

<<SQL Server 2005编程艺术>>

图书基本信息

书名：<<SQL Server 2005编程艺术>>

13位ISBN编号：9787115179357

10位ISBN编号：7115179352

出版时间：2008-7

出版时间：人民邮电出版社

作者：（美）麦查里克，（美）科纳利斯，（美）鲁贝尔克 著，颜 纫

页数：379

字数：663000

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 Server 2005编程艺术>>

### 内容概要

本书是一部关于SQL Server 2005 高级开发技术的权威著作，提供了详尽、全面的技术讨论，并给出了详细的应用实例。

书中前4章着重介绍SQL Server 所涉及的软件开发方法学、测试、异常处理和安全；第5章至第7章深入研究SQL Server 独具的高级特性，主要介绍加密、SQLCLR 和动态SQL；第8章至第11章从数据库体系结构的角度分析问题，深入研究具体的设计和实现问题，包括应用程序并发、空间数据、时态数据和图，这是本书最为复杂的内容，也是最有实际意义的内容。

本书可以作为软件开发人员，特别是数据库设计人员的参考手册，也可供计算机专业、信息系统专业的高年级本科生、研究生作为数据库课程的教材或者参考书。

## 作者简介

Adam Machanic，微软SQL Server MVP，著名的SQL Server专家。  
长期活跃于技术社区，是SQL Server Professional和TechNET等行业杂志的撰稿人，经常在各种技术会议上开设技术讲座、发表演讲。  
除本书外，他还是Pro SQL Server 2005的作者之一。

## &lt;&lt;SQL Server 2005编程艺术&gt;&gt;

## 书籍目录

第1章 数据库领域的软件开发方法学	1.1 架构回顾	1.1.1 耦合、内聚和封装
1.1.2 接口	1.2 核心问题：集成数据库和面向对象系统	1.2.1 该何去何从
1.2.2 对象—关系失配	1.3 ORM：一个问题多多的解决方案	1.4 把数据库操作设计为API
1.5 全面权衡	1.5.1 可测试性	1.5.2 可维护性
1.5.4 性能	1.5.5 步履蹒跚的远期功能	1.6 小结
2.1 黑盒测试与白盒测试简介	2.1.1 单元测试和功能测试	2.1.2 单元测试框架
2.1.3 回归测试的重要性	2.2 实现数据库测试过程和测试例程的指导准则	2.2.1 为什么说软件测试很重要
2.2.2 哪种测试重要	2.2.3 需要进行多少测试	2.2.4 管理层是否愿意接受软件测试
2.3 性能测试和数据库系统事件探查	2.3.1 获取基准度量	2.3.2 使用跟踪器及事件探查器进行查看
2.3.3 评估性能计数器	2.3.4 宏观分析	2.3.5 粒度分析
2.3.6 修正问题：仅仅关注明显的错误是否已经足够	2.4 SQLQueryStress性能测试工具介绍	2.5 小结
3.1 异常和错误	3.1.1 异常和错误	3.1.2 批处理程序级异常
3.2 批处理程序级异常	3.2.1 语句级异常	3.2.2 连接和服务级异常
3.2.3 语法分析和作用域解析异常	3.2.4 连接和服务级异常	3.2.5 XACT_ABORT设置
3.2.6 分析错误信息	3.2.7 SQL Server的RAISERROE函数	3.2.8 跟踪监控异常事件
3.3 异常处理	3.3.1 为什么要在T-SQL中处理异常	3.3.2 使用@@ERROR“处理”异常
3.3.3 SQL Server的TRY/CATCH语法	3.4 事务与异常	3.4.1 事务退出的神话
3.4.2 XACT_ABORT：神话（半）成真	3.4.3 TRY/CATCH和被诅咒事务	4.1 最低特权原则
4.1.1 在SQL Server中创建代理	4.2 使用Schema组织数据	4.3 使用EXECUTE AS实现基本模拟
4.4 所有权链	4.5 非链式特权提升	4.5.1 存储过程与EXECUTE AS
4.5.2 使用证书对存储过程进行签名	4.6 小结	5.1 要保护什么
5.1 要保护什么	5.2 加密术语：我们需要知道的部分	5.3 SQL Server 2005加密密钥体系
5.3.1 服务主密钥	5.3.2 数据库主密钥	5.4 SQL Server 2005数据保护
5.4.1 HashBytes（）	5.4.2 非对称密钥加密和证书加密	5.4.3 对称密钥加密
5.4.4 EncryptByPassphrase	5.5 隔离数据与DBA	5.6 性能优化设计
5.6.1 设计解决方案、定义问题	5.6.2 搜索加密数据	6.1 填充SQL/CLR鸿沟：SqlTypes库
6.1 填充SQL/CLR鸿沟：SqlTypes库	6.2 包装代码以提高跨层重用性	6.3 SQLCLR的安全性特征和可靠性特征
6.3.1 代码安全性需求	6.3.2 通过程序集引用有选择地提升特权	6.3.3 授予交叉程序集特权
6.4 利用SQLCLR加强Service Broker可扩展性功能	6.5 扩展用户自定义聚合	6.6 小结
7.1 动态T-SQL与ad hoc T-SQL	7.2 为什么要使用动态SQL	7.2.1 编译与参数化
7.2.2 自动参数化	7.2.3 应用程序级参数化	7.2.4 参数化与高速缓存的性能暗示
7.3 支持可选参数	7.3.1 通过静态T-SQL实现可选参数	7.3.2 实现动态化：使用EXECUTE
7.3.3 SQL注入	7.3.4 sp_executesql：一个更好的EXECUTE	7.4 动态SQL在安全方面的考虑
7.4.1 被引用对象的权限	7.4.2 接口规则	7.5 小结
8.1 从业务角度观察：进程产生冲突时会发生什么情况	8.1.1 SQL Server隔离级别概述	8.1.2 并发控制与SQL Server提供的隔离级别
8.2.1 在写入过程中使用悲观锁	8.2.2 应用程序锁：对悲观并发控制机制进行泛化	8.3 希望最好的情况发生：乐观的并发控制
8.4 包容冲突：多值并发控制	8.5 通过排队提高可扩展性	8.6 小结
9.1 用纬度和经度表示地球空间数据	9.1.1 设置示例数据	9.1.2 计算两点之间的距离
9.1.3 从一个位置点移动到另一个位置点	9.1.4 邻近搜索	9.1.5 边界框
9.1.6 查找距离最近的邻近点	9.1.7 动态边界框	9.1.8 结论
9.2 使用分层三角形网格表示地球空间数据	9.2.1 HTM的简化描述	9.2.2 实现HtmlID
9.2.3 Spatial数据库提供的函数	9.2.4 结论	9.3 其他类型的空间数据
9.3.1 三维数据	9.3.2 天文数据	

<<SQL Server 2005编程艺术>>

9.3.3 虚拟空间	9.3.4 将区域表达为多边形	9.4 小结	第10章 时态数据处理
10.1 表达的不仅仅是时间	10.2 SQL Server提供的日期/时间数据类型	10.2.1 日期输入格式	10.2.2 格式化输出日期
10.2.3 高效查询日期/时间列	10.2.4 日期/时间计算	10.3 使用日历表定义周期	10.4 设计时态数据存储方案及查询时态数据
10.4.1 处理时区	10.4.2 处理时间区间	10.4.3 为持续时间建立模型	10.4.4 管理双时态数据
10.5 小结	第11章 树、层次结构和图	11.1 术语：万物皆为图	11.2 基础知识：邻接表和图
11.2.1 为边加入约束	11.2.2 针对图的基本查询：给定一个结点，这个结点与其他哪些结点相连	11.2.3 遍历图	11.3 邻接表的层次结构
11.3.1 查询邻接表层次结构：基础知识	11.3.2 查找直接后代结点	11.3.3 在层次结构中向下级层次遍历	11.3.4 在层次结构中向上级层次遍历
11.3.5 插入新结点并重新定位子树	11.3.6 删除已有结点	11.3.7 为层次结构加入约束	11.4 持久保存具体化路径
11.4.1 查找下级结点	11.4.2 在层次结构中向上级层次导航	11.4.3 对具体化路径解决方案进行优化	11.4.4 插入结点
11.4.5 重新定位子树	11.4.6 删除结点	11.4.7 为层次结构加入约束	11.5 嵌套集模型
11.5.1 查找下级结点	11.5.2 在层次结构中向上级层次导航	11.5.3 插入结点	11.5.4 重新定位子树
11.5.5 删除结点	11.5.6 为层次结构加入约束	11.6 小结	

## 章节摘录

第1章 数据库领域的软件开发方法学 数据库应用程序开发是一种软件开发形式，人们应该以软件开发的方式看待数据库应用程序开发。

但是在开发小组讨论架构及测试计划时，往往把数据库部分当做次要部分来考虑——似乎很多数据库开发人员都认为，标准的软件开发最佳实践并不适用于数据库应用程序开发。

事实上，几乎每种应用程序都需要某种形式的数据存储，开发社区中的许多人都不只是持久化应用数据，还要创建数据驱动（data driven）的应用程序。

所谓数据驱动就是应用程序的行为根据数据的变化而动态变化，事实上可以用一个更好的术语来表达这个概念，即数据依赖（data dependent）。

由于应用程序与数据和数据库如此的密切相关，软件开发高手不仅仅要是能干的软件开发人员，毫无疑问还得是数据访问和管理方面的专家。

数据是应用程序中处于中心位置的、控制性的因素，它决定了应用程序对用户的价值。

没有数据，也就无需此应用程序。

本书的主要目的正是为了把Microsoft SQL Server开发人员重新拉回软件开发的范畴。

本书强调严格的测试、深思熟虑的架构设计，以及对数据彼此依赖性的谨慎考虑。

对这些方面的完美驾驭是软件开发专家的重要标志，而作为任何软件开发团队的核心，数据库专家当然不能缺少这个方面的专业技能。

本章将重点介绍软件开发概念和架构相关知识，因为它们完全适用于数据库应用领域。

其中有些主题在开发者社区中尚有极大争议，本书将尽量对各种观点并举，即使有些问题似乎已经有了权威答案。

不过我建议你真认真思考这些问题，而不是把我或者其他人的观点照搬照抄，因为软件架构领域于变万化，只有在对各种案例反复揣摩之后我们才可能奢望找到或许存在的“最佳”答案。

1.1 架构回顾软件架构是一个庞大、复杂的话题，主要是因为很多软件架构设计师往往喜欢把系统弄得尽可能的复杂。

而事实在于，编写一个优秀的软件，并不需要像很多系统设计师提倡的那样使用太多复杂的结构设计，真正高质量的设计往往只需理解并应用其中很少的基本原理就够了。

## <<SQL Server 2005编程艺术>>

### 媒体关注与评论

“ 罕见的深度，书中的所有内容都值得反复咀嚼并实际演练。

我向所有想探索SQL Server的读者推荐本书。

” ——Bob Beauchemin，微软SQL Server MVP “ 数据库方面的书太多了。

可是没有一本像本书这样讨论了软件工程、测试等诸多重要主题，包括我自己的收！

许多内容对我来说也是全新的，强烈推荐！

” ——JoeCelko，SQL标准委员会成员。

世界顶尖SQL专家，《SQL解惑》等经典著作的作者 市面上充斥着大量入门性质的SQL Server 2005图书，而许多重要的主题却很少涉及。

本书由几位世界级SQL Server专家撰写而成，他们从亲身开发经验出发，针对实际开发中会一再遇到的问题，讨论了包括微软公司联机文档在内的其他资料中都难以找到的一些重要主题。

作者将数据库视为软件开发生态系统中的一个有机组成部分，引导读者对SQL Server开发进行思考。

达到从未企及的深度。

通过书中的最佳实践和大量示例，读者可以掌握如何设计高性能、高可维护性的应用程序，从而大大提升自己的数据库开发水平。

## <<SQL Server 2005编程艺术>>

### 编辑推荐

《SQL Server 2005编程艺术》可以作为软件开发人员，特别是数据库设计人员的参考手册，也可供计算机专业、信息系统专业的高年级本科生、研究生作为数据库课程的教材或者参考书。Amazon全五星图书，众多专家盛誉推荐，深入揭示SQL Server开发思想，改变你对数据库开发的看法。

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>