 假设时光倒退 25 年, 如果当时我第一次读到这本书, 那么我一定会产生这种疑虑。20 年前, 我刚刚开始接触 SQL 语言, 当时, 我对 SQL 的整个理念有着巨大的怀疑。SQL 带给人的感觉是对简单的问题提出了一个复杂的解决方法。如果能够直接对各种磁盘文件(有时称为“文件堆”(pile-of-files)数据库)执行读写操作并使用简单的键-值对数据库, 那岂不是更易于操作?

在开始接触 SQL 语言之后的几年中, 我慢慢发现, 实际上 SQL 编程语言确实使很多编程任务变得非常非常轻松。随着各种应用程序越来越复杂并逐渐开始使用日趋丰富的内容, 能够以一种结构化的方式组织内容变得非常重要。文件堆和键-值对等方法迅速暴露出难于操作的问题。尽管 SQL 在初始阶段具有一定的复杂性, 但它确实会极大地简化复杂数据集的管理, 因此综合考虑, 最终的结果仍然是巨大的简化。SQL 使编程人员从单调乏味的低级输入/输出交互操作中解放出来, 同时让开发人员可以集中精力关注于应用程序中面向用户的各个方面。

精通 SQL 的编程人员通常可以用一个仅包含 10 行代码的 SQL 语句来代替成百上千行的低级输入/输出代码, 这样, 不但可以成倍地提高工作效率, 同时还能极大地提高代码的可维护性。使用 SQL 后, 编程人员还可以将数据存储设计的最终决定时间延迟到设计周期的后期, 或者在设计后期对存储策略做出重大变更, 从而将对代码造成的影响降至最低。此外, SQL 支持的事务性模型被证明是一种功能异常强大的抽象化事物, 有助于使应用程序变得更为健壮, 同时也更易于实现。

开始接触 SQL 语言之后的 10 年时间里, 在开发各种自定义工作站应用程序的过程中, 我一直坚持使用 SQL 语言。其中一个应用程序还使用了一种众所周知的商业 SQL 数据库引擎。该数据库运行得很好, 但其安装和管理非常困难, 并且是在缺乏技术熟练的数据库管理员的设置环境中使用的。于是我萌生了另外一种想法, 那就是开发一种新的 SQL 数据库引擎, 这种数据库引擎轻盈小巧, 并且操作成本非常低, 可以说是一种“零管理”数据库引擎。这种想法最终演变为“SQLite”。SQLite 的第一份源代码编写于 2000 年 5 月 29 日。

在接下来的几年中, 我和我的同事们开发的很多自定义项目都使用了早期版本的

SQLite。我将 SQLite 源代码发布到公共域中，这样其他的独立编程人员也可以从中获益，在那之后，SQLite 迅速成为 Internet 上的流行产品。随后，在 2005 年 10 月，Symbian 与我取得联系，并表达了想要在他们的移动操作系统中使用 SQLite 的兴趣。这次接触以后，我和 Symbian 的相关人员进行了一系列的会议讨论，在此期间，我迅速地发现，工作站与移动设备之间存在非常重大的差异。

阅读了这本书之后，您可能会了解到，移动设备的运行时环境与工作站的运行时环境可谓是大相径庭。从本质上来说，工作站具有无限的堆栈空间，无限的功能，并且永远不会出现内存分配问题。但对于移动设备来说，情况就大不相同了。诚然，SQLite 轻盈小巧、高效，而且管理开销趋近于零，正是这些优良品质在一开始吸引了 Symbian 对其产生兴趣，但实际情况却是，在真正应用于嵌入式设备之前，仍然需要对 SQLite 进行大量的细化工作。

在 Symbian 的指导和鼓励下，我和我的团队开始对 SQLite 进行改进，使其更加适合移动设备。我们对 SQLite 进行了增强，使其在内存分配出现故障时总是能够比较平稳的停止运行。我们大大降低了其对堆栈空间的要求，还添加了“共享缓存”模式，这种模式允许两个或更多应用程序共享一个通用的数据库缓存，从而可以降低对系统内存的要求。我们降低了一开始写入数据库所需的原始输入/输出数据量。此外，我们还增强了 SQLite 的总体可靠性和性能，使其能够满足 Symbian 提出的严格要求。经过艰苦卓绝的努力，最终开发出一种全新的“适合手机”的改进型 SQLite，也就是本书所要介绍的对象。

然而，Symbian 并不满足于 SQLite 团队仅仅做一些技术上的改进。他们还希望我们能够建立一种更为稳固可靠的业务模型。在问世后最初的 7 年时间里，SQLite 主要是由一些编程人员或开发人员自愿进行维护。偶尔会有支持合约提供给我们维护人员，但这些合约根本无法满足我们的生活所需。为了确保 SQLite 能够拥有一个更加稳定的未来，Symbian 管理层建议成立 SQLite 联盟。SQLite 联盟是一个由广泛使用 SQLite 的公司组成的群体，其中包括 Nokia、Mozilla、Adobe 以及 Bloomberg，联盟会向成员公司收取一定的年费，从而使三位 SQLite 开发人员(Dan Kennedy、Shane Harrelson 以及我本人)能够继续全职投入 SQLite 的相关工作，同时又能使 SQLite 源代码以免费开放的形式提供给所有用户。在两年的运营过程中，SQLite 联盟让我们的工作发生了翻天覆地的变化，使我们能够向 SQLite 中添加许多新的功能，同时能够改进代码质量，降低缺陷率，而又保持其轻盈小巧且操作高效的特质。

由于 SQLite 存在于公共域中，并且可以免费复制，因此我们没有办法跟踪 SQLite 使用的广泛程度。但保守的估计，仅通过计算已知使用 SQLite 的软件产品和小工具的数量，我们就可以得出这样的结论：当前部署的 SQLite 实例数量已超过 5 亿。这意味着，SQLite 的部署量可能比其他所有 SQL 数据库引擎部署量的总和还要多。我们开发的这个默默无闻的数据库引擎所取得的巨大成功，让我们这些 SQLite 开发人员惊喜不断，并感到无比的荣

耀。此外，我们非常感谢 Symbian 和 Nokia，是他们促使我们不断地对 SQLite 进行改进，而这种激励和敦促也会保证 SQLite 在将来取得更大的成功。

我真诚地希望 SQLite 能够带给您帮助。如果您对上面声明的使用 SQL 和 SQLite 的益处心存疑虑，我希望您能够坚持不懈地亲身尝试一下这些工具。我相信您一定会像我一样发现尽管 SQL 语言在初始阶段存在一定的复杂性，但对于具有复杂数据要求的应用程序的开发而言，它确实实能够大大简化您的工作。而且，我相信您还会发现，SQLite 与操作系统在 Symbian 平台(本书所讨论的对象)内的紧密集成使 SQLite 接口的代码编写成为一项非常轻松的编程任务。

D. Richard Hipp, SQLite 创始人
Charlotte, NC

序 言 二

我第一次接触 SQLite 是在 2005 年为 Symbian 平台选择新的关系数据库的期间。我与 Richard Maynard 以及他的团队合作研究了多种商业和非商业数据库。我坚持认为，应用程序软件组件需要复杂的高端数据库功能，而这些功能是当时现有的解决方案无法提供的，我非常希望能够选择一种正确的数据库。我们研究了每种产品提供的功能，并创建了电话簿和消息传递等应用程序所规划的典型用法基准。我们对广泛的条件进行了评估，其中包括响应时间、内存使用量、磁盘使用量和输入/输出操作。综合以上种种因素，SQLite 成为最佳的选择，它不但具有很高的性能，而且提供了足够的功能，完全可以满足我们规划的使用需求。选择使用 SQLite 并不是只有我们，诸如 Apple、Google 和 Palm 等多家公司也做出了同样的选择，即在其移动产品中使用 SQLite。

本书以一种独特的视角深入探究 Symbian SQL，这是一种将 SQLite 融入 Symbian 的客户端-服务器数据库。通过阅读本书，您可以认识到，要想充分利用 SQLite 数据库提供的性能和可伸缩性，您需要了解哪些内容。本书的作者都是积极参与过 Symbian 平台研发工作的技术专家，并且对 Symbian SQL 的实现和演化产生了积极的影响。

本书的前几章包含您期待看到的内容，即一个简介性质的初级教程，主要讲解 Symbian SQL API 的基本用法，解释用于数据库管理的关系模型，介绍 Symbian 数据库编程的一些重要概念并一个类一个类地说明 Symbian SQL API。此外，在第 7 章还对 SQLite 库的体系结构和内部原理进行了深入的分析研究，其中包括 Symbian 平台移植层的详细信息以及 Symbian 和 SQLite 作者确认的有关性能优化的技术信息。第 8 章介绍了一些不是很容易处理的方面，主要涉及如何设计高效的数据库架构以及优化 SQL 查询和 API 用法以便充分利用 Symbian SQL。本书中包含 Symbian SQL 开发人员和 SQLite 的作者提供的一些建议，此外还有 Symbian SQL 早期用户提供的非常难得的使用经验。

我认为本书的第 8 章是对 Symbian 社区最有价值的一部分。在 2007 年，我曾经作为技术架构师参与了某个团队的工作，该团队负责将联系人模型(Contacts Model，一种在电话簿应用程序中使用的应用程序引擎)从 DBMS 迁移到 Symbian SQL 中。为了设计数据库

架构,我们对多个可能的备选选项进行了基准测试,并与 Symbian SQL 开发人员进行了广泛而深入的探讨。如果当时我们能够读到本书第 8 章中提供的信息,那么一定会为我们节省大量的时间和精力以及最后时刻的性能调整工作。

在第 9 章中,您将了解到更多有关我们从 DBMS 进行迁移时的经历的信息,这部分内容由开发人员 James Clarke 为大家介绍,同时还提供了 3 个案例分析,讲解了 Symbian SQL 的潜力,包括由 Ivan Litovski 和 Martin Platts 创建的维基百科阐释。附录中提供了一些有用的建议,有助于解决遇到的各种问题,还包含了有关用于创建数据库的 SDB 工具的信息以及 Symbian SQL 错误代码汇总。

我从事 Symbian 平台的相关工作已经有很长时间了,并且一直持续到现在。我于 1998 年 9 月加入 Symbian Software Limited,职位是一名软件工程师,最初,我参与开发了适用于 Psion Series 5mx 的应用程序软件,随后开始研发早期的 Symbian 设备,例如 Ericsson R380。当时,所有的手机都使用单色显示屏。1999 年,在伦敦一家宾馆的套房里,我看到了彩屏 Nokia 9210 Communicator 的原型,当时我的激动心情丝毫不亚于我在 2008 年 6 月听到 Symbian Foundation 计划将代码公布于众的消息时的惊喜和兴奋。

现在,Symbian SQL 作为持久性数据服务程序包的一部分提供,而后者又属于 Symbian 平台。我是持久性数据服务程序包的所有者,该程序包是一种操作系统层程序包,可提供用于处理持久性数据存储的标准框架和库。程序包的源代码向所有用户公开,大家可以通过 Eclipse 公共许可证(Eclipse Public License, EPL)访问 developer.symbian.org 获取。

持久性数据服务具有一个专用的邮件列表,用于讨论新的程序包功能;一个用于讨论 OS Base Services 技术领域相关问题的论坛;一个用于报告缺陷的错误(Bug)数据库;一个包含正在开发的功能的已发布未完成工作列表;还包括一些有关我们的技术的 Wiki 文章。所有这些内容都可以通过我们为本书创建的 wiki 页面进行访问,网址为 developer.symbian.org/wiki/index.php/Inside-Symbian-SQL。

我们当前的未完成工作包括在 Symbian 3 中引入标准的 SQLite C API,以及做出一些变更,使 Symbian SQL 能够实现对称多处理(Symmetric Multiprocessing, SMP)。通过 SQLite C API,已经使用 SQLite 的应用程序创作者可以更加轻松地将其软件移植到 Symbian 平台上。对于我来说,性能和内存使用量是始终需要关注的两个方面,而且作为程序包的所有者,我必须确保 Symbian SQL 能够保持当前的系统质量,并在此基础上不断进行改进。

除了我们的程序包开发计划,Nokia 还提出了 Qt 主要功能捐赠倡议,这一倡议主要面向 Symbian 4,它会添加 QtSQL 模块,用于为基于 Qt 的应用程序提供一种全新的 SQLite

数据库访问方式。

Symbian SQL 的未来很大程度掌握在广大用户和开发人员的手中，当然，这其中也包括伙伴贡献者。Symbian 平台的开放式开发模型在各个层面都提供了对程序包的开发施加影响的机会。我个人非常支持并欢迎您加入我们的社区。

最后，我非常感谢 Ivan Litovski、Richard Maynard 以及另外几位作者，他们为使用 Symbian SQL 的广大用户提供了一份非常宝贵的参考资料。如果您正在使用 Symbian SQL，或者希望了解如何使用 Symbian SQL，那么赶快来阅读这本书吧！

Chris Dudding

持久性数据服务程序包的所有者，Nokia

作者致谢

谈到本书的编写过程，我首先想到的是，在单位度过非常满意的一天后高兴地回到家中。当时，我在 Symbian 的持久性数据服务团队工作。Symbian SQL 首次交付给我们的客户，而我们正忙于对其进行非常严谨的性能分析。除了团队成员以外，很少有人真正了解 Symbian SQL。我们积累了很多领域的知识，并希望通过所有可能的方式将这些知识与大家分享，我们尝试了 API 文档、工程文档和示例等多种途径。我认为图书是一种再好不过的实现知识共享的媒介，并开始慢慢构思。

非常巧合的是，没过几天，我就受邀与 George Sewell、Jo Stichbury、Satu Dahl、Richard Maynard 以及另外一些团队成员共同参加一个会议，讨论编写一部有关 Symbian SQL 的图书的相关事宜！显而易见，并不是只有我想到了这一点。当时，George Sewell 和 Lorraine Martin 已经就书的内容提出了一份建议，与本书中的内容差别不大。随后我们组织了一个团队，成员包括曾参与构建 Symbian SQL 的专家工程师，还有一些曾在 Symbian 中使用过 Symbian SQL 的专家工程师。

我首先要感谢的正是上面提到的这些人。George Sewell 对 Symbian SQL 的远见卓识，SQLite 联盟以及本书对 Symbian SQL 的成功起到了不可估量的作用。我还要感谢他所表现出来的友善以及有机会在这样一个出色的团队中工作。

如果没有 Jo Stichbury 和 Satu Dahl 的辛勤工作，我们绝不会走得如此之远。感谢他们在出版方面的丰富经验，在本书的整个出版过程中他们为我们提供了很多非常宝贵的指导意见。非常感谢他们所表现出来的耐心、专业知识以及艰苦卓绝的工作精神，没有他们，本书可能永远也不会出版。

我要特别感谢那些为本书提供了专家级内容的人，他们是：Richard Maynard、Lorraine Martin、Mike Owens、Martin Platts、Philip Cheung、James Clarke、James Aley 以及 Philip Neal。在这其中，有三个人需要特别提一下，那就是 Richard、Lorraine 和 Mike。Richard 领导团队完成了 Symbian SQL 的开发工作，并对营造积极向上、富于建设性的团队氛围做出了突出的贡献。Lorraine 对实现细节方面所具备的独一无二的专业知识、她的奉献精神、专注力以及努力工作的态度对本书产生了巨大的影响。Mike Owens 从他具有开创意义的著作

The Definitive Guide to SQLite 中提取出了一些非常有价值的内容，经过更新修改，然后放入 Symbian SQL 环境中。

我还要感谢 Richard Hipp 博士，是他开发了 SQLite 并将其免费提供给大家。Hipp 博士在初期阶段为我们的工作提供了有力的支持，并为本书题写了非常精彩的序言。

我的感谢还要献给那些参加了本书各章内容草稿审阅工作的顶级 Symbian 工程师，其中包括：Alex Dimitrov、Andrew Thaelke、Renzo Cherin、Lucian Piro 以及 Erhan Cetintas。同时需要特别感谢 Chris Dudding，他对本书的最终定稿提供了非常大的支持，并以其锐利的技术眼光对全书进行了审阅。

此外，还要感谢 John Wiley & Sons 的工作人员，正是有了他们艰苦卓绝的工作，才确保了本书从手稿打印成书并最终提供给广大的读者。感谢我们的开发编辑 Kenyon Brown，是他将本书所有内容整合到一起并校对通顺，同时也是他将我们所有人联系在了一起。

我尤其要感谢我亲爱的妻子 Aleksandra 和可爱的儿子 Boris，在长期的编写过程中，是他们为我提供了出色的后勤保障，并使我能够始终保持愉悦的心情。

最后，我要指出的是，对我而言，本书的编写过程是一段难以忘记的经历，我也从中学到了很多。我真诚地希望您能够喜欢这本书，并在工作中很好的运用。

Ivan Litovski

目 录

第 1 章 简介	1
1.1 从哪里开始	1
1.2 Symbian 术语和版本编号	2
1.3 关系模型和结构化查询语言(SQL)	3
1.4 什么是 Symbian SQL	3
1.5 运转良好的机器	5
1.6 工具和疑难解答	6
1.7 更多读物和资源	6
第 2 章 入门	9
2.1 从哪里获取 Symbian SQL	9
2.2 Symbian SQL API 概述	9
2.3 第一个数据库示例	10
2.3.1 项目设置	11
2.3.2 创建一个数据库	11
2.3.3 创建一个表	11
2.3.4 创建一个索引	12
2.3.5 插入记录	12
2.3.6 选择记录	13
2.3.7 更新一条记录	14
2.3.8 使用列绑定和流式处理 更新记录	15
2.3.9 删除一条记录	16
2.3.10 关闭数据库	16
2.4 小结	17

第 3 章 关系模型	19
3.1 背景	19
3.1.1 3 个组件	20
3.1.2 SQL 与关系模型	21
3.2 结构化组件	21
3.2.1 信息原则	21
3.2.2 逻辑级别不可侵犯性	23
3.2.3 逻辑级别剖析	23
3.2.4 元组	24
3.2.5 关系	25
3.2.6 表: 关系变量	29
3.2.7 视图: 虚拟表	30
3.2.8 系统目录	32
3.3 完整性组件	32
3.3.1 主键	33
3.3.2 外键	33
3.3.3 约束	35
3.3.4 空值	36
3.4 规范化	37
3.4.1 范式	37
3.4.2 第一范式	37
3.4.3 函数依赖	38
3.4.4 第二范式	39
3.4.5 第三范式	40
3.5 操纵组件	42
3.5.1 关系代数和关系演算	42

3.5.2	关系查询语言	43	4.5.13	深入介绍 NULL	98
3.5.3	SQL 的诞生	44	4.5.14	集合运算	102
3.6	关系的含义	45	4.6	修改数据	103
3.7	小结	45	4.6.1	插入记录	103
第 4 章	详细介绍您希望了解的 SQL 知识	47	4.6.2	更新记录	108
4.1	关系模型	48	4.6.3	删除记录	108
4.1.1	查询语言	48	4.7	数据完整性	109
4.1.2	SQL 的发展	49	4.7.1	实体完整性	110
4.2	数据库示例	50	4.7.2	域完整性	114
4.2.1	安装	51	4.7.3	存储类	118
4.2.2	运行示例	51	4.7.4	弱类型	121
4.3	语法	52	4.7.5	类型关系	124
4.3.1	命令	53	4.8	事务	130
4.3.2	文字	54	4.8.1	事务作用域	130
4.3.3	关键字和标识符	55	4.8.2	冲突解决	131
4.3.4	注释	55	4.8.3	事务类型	134
4.4	创建数据库	55	4.9	数据库管理	135
4.4.1	创建一个表	56	4.9.1	视图	135
4.4.2	修改一个表	57	4.9.2	索引	137
4.5	查询数据库	58	4.9.3	触发器	143
4.5.1	关系运算	58	4.9.4	附加数据库	149
4.5.2	运算流水线	59	4.9.5	清除数据库	150
4.5.3	对行进行筛选	63	4.9.6	系统目录	150
4.5.4	限制和排序	70	4.9.7	查看查询计划	151
4.5.5	函数和聚合	72	4.10	小结	152
4.5.6	分组	74	第 5 章	数据库概念	155
4.5.7	删除重复项	78	5.1	数据库类型	155
4.5.8	连接表	79	5.1.1	公共数据库	156
4.5.9	列名称和别名	87	5.1.2	专用数据库	156
4.5.10	子查询	89	5.1.3	安全共享数据库	156
4.5.11	复合查询	93	5.2	执行 SQL 语句	158
4.5.12	条件结果	96	5.2.1	数据库如何处理 SQL 语句	158

5.2.2	一次性执行	159
5.2.3	预定义语句	159
5.3	多客户端访问	163
5.3.1	事务	163
5.3.2	连接	164
5.3.3	锁定	165
5.3.4	连接、事务和锁定	167
5.3.5	预定义语句和锁定	167
5.4	小结	168
第 6 章	使用 Symbian SQL	169
6.1	数据库连接类	169
6.1.1	连接和断开连接	169
6.1.2	创建和删除数据库	171
6.1.3	附加其他数据库	172
6.1.4	执行 SQL 语句	173
6.1.5	复制和删除数据库文件	173
6.1.6	管理磁盘使用量	174
6.1.7	事务 API	177
6.1.8	检索安全策略	177
6.2	预定义语句类	177
6.2.1	准备和执行 SQL 语句	178
6.2.2	检索列值	179
6.2.3	参数化查询	183
6.3	使用可变长度数据对象	184
6.3.1	检索数据	184
6.3.2	检索大量的数据	185
6.3.3	流式处理数据	186
6.4	标量查询类	188
6.5	安全策略	190
6.6	小结	191

第 7 章	Symbian 上的 SQLite 内部 原理	193
7.1	为什么 SQLite 适合于 Symbian	193
7.2	SQLite 模块	195
7.2.1	公共接口	196
7.2.2	编译器	207
7.2.3	虚拟机	213
7.2.4	B-树模块	216
7.2.5	分页器	224
7.2.6	OS 接口	230
7.3	SQLite 配置选项	233
7.3.1	自动收缩	233
7.3.2	页面大小	234
7.3.3	最大缓存大小	235
7.3.4	数据库编码	235
7.3.5	SQL 语句编码	236
7.4	Symbian OS 上的 SQLite 优化	236
7.4.1	共享缓存	237
7.4.2	保留页面缓存内容	238
7.4.3	施加软堆限制	239
7.4.4	增加最大缓存大小	240
7.4.5	避免不必要的读取操作	243
7.4.6	避免不必要的写入操作	245
7.4.7	最大程度地降低使用日志 文件的开销	246
7.4.8	大容量存储和快速检索 二进制数据(BLOB)	248
7.4.9	后台数据库压缩	254
7.5	小结	256

第 8 章 性能优化	257
8.1 设计和开发过程中的注意 事项	257
8.1.1 确定关键用例	259
8.1.2 设定性能目标	259
8.1.3 度量性能	260
8.1.4 开发性能测试	261
8.1.5 跟进测试结果	261
8.2 优化简介	261
8.2.1 了解应用程序的操作	262
8.2.2 在优化之前确定性能瓶颈	262
8.2.3 避免过早优化	262
8.2.4 维护性能测试记录	263
8.2.5 一次执行一项优化	263
8.2.6 在正确的环境中进行优化	263
8.3 优化应用程序	263
8.3.1 应用程序层	264
8.3.2 数据模型	265
8.3.3 索引编制	266
8.3.4 SQL 查询	269
8.3.5 内存使用	270
8.3.6 系统优化	272
8.4 Symbian SQL 优化提示	274
8.4.1 数据组织	274
8.4.2 表达式	279
8.4.3 语句	286
8.4.4 索引编制	291
8.4.5 使用优化器	304
8.4.6 资源使用量	307
8.5 小结	317
第 9 章 使用 Symbian SQL: 3 个案例分析	319
9.1 联系人模型	319
9.1.1 DBMS 上的联系人模型	320
9.1.2 迁移的目标	321
9.1.3 Symbian SQL 中的联系人 模型	322
9.1.4 将来可能实现的改进	324
9.1.5 案例分析小结	324
9.2 Columbo 搜索服务	324
9.2.1 为什么选择使用 Symbian SQL	325
9.2.2 使用 SQLite 进行开发	326
9.2.3 案例分析小结	332
9.3 Wikipedia 演示应用程序	332
9.3.1 下载 Symbian Wikipedia 演示应用程序	333
9.3.2 在手机中存储 Wikipedia	333
9.3.3 设计架构(错误的方式)	333
9.3.4 设计架构(正确的方式)	334
9.3.5 显示文章列表	335
9.3.6 检索文章的内容	336
9.3.7 案例分析小结	336
9.4 小结	337
附录 A 疑难解答	339
附录 B SDB 数据库创建工具	351
附录 C Symbian SQL 错误代码	357
参考文献	359