

Microsoft
SQL Server 2005
Developer's Guide



Microsoft SQL Server 2005 开发指南

数据库专业人员的必备技能

- 构建、部署和管理企业级数据库应用程序
- 将 SQL Server 2005 集成到 Visual Studio 2005 和 .NET Framework
- 优化性能、实用性、可伸缩性和安全性

(美) Michael Otey 著
Denielle Otey 译
高 猛 译



清华大学出版社

Microsoft SQL Server 2005

开发指南

(美) Michael Otey 著
Denielle Otey
高 猛 译

清华大学出版社

北 京

Michael Otey Denielle Otey
Microsoft SQL Server 2005 Developer's Guide
EISBN: 0-07-226099-8

Copyright © 2006 by The McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by McGraw-Hill Education(Asia) Co., within the territory of the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. Violation of this Law is subject to Civil and Criminal Penalties.

本书中文简体字翻译版由美国麦格劳-希尔教育出版(亚洲)公司授权清华大学出版社在中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)独家出版发行。未经许可之出口视为违反著作权法,将受法律之制裁。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字:01-2005-5110

本书封面贴有 McGraw-Hill 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 13501256678 13801310933

图书在版编目(CIP)数据

Microsoft SQL Server 2005 开发指南/(美)奥蒂(Otey, M.), (美)奥蒂(Otey, D.)著;高猛译.

—北京:清华大学出版社,2007.1

书名原文:Microsoft SQL Server 2005 Developer's Guide

ISBN 978-7-302-14352-9

I.M… II.①奥…②奥…③高… III.关系数据库—数据库管理系统,SQL Server 2005

IV.TP311.138

中国版本图书馆 CIP 数据核字(2006)第 156410 号

责任编辑:王 军 李 阳

装帧设计:孔祥丰

责任校对:成凤进

责任印制:孟凡玉

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

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京市世界知识印刷厂

装 订 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:27.5 字 数:570 千字

版 次:2007 年 1 月第 1 版 印 次:2007 年 1 月第 1 次印刷

印 数:1~5000

定 价:49.90 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:019889-01

前 言

SQL Server 2005 的功能十分丰富,它为数据库开发人员提供了大量新工具和技术。编写本书的目的在于:使用 SQL Server 2005 中提供的新特性和功能帮助数据库开发人员和 DBA 提高工作效率,本书涉及 SQL Server 2005 开发技术的各个方面:从使用 T-SQL 的服务器端开发到使用 ADO、ADO.NET 以及 ADOMD.NET 的客户端开发。此外,还介绍了如何使用新的 SQL Server 2005 的 Notification Services、SQL Server Service Broker、Reporting Services 以及 SQL Server Integration Services 子系统来开发应用程序。

在 SQL Server 2005 中,开发管理方面已经发生了巨大的变化。因此,第 1 章首先对 SQL Server 2005 中新的开发和管理工具提供了指导性介绍。虽然 SQL Server 2005 包含了大量显著的变化,但仍有一些内容保持不变,其中一个事实就是 T-SQL 仍然是 SQL Server 2005 自带的开发语言,也是所有 SQL Server 2005 数据库开发的核心。第 2 章介绍了如何使用 SQL Server 2005 和 Visual Studio 2005 中提供的新 T-SQL 开发工具,如何创建 T-SQL DDL 和 DML 解决方案。第 3 章深入研究了 SQL Server 2005 中新的 SQL CLR 集成功能。.NET CLR 运行库与 SQL Server 2005 的集成是 SQL Server 2005 中最大的新变化之一。这一章介绍了如何创建和使用所有新的 SQL CLR 数据库对象,包括存储程序、函数、触发器、用户定义的类型以及用户定义的聚集。第 4 章介绍了新的 SQL Server Service Broker 子系统,它提供了构建异步应用程序的基础。SQL Service Broker 和 Notification Services 章节(第 5 章)概述了这个新的子系统,然后继续在示例应用程序中介绍它们的用法。ADO.NET 是 Microsoft 的核心数据访问技术,第 6 章也介绍了如何使用所有基本的 ADO.NET 对象创建功能强大的数据应用程序。XML 与相关数据库引擎的集成是 SQL Server 2005 中的另一个重大改进。第 7 章介绍了如何对有类型的和无类型的数据使用新的 XML 数据类型,如何创建 Web Services 为不同的平台集成提供 SQL Server 存储程序。虽然本书大部分着重介绍最新的 .NET 和基于 XML 的技术,但主要的 SQL Server 客户端应用程序都是使用 ADO 和 VB6 编写的。第 8 章介绍了构建 SQL Server 数据库应用程序的所有基本的 ADO 技术。SQL Server 2005 中最热门的两种技术是 Reporting Services 和面向终端用户的 Report Builder 报表设计器(report designer)应用程序。第 9 章深入研究了这些新特性,介绍了如何使用 Reporting Services 构建报表,

如何建立使用 Report Builder 的数据模型。第 10 章介绍了新的 SQL Server Integration Services 子系统。SQL Server Integration Services 完全替代了原有的 DTS 子系统,本章还介绍了如何使用设计器和 SSIS API 来构建和部署 SSIS 包。第 11 章介绍了如何使用新的 ADOMD.NET 数据访问编程架构为 Analysis Services 构建客户端 Business Intelligence 应用程序。SQL Server 2005 还引入了另一个全新的管理架构 System Management Objects(SMO),它替代了早期的 SQL Server 版本中使用的老式的 Distributed Management Objects(DMO)对象架构。在第 12 章中,您可以看到如何使用 SMO 来构建自定义的 SQL Server 管理应用程序。SQL Server 2005 还提供了一个全新的命令行界面 sqlcmd,它替代了原有的 isql 和 osql 实用程序。在第 13 章中,您可以看到如何使用 sqlcmd 工具开发管理脚本和数据访问脚本。最后,本书介绍了如何使用 SQL Profiler,它是解决应用程序性能问题以及调整数据访问查询的关键工具。

本书中提供的所有代码都可以从站点 www.teca.com 或 www.tupwk.com.cn/downpage 上下载使用。

SQL Server 2005 的设计目标

SQL Server 2005 比 20 世纪 80 年代 SQL Server 首次面世时面临了更多的挑战。当时,便于使用是首要的考虑因素,而且升级数据库来满足所有小型业务或部门的需求就已足够。但今天,SQL Server 不再是一个部门的数据库。而是一个充分成熟的企业数据库,能够提供数据访问功能。为了满足这些企业需求,Microsoft 设计的 SQL Server 2005 已经具有了高度的可伸缩性。此外,它还必须是安全的;必须能够方便地与其他平台集成;必须是一个高效的开发平台;而且必须提供很好的投资回报。

可伸缩性

可伸缩性曾经是 Microsoft SQL Server 备受指责的方面。由于它起初作为部门系统,并且在 Microsoft SQL Server 6.5 及早期版本中存在的一些局限,因此许多公司都没有将 SQL Server 视为企业数据库市场的一个合法成员。但是,现在所有一切都发生了变化。随着 SQL Server 7 版本的问世,Microsoft 在 SQL Server 平台的可伸缩性方面取得了长足的进步。使用分布式的分区视图,SQL Server 7 走到了 TPC-C 的前列,而且它获得的评价非常之高,以至于 SQL Server 7 对于 TPC(Transaction Processing Councils)决策来说是一个起到重要作用的因素,它将相互影响的 TPC-C 测试划分成群集(clustered)和非群集两部分。虽然 Microsoft 和 SQL Server 7 承认群集 TPC-C 的评价(score)(表示在多个系统中向外扩展(scale out)的能力),但对单个平

台上的向上扩展能力仍存有一些疑问。随着 Windows Server 2003 的启动以及 SQL Server 2000 Enterprise Edition 64 位的发布, 它也发生了一些变化, Microsoft 宣称 Microsoft SQL Server 第一次走到了非群集的 TPC-C 评价的前列。今天, 由于有了基于 Web 的应用程序的优势, 可伸缩性比以往显得更加重要。与传统的客户端/服务器和企业内联网应用程序不同的是, 您可以轻松地预知应用程序用户的数量, Web 应用程序对很多用户敞开大门, 并允许资源需求的快速变化。SQL Server 2005 包含了 Microsoft 的可伸缩性工作的所有积累, 并建立在使用分布式分区视图向外扩展的能力以及使用其 64 位版本向上扩展(scale up)的能力之上。它的 TPC-C 评价明确表明 SQL Server 2005 可以处理巨型数据库——甚至达到主机级的层次。而且 SQL Server 2005 的自调整能力可以让数据库快速优化自身的资源, 从而匹配使用需求。

安全性

虽然可伸缩性是通向企业级应用程序的跳板, 但安全性是真正获取企业信任所必须通过的一扇门。过去, SQL Server 与其他许多 Microsoft 产品一样, 受到了两个不同安全问题的困扰。这两个问题都与实现问题相关, 而与所有的实际代码缺陷无关。一家研究机构的研究显示, 有多达 5000 个 SQL Server 系统没有使用任何 sa 密码就部署在 Internet 上, 这使得所有危及这些系统上信息安全的入侵者都可以轻松访问。在 2002 年, SQL Slammer 病毒利用了 SQL Server 一个众所周知的隐患, Microsoft 以前也为这个隐患发布过一个补丁, 甚至将该补丁放入一个通用服务包中。对于第一种情况, SQL Server 实际上已经有了这个问题的答案, 它支持标准化安全和 Windows 身份认证, 用户只要执行一些非常基础的安全步骤即可。在第二种情况下, Microsoft 对某个已知问题已经生成一个补丁, 但该补丁的适用范围有限。而且, 这种情况下还存在另一个基础的安全问题, 即受到病毒侵袭的业务使得防火墙上某个应该关闭的端口没有关闭。

为了解决这些安全威胁, SQL Server 2005 按照 Microsoft 的新安全架构进行了设计, 该架构有时候称为 SD3, 这样的产品在设计、默认值以及部署方面都比较安全。对于 SQL Server 2005 来说, 这意味着产品的最初设计着重强调了安全性。继它们首创的 Trustworthy Computing 之后, Microsoft 从所有开发人员的全面的安全培训着手, 指导代码检查, 并对 SQL Server 2005 进行了全面的详细分析。此外, 集成到 SQL Server 2000 的 SP3 中的所有安全补丁都被引入到 SQL Server 2005 中。其次, 默认值的安全表示, 在安装该产品时, Microsoft 在安装过程中提供了安全的默认值。因此, 只要按照默认值进行安装就可以获得安全保障。例如, 以 sa 密码为例, 安装过程中会提示要为 sa 账户提供一个安全密码。虽然可以选择不使用任何密码继续安装, 但必须明确选择这种方法, 而且要对 Microsoft 的对话框警告(关于不使用密码的危险)作出应答。最后, Microsoft 为客户提供了一些工具和培训来帮助创建 SQL Server 2005 的安全部署。这里, Microsoft 提供的工具与 Microsoft Baseline Security

Analysis 类似，它可以扫描已知的安全隐患，此外还制订了一些白皮书，用户可以通过学习这些最佳实践来为各种不同的部署情况创建安全实现。

集成

在今天的公司计算环境中，很少会发生在一个异构设置中只安装一个供应商产品的情况。相反，多个不同的平台同时会执行各种不同的独立任务，公司所面临的主要挑战之一就是在这些不同的平台之间交换信息。SQL Server 2005 提供了一些不同的机制来为应用程序和平台之间的互操作性提供便利。对于应用程序互操作性，SQL Server 2005 支持行业标准 HTTP、XML 和 SOAP 协议。它还可以将存储程序提供为 Web 服务，并提供了 4 级的 JDBC 驱动程序，允许将 SQL Server 用作 Java 应用程序的后端数据库。对于平台互操作性，SQL Server 2005 对 Access、Oracle 和 IBM DB2 UDB 系统采用了全部重新设计的 Integration Services 以及异构数据库复制。

生产率

生产率是企业需求的另一个主要因素，而且这可能就是 SQL Server 2005 所取得的最大进步。这个新的 SQL Server 2005 版本将 .NET Framework CLR 集成到 SQL Server 数据库引擎中。这种新的集成允许使用任何 .NET 允许的语言(包括 C#、VB.NET 托管的 C++ 和 J#)来创建数据库对象，例如存储程序、触发器和用户定义的函数。在这个版本之前，SQL Server 的数据库编程能力只支持过程化的 T-SQL 语言。.NET Framework 的集成给它带来了完全面向对象的编程模型，它可以用于开发复杂的数据访问和业务逻辑程序。能够使用 .NET 语言编写数据库对象，也方便了在数据库和 n 层 Web 应用程序的数据访问层之间移动数据库对象。

虽然 SQL Server 2005 版本的关注重点是 .NET Framework，但 Microsoft 也在继续改进 T-SQL，并给它们的过程化语言带来了一些新功能，同时让开发人员和 DBA 放心，他们不会在未来放弃对 T-SQL 的支持。此外，SQL Server 2005 也从 DBA 的角度回答了有关生产率的问题。SQL Server 2005 重新设计了管理控制台，并将它集成到集成了开发环境的 Visual Studio .NET 中。现在所有对话框都是完全模式化的，允许 DBA 方便地在多个管理任务之间进行切换。

投资回报

今天，对于 IT 企业来说，其中一个主要挑战就是如何合理利用业务成本。这通常意味着花费较少的代价做更多的事情，并且 SQL Server 提供了一些工具能够帮助大多数业务使用它们已有的资本做更多事情。SQL Server 2005 远远不只是一个关系数据库，它紧密地集成了 Business Intelligence(BI)工具集(包括内置的 Analysis Services 和 Reporting Services)，比其他所有数据库平台能带来更多的价值。BI 为公司提供了分析数据和作出更好业务决策(使公司盈利并节约成本的决策)的能力。自

从 SQL Server 7 版本发布以来,由于集成了 OLAP Services(后来重新命名为 Analysis Services),SQL Server 已经成为 BI 市场的领先产品。

总之,SQL Server 2005 中的新特性给出了相当高的投资回报。如 Web 服务等特性为客户提供了更好的连通性,铺平了提高收益率的道路。同样,XML 集成提供了与业务成员的更好集成,从而提高了收益率。此外,.NET Framework 的包含内容(改进的管理工具和 Report Services 授权成员)使它们具备了更高的生产率。

Microsoft SQL Server 的简史

SQL Server 2005 是自 20 世纪 80 年代后期以来数据库服务器产品的最新版本。Microsoft SQL Server 起源于 1987 年的 Sybase SQL Server。在 1988 年,Microsoft、Sybase 和 Aston-Tate 将该产品移植到 OS/2 中。后来,Aston-Tate 放弃了 SQL Server 开发计划,而 Microsoft 和 Sybase 签署了一个联合开发协议,将 SQL Server 移植到 Windows NT 中。这个联合开发作业进而发展成 Windows NT 的 SQL Server 4.0 版本。在 4.0 版本之后,Microsoft 和 Sybase 分开进行 SQL Server 的开发。Microsoft 继续以 Windows NT 平台为目标开发未来的版本,而 Sybase 转移到以 UNIX 平台为目标的版本,这在今天仍然畅销。SQL Server 6.0 是完全由 Microsoft 开发的第一个 SQL Server 版本。在 1996 年,Microsoft 将 SQL Server 升级到 6.5 版本。在两年的开发周期之后,Microsoft 在 1998 年发布了有重大更新的 SQL Server 7.0 版本。SQL Server 7.0 在 SQL Server 中使用的底层存储器和数据库引擎技术方面包含了许多根本性的变化。又一个两年开发工作的成果 SQL Server 2000 在 2000 年 9 月公开发布。从 SQL Server 7.0 到 SQL Server 2000 的发展不只是一个进化,它与从 6.5 到 7.0 的发展所做的重大变化并不相同。相反,SQL Server 2000 进一步构建在 7.0 版本中建立的代码基础之上。从 SQL Server 2000 开始,Microsoft 在接下来的几年中开始对 SQL Server 的这个基础版本进行更新,首先开发了 XML for SQL Server Web Release 1,它添加了一些 XML 特性,包括接收作为 XML 文档的结果集的功能。接下来的几年,他们将这个 Web 版本重新命名为一个更简洁的名称 SQLXML 2.0,另外,它还添加了使用 XML updategram 更新 SQL Server 数据库的功能。接下来很快又发布了 SQLXML 3.0 的 Web 版本,它包含了将存储程序提供为 Web 服务的功能。两年之后,Microsoft SQL Server 发布了它的历史结晶——SQL Server 2005 版本。SQL Server 2005 使用了 SQL Server 7 中所建立的相同基础构造,增加了 SQL Server 2000 及其 Web 版本中介绍的所有特性,并集成了 .NET CLR 和一组强大的新的 BI 函数。下面的时间表概述了 SQL Server 的发展史:

- 1987 年 Sybase 发布了用于 UNIX 的 SQL Server。
- 1988 年 Microsoft、Sybase 和 Aston-Tate 将 SQL Server 移植到 OS/2。

- 1989 年 Microsoft、Sybase 和 Aston-Tate 为 OS/2 发布了 SQL Server 1.0。
- 1990 年 发布了支持 Windows 3.0 客户端的 SQL Server 1.1。
 - Aston-Tate 放弃了 SQL Server 开发。
- 1991 年 Microsoft 和 IBM 结束了 OS/2 的联合开发。
- 1992 年 发布了 16-bit OS/2 1.3 的 Microsoft SQL Server 4.2。
- 1992 年 Microsoft 和 Sybase 将 SQL Server 移植到 Windows NT。
- 1993 年 发布了 Windows NT 3.1。
- 1993 年 Microsoft 和 Sybase 为 Windows NT 发布了 SQL Server 的 4.2 版本。
- 1994 年 Microsoft 和 Sybase 对 SQL Server 的联合开发正式终止。
 - Microsoft 继续开发 SQL Server 的 Windows 版本。
 - Sybase 继续开发 SQL Server 的 UNIX 版本。
- 1995 年 Microsoft 发布了 SQL Server 的 6.0 版本。
- 1996 年 Microsoft 发布了 SQL Server 的 6.5 版本。
- 1999 年 Microsoft 发布了 SQL Server 的 7.0 版本。
- 2000 年 Microsoft 发布了 SQL Server 2000。
- 2001 年 Microsoft 发布了 XML for SQL Server Web Release 1(下载)。
- 2002 年 Microsoft 发布了 SQLXML 2.0(由 XML for SQL Server 重新命名而得)。
- 2002 年 Microsoft 发布了 SQLXML 3.0。
- 2005 年 Microsoft 在 2005 年 11 月 7 日发布了 SQL Server 2005。

目 录

第 1 章 开发环境	1
1.1 SQL Server Management Studio	2
1.1.1 SQL Server Management Studio 用户界面	3
1.1.2 SQL Server Management Studio 用户界面窗口	3
1.1.3 SQL Server 2005 管理工具	11
1.2 BI Development Studio	12
1.2.1 BI Development Studio 用户界面	13
1.2.2 BI Development Studio 用户界面窗口	13
1.3 小结	17
第 2 章 使用 T-SQL 开发	19
2.1 T-SQL 开发工具	19
2.1.1 SQL Server Management Studio	20
2.1.2 Visual Studio 2005	24
2.2 使用 T-SQL DDL 创建数据库对象	31
2.2.1 数据库	31
2.2.2 表	32
2.2.3 视图	36
2.2.4 同名对象	36
2.2.5 存储过程	36

2.2.6 函数	38
2.2.7 触发器	39
2.2.8 安全性	40
2.2.9 搜索存储器	42
2.3 使用 T-SQL DML 进行查询和更新	43
2.3.1 选择和连接	43
2.3.2 修改数据	57
2.3.3 错误处理	65
2.4 小结	65
第 3 章 开发 CLR 数据库对象	67
3.1 了解 CLR 和 SQL Server 2005 数据库引擎	67
3.1.1 CLR 体系结构	68
3.1.2 启用 CLR 支持	69
3.1.3 CLR 数据库对象组件	69
3.2 创建 CLR 数据库对象	72
3.2.1 CLR 存储过程	72
3.2.2 用户自定义函数	77
3.2.3 触发器	81
3.2.4 用户自定义类型	84
3.2.5 聚集	90
3.3 调试 CLR 数据库对象	94
3.3.1 .NET 数据库对象的安全	97
3.3.2 管理 CLR 数据库对象	98
3.4 小结	99

第4章 SQL Server Service**Broker 101****4.1 SQL Server Service Broker****体系结构 102**

4.1.1 消息 103

4.1.2 队列 103

4.1.3 合同 103

4.1.4 服务 103

4.1.5 对话 103

4.2 开发 SQL Service Broker**应用程序 104**

4.2.1 SQL Server Service

Broker DDL 和 DML 105

4.2.2 T-SQL DDL 105

4.2.3 T-SQL DML 105

4.2.4 启用 SQL Server

Broker 106

4.2.5 运用队列 106

4.2.6 SQL Server Service

Broker 应用程序示例 107

4.3 SQL Server Service Broker**激活 112**

4.4 对话安全 113

4.5 系统视图 113

4.6 小结 114

第5章 使用 Notification Services**开发 115****5.1 Notification Services 概述 116**

5.1.1 事件 116

5.1.2 订阅 117

5.1.3 通知 117

5.2 开发 Notification Services**应用程序 118**

5.2.1 定义应用程序 118

5.2.2 编译应用程序 118

5.2.3 创建通知订阅管理

应用程序 119

5.2.4 添加自定义组件 119

5.3 Notification Services**应用程序示例 119**

5.3.1 创建 ICF 文件 119

5.3.2 定义 ADF 文件 122

5.3.3 创建 Notification Services

应用程序 129

5.4 更新 Notification Services**应用程序 133****5.5 创建 .NET 订阅/事件****应用程序 134**

5.5.1 列出订阅 135

5.5.2 添加订阅 136

5.5.3 删除订阅 138

5.5.4 使用 .NET 激活数据

事件 138

5.5.5 使用 T-SQL 激活数据

事件 140

5.6 小结 141

第6章 使用 ADO.NET 开发**数据库应用程序 143**

6.1 ADO.NET 体系结构 144

6.2 ADO.NET 命名空间 145

6.3 .NET Data Provider 146

6.3.1 .NET Data Provider

的命名空间 147

6.3.2 .NET Data Provider

的核心类 148

6.4 ADO.NET 的 System.Data**命名空间中的核心类 150**

6.4.1 DataSet 类 150

6.4.2 DataTable 类 151

6.4.3	DataColumn 类	151
6.4.4	DataRow 类	152
6.4.5	DataRowView 类	152
6.4.6	DataRowManager 类	152
6.4.7	DataRelation 类	152
6.4.8	Constraint 类	153
6.4.9	ForeignKeyConstraint 类	153
6.4.10	UniqueConstraint 类	153
6.4.11	DataException 类	153
6.5	使用 .NET Framework Data Provider for SQL Server	154
6.6	使用 SqlConnection 对象	154
6.6.1	.NET Framework Data Provider for SQL Server 连接字符串的关键字	156
6.6.2	打开可靠连接	157
6.6.3	使用连接池	158
6.7	使用 SqlCommand 对象	160
6.7.1	执行动态 SQL 语句	161
6.7.2	执行参数化的 SQL 语句	163
6.7.3	执行带返回值的 存储过程	165
6.7.4	执行事务	167
6.8	使用 SqlDependency 对象	169
6.9	使用 SqlDataReader 对象	172
6.9.1	检索快速只向前的 结果集	172
6.9.2	读取 Schema-Only 信息	175
6.9.3	异步支持	176
6.9.4	多活动结果集	177
6.9.5	检索 BLOB 数据	178
6.10	使用 SqlDataAdapter 对象	181

6.10.1	填充 DataSet	181
6.10.2	使用 CommandBuilder 类	182
6.11	小结	185
第 7 章	使用 XML 开发	187
7.1	XML 数据类型	188
7.2	XQuery 支持	192
7.3	XML 数据类型的方法	195
7.3.1	Exist(XQuery)	195
7.3.2	Modify(XML DML)	196
7.3.3	Query(XQuery)	197
7.3.4	Value(XQuery, [node ref])	198
7.4	XML 索引	198
7.4.1	主 XML 索引	198
7.4.2	从 XML 索引	199
7.5	使用 For XML 子句	199
7.5.1	For XML Raw	199
7.5.2	For XML Auto	200
7.5.3	For XML Explicit	200
7.5.4	Type Mode	202
7.5.5	For XML Path	202
7.5.6	嵌套的 FOR XML 查询	204
7.5.7	内嵌的 XSD 模式生成	204
7.6	OPENXML	206
7.7	XML 批量载入	207
7.8	本地 HTTP SOAP 访问	208
7.8.1	创建 SOAP 端点	208
7.8.2	使用 SOAP 终端	211
7.9	小结	214
第 8 章	使用 ADO 开发数据库 应用程序	215
8.1	OLE DB 概述	216
8.2	OLE DB 体系结构概述	216
8.3	ADO	218

8.4	OLE DB 和 ADO 文件	219			
8.5	ADO 体系结构	219			
8.6	向 Visual Basic 中添加 ADO 引用	221			
8.7	在 Visual Basic 中使用 ADO 对象	223			
8.7.1	连接到 SQL Server	223			
8.7.2	使用 ADO Recordset 检索数据	237			
8.7.3	使用 ADO Connection 对象执行动态 SQL	258			
8.7.4	使用 ADO 修改数据	259			
8.7.5	使用 Command 对象 执行存储过程	266			
8.7.6	错误处理	269			
8.8	使用 ADO 的高级数据库 功能	270			
8.8.1	批处理更新	270			
8.8.2	使用事务	271			
8.9	小结	273			
第 9 章	Reporting Services	275			
9.1	Reporting Services 的 体系结构	276			
9.1.1	Reporting Services 组件	277			
9.1.2	安装 Reporting Services	278			
9.2	Report Server	285			
9.2.1	Report Server 处理器	286			
9.2.2	Report Server 扩展	287			
9.3	Report Manager	289			
9.4	Reporting Services 配置 和管理工具	290			
9.4.1	Reporting Services Configuration 工具	290			
9.4.2	Report Server 命令				
	提示工具	292			
9.5	报表编制工具	295			
9.5.1	Report Designer	296			
9.5.2	Report Model Designer	300			
9.5.3	Report Builder	303			
9.6	编程能力	304			
9.6.1	在 Window Form 中 使用 URL 访问	305			
9.6.2	使用 SOAP 集成 Reporting Services	306			
9.6.3	扩展	307			
9.6.4	RDL	307			
9.7	访问报表	307			
9.7.1	使用 URL 访问	308			
9.7.2	通过 Form POST 方法 进行 URL 访问	308			
9.8	报表编制	308			
9.8.1	开发步骤	308			
9.8.2	创建 Reporting Services 报表	309			
9.8.3	部署 Reporting Services 报表	313			
9.8.4	运行 Reporting Services 报表	313			
9.9	小结	315			
第 10 章	SQL Server Integration Services	317			
10.1	SQL Server Integration Services 概述	318			
10.1.1	Data Transformation Pipeline(DTP)	318			
10.1.2	Data Transformation Runtime(DTR)	319			
10.2	创建包	320			
10.2.1	使用 SSIS Import and Export Wizard	320			

10.2.2	使用 SSIS Designer	321	第 12 章	使用 SMO 开发	373
10.2.3	使用断点	337	12.1	使用 SMO	374
10.2.4	使用检查点	337	12.1.1	向 Visual Studio 中 添加 SMO 对象	374
10.2.5	使用事务	339	12.1.2	创建 Server 对象	375
10.2.6	包安全	339	12.1.3	使用 SMO 属性	376
10.3	部署包	340	12.1.4	SMO 属性集合	378
10.3.1	创建配置	340	12.2	SMO 的层次结构	381
10.3.2	使用包部署实用 程序	343	12.3	创建 SMO 示例应用程序	389
10.4	使用 SQL Server Integration Services API 编程	344	12.3.1	创建 Server 对象	391
10.5	小结	352	12.3.2	列出已注册的 SQL 系统	391
第 11 章	使用 ADOMD.NET 开发 BI 应用程序	353	12.3.3	连接到选定的 SQL Server 系统	391
11.1	Analysis Services 概述	354	12.3.4	列出数据库	393
11.1.1	XML for Analysis	354	12.3.5	列出表	393
11.1.2	Analysis Management Objects(AMO)概述	355	12.3.6	列出列	395
11.1.3	ADOMD.NET 概述	355	12.3.7	检索列属性	396
11.1.4	AMO 的层次结构	355	12.3.8	创建数据库	397
11.1.5	ADOMD.NET 对象 模型	356	12.3.9	复制表	398
11.2	使用 ADOMD.NET 构建 BI 应用程序	358	12.3.10	显示表的 T-SQL 脚本	401
11.2.1	为 ADOMD.NET 添加引用	360	12.3.11	SMO 错误处理	402
11.2.2	使用 AdomdConnection 对象	360	12.4	小结	403
11.2.3	使用 AdomdCommand 对象	363	第 13 章	使用 sqlcmd	405
11.2.4	使用 AdomdDataAdapter 对象	370	13.1	sqlcmd 组件	405
11.2.5	使用 CubeDef 对象	371	13.1.1	命令 shell	406
11.3	小结	372	13.1.2	命令行参数	406
			13.1.3	sqlcmd 扩展命令	410
			13.1.4	sqlcmd 变量	411
			13.2	开发 sqlcmd 脚本	412
			13.2.1	使用 Query Editor 开发 sqlcmd 脚本	412
			13.2.2	使用 sqlcmd 变量	413

- 13.2.3 使用 sqlcmd 脚本
 嵌套414
- 13.2.4 使用 sqlcmd 变量
 和 T-SQL 语句415
- 13.3 小结 415

- 附录 A SQL Profiler..... 417**
 - A.1 启动 SQL Profiler.....418
 - A.2 开始、暂停和停止跟踪422
 - A.3 重放跟踪422
 - A.4 Showplan 事件422



CHAPTER

1

开 发 环 境

本章主要内容：

- SQL Server Management Studio
- BI Development Studio

在涉及到服务器管理时，SQL Server 2005 已经为 DBA 做出了很多修改。用于管理先前 SQL Server 版本的管理工具已经被替代，并且添加了一些新的管理工具来帮助 DBA 更有效地与数据库服务器交互作用。在本章中，我们首先介绍新的 SQL Server Management Studio，它将 4 个原有的工具组合到一个集成环境中：Enterprise Manager、Query Analyzer、Profiler 以及 Analysis Manager。虽然每个工具都能让 DBA 实现各种具体的任务，但是，在这些工具之间切换并记住不同的界面和语法可能会产生不必要的麻烦。通过使用一个管理环境，DBA 可以更有效地集中管理服务器对象。本章第二部分将介绍这个新的 Business Intelligence(BI) Development Studio。BI Development Studio 是一种用于创建 Analysis Services 数据库、DTS 包和 Reporting Services 报表的集成开发环境。您可以在 BI Development Studio 中使用无连接模式将组件组织到项目和解决方案中，并在稍后部署解决方案。

1.1 SQL Server Management Studio

SQL Server Enterprise Manager 是 SQL Server 7 和 2000 版本的主要管理工具，它已经被新的 SQL Server Management Studio 替代，这个新工具也替代了 Query Analyzer 工具(SQL Server 7 和 2000 版本中的核心 T-SQL 开发工具)。SQL Server 2005 还包括其他一些管理工具，如新的 Administration Console、Database Tuning Adviser 和 Profiler。使用 Start | Programs | Microsoft SQL Server | SQL Server Management Studio 菜单选项可以打开 SQL Server Management Studio。图 1-1 显示了 SQL Server Management Studio。

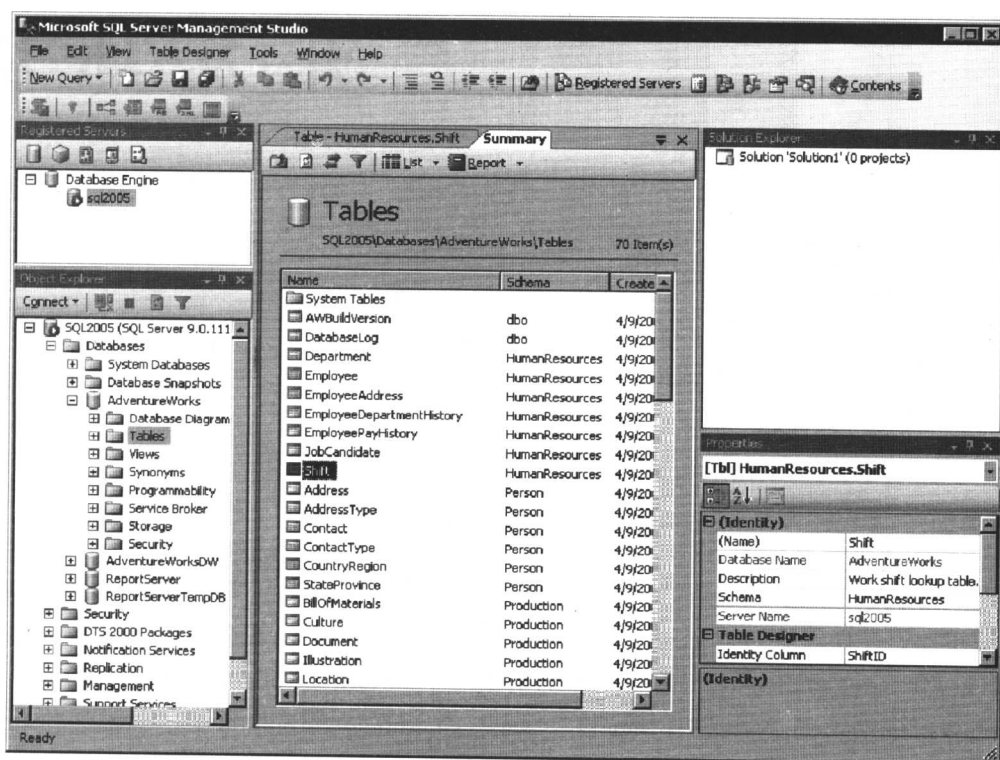


图 1-1 SQL Server Management Studio

SQL Server Management Studio 可以用于管理 SQL Server 2005 系统以及 SQL Server 2000 和 SQL Server 7 系统。但是，它不能用于管理 SQL Server 6.5 或更早的版本。您也可以使用先前的 SQL Server 7/2000 Enterprise Manager 来管理新的 SQL Server 2005 系统，但本书并不支持或推荐这种用法，因为这两种版本中的一些体系结构发生了变化。同样，原有的管理工具也不能访问 SQL Server 2005 中添加的任何新特性。SQL Server Management Studio 是管理 SQL Server 2005 和 SQL Server 7/2000