SQL Server 数据库经典译丛

SQL Server 2012 管理高级教程(第 2 版)

[美] Adam Jorgensen
Steven Wort
Ross LoForte
Brian Knight 著

宋沄剑　曹仰杰　译

清华大学出版社

北　京

Adam Jorgensen, Steven Wort, Ross LoForte, Brian Knight
Professional Microsoft SQL Server 2012 Administration
EISBN：978-1-118-10688-4

图书在版编目(CIP)数据

SQL Server 2012 管理高级教程/(美)乔根森(Jorgensen, A.) 等著；宋沄剑，曹仰杰 译. —2 版
—北京：清华大学出版社，2013.9
(SQL Server 数据库经典译丛)
书名原文：Professional Microsoft SQL Server 2012 Administration
ISBN 978-7-302-32927-5

Ⅰ. ①S…　Ⅱ. ①乔…　②宋…　③曹…　Ⅲ. ①关系数据库系统　Ⅳ. ①TP311.138

中国版本图书馆 CIP 数据核字(2013)第 145623 号

责任编辑：王　军　李维杰
装帧设计：牛艳敏
责任校对：成凤进
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　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 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：48.25　　字　　数：1234 千字
版　　次：2011 年 2 月第 1 版　　2013 年 9 月第 2 版　　印　　次：2013 年 9 月第 1 次印刷
印　　数：1~3000
定　　价：118.00 元

产品编号：

作 者 简 介

Adam Jorgensen(http://www.adamjorgensen.com)是 Pragmatic Works Consulting 公司的总裁，SQL Server 专业协会(Professional Association of SQL Server，PASS)的理事，也是一位享有盛名的演讲人、作家和执行导师。他关注于帮助公司发挥最大潜力，以他们原本没有想过的方式使用数据。对于 SQL Server 社区，Adam 做了很多贡献。他是 PASS 的理事，并且每年举办超过 75 场社区会议。他生活在佛罗里达州的 Jacksonville。Adam 已经独自撰写或参与撰写了 5 本关于 SQL Server、分析和 SharePoint 的图书。

Steven Wort 从 1993 年就开始使用 SQL Server，当时使用的是运行在 OS 2 上的 4.2 版本。他有超过 30 年的应用程序开发经验，曾在多个行业工作过。在 2000 年，Steven 加入了微软，担任 Systems Integration Engineering(SIE)团队的专家级工程师，并与人合办了多个关于 Windows 和.NET 调试的学习班。在 2004 年，他转到 SQL Server 产品小组研究 SQL Server 2005 的可扩展性。之后他转到 Windows 部门，负责扩展大型数据系统。现在，Steven 回到了 SQL Server 产品部门，担任 SQL Server Appliance Engineering Team 的首席工程师，负责 Microsoft Database Consolidation Appliance。Steven 参与撰写了几本关于 SQL Server 管理、故障排除和性能调整的图书。

Ross LoForte 是 Microsoft Technology Center Chicago 的技术架构师，主要致力于 Microsoft SQL Server 解决方案。Ross 有 20 年以上从事业务开发、项目管理以及设计 SQL 解决方案的经验。过去 11 年里，Ross 在 Microsoft Technology Center 工作，为微软最大的战略性客户设计企业级的关键任务 SQL Server 解决方案。Ross 在芝加哥 DePaul 大学讲授 SQL Server，还经常在 TechEd、SQL PASS、Gartner、TDWI 及 Microsoft 内部会议上演讲。Ross 也是 SQL Server 专业协会、芝加哥 SQL Server 用户组及其他 SQL Server 社区的积极参与者。

Brian Knight(SQL Server MVP、MCITP、MCSE、MCDBA)是 Pragmatic Works Consulting 公司的创始人，也是 BIDN.com、SQLServerCentral.com 和 SQLshare.com 的共同创始人之一。他管理着位于佛罗里达州 Jacksonville 的本地 SQL Server 用户组(JSSUG)。Brian 是多本技术杂志的专栏作家，并经常在 Jumpstart TV 上发表 Web 讲座。他共撰写了 12 本 SQL Server 相关的书籍。Brian 参加过许多会议，如 PASS、SQL Connections、TechEd、SQL Sundays 以及众多 Code Camp。可以在 http://www.bidn.com 找到他的博客。Brian 居住在佛罗里达州的 Jacksonville。

Robert C. Cain(http://arcanecode.com)是 Microsoft SQL Server MVP，获得了 BI 方面的 MCTS 认证，还是 Pragmatic Works Consulting 公司的高级顾问。他是 Plurasight Training 的技术贡献者，并且是 *SQL Server MVP Deep Dives*(Manning Publications，2009)第 1 卷和第 2 卷的合著者。作为一名颇受欢迎的演讲人，Robert 在 TechEd、SQL Rally 和 SQL Saturdays 上多次发表过演讲。Robert 在 IT 行业有超过 25 年的经验，他曾在多个领域工作过，包括制造业、电信业乃至核能工业。

Denny Cherry 是一名独立顾问，在使用 Microsoft SQL Server、Hyper-V、vSphere 和 Enterprise Storage 解决方案等平台方面有超过 10 年的经验。Denny 在系统架构、性能调整、安全性、复制和故障排除方面有专长。他获得了 SQL Server 2000 至 SQL Server 2008 的多个 Microsoft 认证，包括 Microsoft Certified Master，而且是连续几年的 Microsoft MVP。针对 SQL Server 管理和 SQL Server 如何与其他多种技术整合到一起，Denny 撰写了多本图书和几十篇技术文章。

Jose Chinchilla(http://sqljoe.com)是 Microsoft 认证的 SQL Server 数据库管理员和商业智能专家，在信息技术领域有超过 12 年的经验。Jose 成功地为零售业、制造业、金融业、医疗业、非营利组织以及政府部门架构并开发了数据仓库和商业智能解决方案。Jose 是 Agile Bay 公司的总裁(http://agilebay.com)，这是位于佛罗里达州 Tampa 的一家提供全面咨询服务的公司。Hose 经常在 SQL Saturday 和各地的 Code Camp 发表演讲。他也是 Tampa Bay Business Intelligence 用户组和 PASS 分会的主席。Jose 是一名积极的网民和博客作者，在社交媒体中很活跃，他的 Twitter 用户名是@SQLJoe。

Audrey Hammonds 是一名数据库开发人员、博客作者、演示者和作者。15 年前，她志愿加入一个新成立的数据库小组，目的是不再编写 COBOL 程序(从那以后她再也没写过 COBOL 程序)。Andrey 确信，如果人们能够停下脚步放松一下，享受眼前的风景，规范化他们的数据，世界会变得更好。她是 Jeremy 的妻子，Chase 和 Gavin 的妈妈，并且还养了三只猫——Bela、Elmindreda 和 Aviendha。她居住在佐治亚州亚特兰大市，她的博客地址是 http://datachix.com。

Scott Klein 把自己对 SQL Server 的热爱带到了 Microsoft。他不久前加入了 Microsfot 公司，成为一名 SQL Azure 技术推广者。在加入 Microsoft 之前，Scott 是 Blue Syntax 的合作创始人，这是一家提供 Azure 咨询和服务的公司。Scott 有深厚的 SQL Server 背景，从 SQL Server 4.2 开始，他在过去的 20 多年里一直使用 SQL Server。但是在知道 Azure 平台后，他的精力转移到了 SQL Azure 和 Azure 平台。他在美国和加拿大的 20 多个 Azure Boot Camp 上发表过演讲。Scott 没有忘记自己来自南佛罗里达，他管理着 South Florida SQL Saturday 和 South Florida SQL Server 用户组。Scott 还撰写了其他几本 Wrox 图书，包括 *Professional SQL Server 2005 XML*(John Wiley & Sons，2006)和 *Professional LINQ*(John Wiley & Sons，2008)，最近他还出版了 *Pro SQL Azure*(Apress，2010)。他还为其他几本书写过几个章节，并为 MSDN 杂志写过一些文章。Scott 有个很出色的妻子，4 个很可爱的孩子。

Jorge Segarra 原本是一名 DBA，现在是位于佛罗里达州 Jacksonville 的 Pragmatic Works Consulting 公司的 BI 顾问。他是一名 SQL Server MVP、博客作者(地址为 www.sqlchicken.com)、PASS 志愿者和 Regional Mentor，并且是网上社区博客项目 SQL University(www.sqluniversity.org)的发起人。

Gareth Swanepoel(http://mygareth.com)原本是一名系统管理员，后来改行为 SQL Server DBA。他在 IT 业做支持和管理工作超过 20 年了。他喜欢解决客户在数据仓库环境中部署 SQL Server 时遇到的复杂问题。他来自南非，现在和他美丽的妻子、5 岁的儿子和即将出生的女儿一起居住在佛罗里达州 Jacksonville 宁静的郊区。

技术编辑简介

Jason Strate 来自 Digineer 公司。他是一名拥有超过 15 年经验的数据库顾问。具体来说，他的经验包括设计和实现 OLTP 和 OLAP 解决方案，以及评估和实现 SQL Server 环境以遵循最佳实践、实现高性能和得到高可用的解决方案。他从 2009 年 7 月开始，就是 Microsoft SQL Server MVP。Jason 是一名 SQL Server MCITP，并参与设计 SQL Server 2008 和 2012 认证考试。

Denny Cherry 是一名独立顾问，在使用 Microsoft SQL Server、Hyper-V、vSphere 和 Enterprise Storage 解决方案等平台方面有超过 10 年的经验。Denny 在系统架构、性能调整、安全性、复制和故障排除方面有专长。他获得了 SQL Server 2000 至 SQL Server 2008 的多个 Microsoft 认证，包括 Microsoft Certified Master，而且是连续几年的 Microsoft MVP。针对 SQL Server 管理和 SQL Server 如何与其他多种技术整合到一起，Denny 撰写了多本图书和几十篇技术文章。

致　　谢

感谢家人和未婚妻 Cristina 在我写作这本书的过程中给予的支持。我也要为这个卓越的作者和技术编辑团队鼓掌致敬！从本书的作者简介和技术编辑简介中，你可以知道这些人是多么出色！特别要感谢 Brad Schacht，他的帮助对于完成本书中的一些章节至关重要。感谢我的两只小狗 Ladybird 和 Mac，冬天我在院里写作的时候，是它们让我的脚保持温暖(即使在佛罗里达，冬天也有些冷)。还要特别感谢 Brian Knight 和 Bob Elliot 把我拉到这个疯狂的写作比赛中。感谢 Wrox 团队让我们没有偏离航向，让这本书最终得以问世。你们对我和整个写作团队的支持是最重要的！

——Adam Jorgensen

感谢妻子 Anna 和女儿 Jennifer 在我写作这本书的过程中支持和照顾我。还要感谢 Microsoft Technology Center 的员工(包括主管 Adam Hecktman)的支持，他们使 Microsoft Technology Center Chicago 成为一个适合学习和工作的地方。另外，最重要的是要感谢 SQL Server 开发小组奉献了又一个功能丰富的版本：SQL Server 2012。最后，感谢技术编辑 Jason Strate 和 Wiley 的一些工作人员，包括 Victoria Swider 和 Robert Elliott，他们帮助这本书出版，让读者能够分享我对 SQL Server 的热爱。

——Ross LoForte

感谢每个为本书出版作出贡献的人。我亏欠妻子 Jenn 和孩子 Colton、Liam、Camille 以及新出生的儿子 John 很多。Jenn 总是忍受着我熬夜到很晚，而孩子们也对他们疲倦而繁忙的父亲保持着耐心。还要感谢吉尼斯黑啤酒和其他一些烈酒的酿造商，有了这些酒，我才能维持写技术图书的精力。感谢所有努力帮助其他人掌握技术的用户组领跑者。你们对技术的发展做出了很重要的贡献！最后，感谢声乐教练 Mike Davis 帮助我为这一季的美国偶像海选做好准备。这一次我不会让你失望！

——Brian Knight

首先，也是最重要的，我要感谢我可爱的妻子 Ammie。没有她的耐心和支持，我是什么也做不成的。我还要感谢我的女儿 Raven 和 Anna，谢谢你们容忍我如此繁忙，而且经常熬夜。

我不会健忘到忽视我在 Pragmatic Works 和 Pluralsight 的同事们对我的帮助。和这么多的技术专家们工作对我不只是一种挑战，更促使我不断地让自己在技术上取得越来越高的成就。

最后，我需要感谢 SQL 社区的所有人。谢谢你们阅读我的博客和图书，并在现场或者通过视频听我演讲。是你们给了我不断学习、讲解和分享知识的热情。

——Robert C. Cain

为了准备这本书中介绍的内容，我付出了大量的时间和精力。如果没有妻子和两个女儿的支持，没有她们对我无条件的爱和理解我对专业技术的追求，我是不可能付出这样的努力的。我还要特别感谢 Adam Jorgensen (http://adamjorgensen.com)让我有机会为这本书贡献一己之力，以及我的好朋友 Nicholas Cain (http://sirsql.net)，他为本书关于 SQL Server 群集的章节贡献了自己的专业知识。

——Jose Chinchilla

感谢 Adam Jorgensen 邀请我参与这本书的创作，感谢技术编辑 Jason Strate 改进我的工作，感谢 Wiley Publishing 的好人们有足够的耐心和毅力让我们走在正确的路上。特别要感谢我的家人们：Jeremy、Chase 和 Gavin。他们在我忙于工作的时候保持一切井井有条。感谢我在 Datachix 的同事 Julie Smith，谢谢你陪我海阔天空地聊天、给我打气还有一块喝葡萄酒。感谢 SQL 社区，谢谢你们这么出色，特别是我在亚特兰大的朋友们……你们是最棒的。最后，感谢我的外祖父 Bruce Bryant，他教会了我学无止境这个道理。

——Audrey Hammonds

首先，我要感谢我的妻子 Jessica，她的爱、理解和支持让我有能力完成写书这样的工作。感谢 Adam Jorgensen 和 Brian Knight 让我有机会参与这本书的写作。谢谢你们的支持和指导。感谢本书的编辑 Victoria Swider 和 Bob Elliot，抱歉给你们带来了不小的压力！最后，特别感谢整个 SQL Server 社区。这个社区就像个大家庭一样，没有每个人的彼此支持和分享知识，我不会成为今天这个样子。

——Jorge Segarra

感谢主耶稣基督给了我这样的才能，让我有机会参与这本书的创作。感谢我可爱的妻子 Jen 忍受着我的固执。感谢我的小儿子在我忙于及时完成工作时能够不需要爸爸陪着。感谢我在非洲的家人们让我来到希望之乡追逐梦想。妈妈、爸爸、Bernie、Kirsty 和 Tessie，这本书献给你们！

——Gareth Swanepoel

译 者 序

SQL Server 2012 自 2012 年 3 月发布以来，已经毋庸置疑成为数据库领域最有竞争力的产品之一。去年冬天我在西雅图参加微软数据库产品反馈和 Professional Association for SQL Server(PASS)峰会时，见识到了很多基于 SQL Server 和云端的大型解决方案，也见识了欧美 SQL Server 社区和生态环境有着让人震惊的强大。在 PASS 峰会期间，我有幸在 Wrox 出版社的活动中获得本书作为奖品，翻看此书时就已经惊叹本书的广度和深度。回国后，机缘巧合认识的清华大学出版社的李阳老师希望找业内的专业人士翻译此书的最新章节，我欣然接受。

本书由多位业内知名的专家共同撰写，几乎涵盖到了 SQL Server 管理领域的方方面面。本书不仅仅是 Step by Step 地教会你如何去做，而且还会对背后的原理进行剖析以便你更加深刻地理解 SQL Server 的内核和工作机理。由于书中的内容是最新的，并且有着足够的深度和广度，甚至有些内容在微软的 Book Online 上也没有对应的中文资料；因此在翻译的过程中，让自诩为 SQL Server 专家的我也颇为汗颜，深刻感受到自己知识的不完整。我在翻译过程中查阅了大量资料，力求翻译精益求精。至于本书中对原理的剖析部分，我推荐所有读者反复阅读，只有做到对深层次机理的理解，才能在管理和调优时游刃有余。

在这里我要感谢我的母亲在行文方面给予的指导，还要感谢清华大学出版社的李阳和李维杰编辑给我的莫大帮助，正因为有你们的严谨和专业，才能保证本书的高质量。

SQL Server MVP 宋沄剑
于北京

前　言

对于数据库管理员(DBA)、开发人员以及商业智能(BI)开发人员来说，SQL Server 2012在可扩展性、性能以及可用性方面有了很大的进步。在SQL Server上运行40TB数据库并不新鲜。以前，管理SQL Server只是DBA的工作，但随着SQL Server在众多小型公司的不断应用，许多开发人员也开始担任管理员和BI开发人员角色。另外，SQL Server新增的一些功能更以开发人员为中心，如果没有正确配置这些功能，就将导致性能低下。如今，SQL Server对数据工具、安全性和数据集成做了显著改进，简化了每个人的工作。本书是一本指导性的综合图书，可以帮助读者轻松地学习配置和管理SQL Server 2012。

本书读者对象

不管您是SQL Server管理员还是开发人员，都需要在某个时候承担DBA职责。开发人员经常需要在其工作站中安装SQL Server，并向管理员提供有关如何配置SQL Server生产服务器的指导。通常，他们负责创建数据库表和索引。负责支持生产服务器的管理员通常继承开发人员的数据库。

本书主要面向开发人员、DBA以及希望管理或正在管理SQL Server 2012系统及其商业智能功能(如Integration Services)的用户。这是一本专业书籍，意味着读者要具备有关如何查询SQL Server的基本知识并掌握一些SQL Server基本概念。例如，本书不会逐步演示如何创建数据库或通过向导安装SQL Server，但会介绍一些更高级的安装概念。尽管本书不介绍如何查询SQL Server数据库，但将讨论如何调整现有查询。

本书结构

本书采用上一版的基本结构，但是有一点主要改变：我们选择在各个领域中最出色的人组成本书的作者团队，让他们只撰写自己最擅长的地方。读者经常可以在各个主流会议上看到他们提供顶级的服务，例如性能调整、商业智能、数据库设计、高可用性、PowerShell甚至SQL Azure！这种方法让我们得以更加专注于本书的高质量，而且由于能够更好地与Microsoft内部人士进行交流，本书的内容也更加准确。本书的作者团队为了能够为读者提供最高质量的内容，通过Connect发表了数百个问题并最终确定了准确的答案。Connect是业界专家和SQL Sever MVP发布bug报告以及对Microsoft提议要添加的新功能进行投票的主要方法。这是改进产品的一个好地方。

本书仍然涵盖了上一版的所有精彩内容，同时添加了大量关于SQL Server 2012的新内容，帮助DBA利用SQL Server 2012的许多新功能简化工作。简言之，SQL Server 2012更加关注于提高效率、服务器的扩展性以及环境的性能，使用户在投入更少的时间、资源和人力的同时，

完成更多的任务。下面简要介绍各章的内容。

第 1 章：SQL Server 2012 体系结构——本书首先介绍 SQL Server 2012 在体系结构上的变化，并重点介绍 SQL Server 2012 的主要组件。

第 2 章：SQL Server 2012 安装最佳实践——本章介绍安装 SQL Server 2012 的不同方式，以及安装过程的最佳实践。

第 3 章：升级到 SQL Server 2012 的最佳实践——本章介绍如何升级到 SQL Server 2012 以及在升级时应该遵循的最佳实践。还介绍了如何选择最佳升级方法，以及升级的需求和优势。

第 4 章：数据库引擎管理与故障排除——本章重点介绍数据库引擎，以及在发生问题时如何解决它们。还介绍了数据库引擎的管理以及在解决问题时可以使用的工具。

第 5 章：自动化 SQL Server——本章介绍 SQL Server 2012 中的自动化功能，包括作业、PowerShell 和其他一些自动化方法。

第 6 章：SQL Server 2012 中的 Service Broker——Server Broker 是处理数据库内的消息传递的出色工具。本章介绍 Service Broker 的安装、操作和管理。

第 7 章：SQL Server 中的 CLR 集成——在 CLR 中可以结合使用 SQL Server 和.NET。本章介绍如何将.NET、CLR 和 SQL Server 集成起来，包括使用程序集和其他选项。

第 8 章：保护数据库实例——安全性对于数据库引擎至关重要。本章帮助你写出以及实现自己的安全计划。

第 9 章：变更管理——变更的管理对于稳定地运行数据库十分重要。本章介绍 SQL Server 中支持变更管理的功能。

第 10 章：配置服务器来调整性能——正确地配置服务器对于让应用程序和数据库的性能达到最优很重要。本章介绍对于系统性能很关键的存储选项、服务器选项和其他一些设置。

第 11 章：优化 SQL Server 2012——本章介绍的内容可帮助读者查看和分析性能。还介绍了可以提高 SQL Server 性能的配置选项。

第 12 章：监控 SQL Server——监控 SQL Server 对于确保实现自己想要的性能水平很关键。本章介绍应该监控 SQL Server 2012 的哪些地方以及可以使用哪些工具。

第 13 章：T-SQL 性能调整——编写有效且高效的 T-SQL 对于获得良好的应用程序性能和可扩展性很重要。本章介绍如何调整 T-SQL 以使其更加高效，还介绍了 SQL Server 引擎和内部机制如何读取和执行用户的查询，然后介绍了在什么地方可以调整这个查询，以及可以遵循哪些最佳实践。

第 14 章：创建数据库索引——索引对于获得良好的数据库性能很关键。本章讨论为数据库创建高效索引时需要考虑的事项以及可以采取的策略。

第 15 章：复制——复制是 SQL Server 中保持表和数据库同步以及支持应用程序的关键功能。本章介绍复制的类型，如何设置复制，以及每种方法的优缺点。

第 16 章：SQL Server 2012 群集——SQL Server 2012 对群集再次做了改进。本章介绍群集配置的建立、配置和测试。

第 17 章：备份与恢复——备份与恢复对于成功地实现连续运行计划以及获得良好运行效果很关键。本章介绍 SQL Server 中的备份与恢复选项，并就如何充分利用这些功能给出了一些建议。

第 18 章：SQL Server 2012 日志传送——本章介绍日志传送的建立、配置和管理。

第 19 章：数据库镜像——SQL Server 2012 中的可用性功能比以往任何版本都多。本章介绍使组织内的系统保持联机的新功能和原有功能。

第 20 章：Integration Services 管理和性能调整——集成是确保系统同步的关键。本章介绍 SQL Server 中这项功能的管理和调整。

第 21 章：Analysis Services 管理和性能调整——Analysis Services 是首选的联机分析处理(Online Analytical Processing，OLAP)产品，数据库管理员一定不能忽视。本章介绍这项功能的管理与调整。

第 22 章：SQL Server Reporting Services 管理——Reporting Services 通常由 DBA 管理。本章帮助读者解决使用 Reporting Services 时面临的挑战，无论读者的角色是什么，都可以掌握这项功能的使用。

第 23 章：SQL Server 2012 与 SharePoint 2010 集成——在 SQL Server 2012 中，SharePoint 占据的部分比以前的版本更大。本章介绍 SharePoint 2012 与 SQL Server 是如何集成的，了解这项知识后，读者就可以更好地与 SharePoint 团队交流，甚至自己承担一些 SharePoint 数据库管理责任。

第 24 章：SQL Azure 的管理和配置——本章介绍 SQL Server Azure，帮助读者掌握这个令人兴奋的新生云端平台。

第 25 章：AlwaysOn 可用性组——本章介绍 AlwaysOn 可用性组。使用这些可用性组可以将多个实例和服务器当作组来进行控制以及分配优先级，以灵活地控制故障转移和高可用性在自己的环境中是如何处理的。

软硬件要求

为了实践本书中的示例，读者需要安装 SQL Server 2012。如果要学习如何管理商业智能功能，还需要安装 Analysis Services 和 Integration Services 组件。读者需要一台满足 SQL Server 2012 最低硬件需求的计算机，还可能需要安装 AdventureWorks 和 AdventureWorksDW 数据库。访问这些数据库的指导参见本书配套网站上的 ReadMe 文件。

本书介绍的一些功能(特别是高可用性功能部分)需要 SQL Server 企业版或开发版。如果读者没有这些版本，那么可以使用标准版完成本书章节中的部分示例。

源代码

在读者学习本书中的示例时，可以手动输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 http://www.wrox.com 或 www.tupwk.com.cn/downpage 上下载。登录到站点 http://www.wrox.com，使用 Search 工具或书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

由于许多图书的标题都很类似，因此按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-1-118-10688-4。

在下载了代码后，只需要用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 http://www.wrox.com/dynamic/books/download.aspx 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫。当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们就会检查您的信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书的勘误表，可以登录 http://www.wrox.com，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的话题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 http://p2p.wrox.com 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

(1) 进入 p2p.wrox.com，单击 Register 链接。
(2) 阅读使用协议，并单击 Agree。
(3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit。
(4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目　录

第 1 章

SQL Server 2012 体系结构

本章主要内容：

- SQL Server 2012 的重要新增功能
- 新增功能与各类数据库管理员的关系
- SQL Server 体系结构概述
- SQL Server 的版本以及它们对数据库管理员的影响

SQL Server 2012 为组织及其开发人员、信息工作者和主管使用和整合数据的方式提供了一种新的视角。新增的大量功能和改进之处关注于使 SQL Server 进一步扩展到 SharePoint 中，改进自助服务选项，以及提升数据可视化、开发、监视和探索功能。本章不会深入介绍体系结构，但提供了帮助理解 SQL Server 如何运行的足够信息。

1.1 SQL Server 2012 生态系统

随着发布版本的增加，SQL Server 变得越来越庞大。本节将从整体上介绍 SQL Server 生态系统。现在越来越多地采用“生态系统”这个词，而不是“产品”，是因为 SQL Server 与其他产品和功能存在大量的交互，使它的性能、规模和可用性越来越好。SQL Server 2012 主要关注以下 3 个领域：

- 性能：改进的核心支持、列存储索引、压缩能力的增强以及 AlwaysOn 等功能使得 SQL Server 2012 成了最强大的 SQL Server 版本。
- 自助服务：借助于新的数据探索工具(如 Power View)，SQL Azure Business Intelligence(BI)、数据质量和主数据选项，以及 PowerPivot for SharePoint 的改进，使用户在任何时候都可以方便地访问数据，并且能比以往任何时候都要快速地查询和交付商业智能信息。

- 集成和协作：SharePoint 2010 中新集成了报表服务、PowerPivot 和声明验证(claim authentication)，这为 SQL Server 2012 版本中对自助服务的侧重提供了坚实的基础。现在报表服务已经包含在 SQL Azure 中，新的 BI 语义模型方法也扩展到了云。更多的功能还会在将来发布。

1.2 SQL Server 2012 的重要新增功能

取决于读者的角色和使用 SQL Server 的方式，SQL Server 2012 版本中有一些值得兴奋的新功能。本节将简要介绍你需要了解并实践的一些功能。其中的许多功能都可以快速上手，这对想要立即看到成果的读者来说是个好消息。

1.2.1 生产 DBA

生产 DBA 是公司的“保险单”，可以保证生产数据库不会宕机。如果数据库出现宕机，公司就要求生产 DBA 恢复数据库。生产 DBA 还确保了服务器以最优的方式运行，并促进数据库从开发转入 QA(Quality Assurance，质量保证)，再到生产。生产 DBA 执行的其他任务包括：

- AlwaysOn：一种可用性功能，包括可用性组和模仿应用程序的行为以组的形式进行数据库故障转移。这种功能包括新的可读辅助服务器，是一种显著的增强。
- FileTable：额外的基于文件的数据存储。
- 扩展事件：SQL Server 2012 中一种新增的内置功能，提供了轻量级的、覆盖范围广的跟踪功能。
- 功能和稳定性得以改进的SQL Server Management Studio(现在包含在Visual Studio 2010 shell 中)。
- 分布式重播能力。
- 改进的调试功能，包括支持表达式和断点验证。
- 列存储索引，用于优化大数据卷。
- 针对超大数据库改进的统计算法。
- 改进的压缩和分区能力。

1.2.2 开发 DBA

自 SQL Server 2000 发布后，专职的生产 DBA 出现了一种趋势，该角色与开发 DBA 的角色相合并。不过，这一趋势由于一些法案的存在而发展得较慢，如 Sarbanes-Oxley 法案(该法案要求区分开发更改人员和实现更改人员的职责)。在大型的组织中，生产 DBA 可能归运营部门(由网络管理员和 Windows 支持人员构成)。将生产 DBA 放到开发小组中，就无法实现一些规章原因所要求的职责分离。

开发 DBA 在组织中也扮演着非常传统的角色。他们大多戴着“开发人员”的帽子，是开发人员眼中的数据库专家和代表。这类管理员确保所有存储过程都以最优方式编写，数据库在物理上和逻辑上都正确建模。他们还编写迁移过程来将数据库从一个版本升级为下一个版本。开发 DBA 通常不会在凌晨两点接到电话，这与生产 DBA 不同，后者可能因为备份失败或其他类似问题在任何时间接到电话。开发 DBA 应该会对新版本中的以下地方感到兴奋：

- 新的 T-SQL 和空间数据功能。
- SQL Server Data Tools：集成到 Visual Studio 中的新的 T-SQL 开发环境。
- 新的 DAX 表达式语言，像 Excel 一样便于使用，具有多维数据处理能力。
- Analysis Services 的新的表格模型：内存优化的 OLAP 技术展现出一种快速取得价值的形式。

开发 DBA 通常向开发小组作汇报。他们接收来自业务分析师或其他开发人员的要求。从传统意义上讲，开发 DBA 不应该获得生产数据库的修改权限。不过，他们可以只读访问生产数据库以在升级阶段进行调试。

1.2.3　商业智能 DBA 和开发人员

商业智能(BI)DBA 是随 SQL Server 不断增长的能力而发展起来的新角色。在 SQL Server 2012 中，BI 发展成为许多业务必不可少的一个非常重要的功能集。BI DBA 或开发人员是这些功能的专家。SQL Server 2012 中包含了许多新的 BI 功能，包括对 Reporting Services 集成进行了增强，提供了 Power View 等数据探索工具，并使得 PowerPivot 比以往任何时候更易于使用。另外，SSAS 的新的表格模型能够向 SharePoint 创建新的、类似于 PowerPivot 的“内存优化的”BI 项目，以便大量用户使用。

开发 BI DBA 主要关注最佳实践、优化和 BI 工具集的使用。在小型组织中，他们可能创建 SSIS 包，为用户执行提取、转换和加载(Extract Transform and Load，ETL)过程或报表。在大型组织中，开发人员创建 SSIS 包和 SSRS 报表。开发 BI DBA 被咨询有关 SSIS 程序包和 Analysis Services(SSAS)多维数据集的物理实现的内容。开发 BI DBA 可能有下列职责：

- 有关 Analysis Services 多维数据集和解决方案的建模和咨询。
- 使用 Reporting Services 创建报表。
- 使用 Integration Services 创建 ETL 及提供相关咨询。
- 开发将发送至生产 DBA 的部署包。

这些职责，再加上下面的新功能，使得 BI 相关人员的工作十分有吸引力：

- 使用 Power View 和 PowerPivot 快速发现数据。
- 托管的自助式 BI，能够使用 SharePoint 和 BI 语义模型。
- 使用 Data Quality Services 和 Master Data Management 得到可信而一致的数据。
- 使用 Parallel Data Warehouse 和 Reference Architectures 实现健壮的 DW 解决方案。

1.3　SQL Server 体系结构

许多人只是将 SQL Server 用于传统的用途：存储数据。SQL Server 的这个版本的关注点是扩展 SQL Server 2008 R2 中引入的功能，主要是自助的商业智能和 SharePoint 功能发布。SQL Server 2012 中的另外一些功能不只支持、而且鼓励用户跳出只使用 SQL Server 存储数据的惯性思维，可以把这个版本作为完整的数据策略的核心。新的 Power View 和 PowerPivot 等工具可以快速地在 SQL Server 之上集成，从而为 SQL Server 和其他系统的数据提供易用的用户界面(UI)。本节将介绍 SQL Server 2012 中的主要文件类型、文件管理、SQL Client 和系统数据库，还会概述架构、同义词和动态管理对象。最后，本节还讲解了 SQL Server 2012 中新的数据类型。

1.3.1 数据库文件和事务日志

数据库和事务日志文件的体系结构自发布以来并未改变。取决于具体的类型，数据库文件有两个主要的目的。数据文件保存数据、索引和数据库内的其他数据支持结构。日志文件保存已提交事务的数据，以保证数据库内的一致性。

1. 数据库文件

数据库由多个文件组构成。每个文件组可能包含一个或多个物理数据文件。文件组用于简化文件集合的管理工作。数据文件被划分到 8KB 的数据页中，这些数据页是 64KB 的区段的一部分。可通过 T-SQL 命令 create/alter index 的填充系数选项指定每个数据页的填充情况。在 SQL Server 2012 企业版中，如果单个文件被破坏，仍可使数据库部分联机。在这种情况下，DBA 可使其余文件联机并进行读写，如果用户尝试访问数据库中的脱机部分，就会接收到错误。

在 SQL Server 2000 及更早版本中，一行最多可写 8060 字节。但有一些例外：text、ntext、image、varchar(max)、varbinary(max)以及 nvarchar(max)列最多可达 2GB 并可单独管理。从 SQL Server 2005 开始，8KB 限制只适用于那些定长列。定长列的合计值以及其他列类型的指针仍必须小于每行 8060 字节。不过，每个变长列可能达到 8KB，因此行的总尺寸可能大于 8060 字节。如果实际行尺寸超出 8060 字节，可能导致性能降级，因为现在逻辑行必须跨多个 8060 字节的物理行。

2. 事务日志

事务日志用于确保所有提交的事务在数据库中持久保存并可恢复(如回滚或时间点恢复)。事务日志是预写式(write-ahead)日志。在对 SQL Server 中的数据库作出更改时，数据会写入日志，然后需要更改的页会被加载到内存中(具体来说，就是加载到缓冲池的写缓冲区)。然后更改将写入这些页，使其成为脏页。到了检查点时，这些脏页会被写入到磁盘中，从而使它们再次成为干净页，不再需要作为写缓冲的一部分。这就是你在长时间运行的事务中看到事务日志显著增长的原因(尽管恢复模型很简单)。在第 17 章中将对此作更详细的介绍。

1.3.2 SQL Native Client

SQL Native Client 是 SQL Server 2005 自带的一种数据访问方法，在 SQL Server 2012 中得到增强，由 OLE DB 和 ODBC 用于访问 SQL Server。它通过将 OLE DB 和 ODBC 库组合成一种访问方法，简化了对 SQL Server 的访问。这种访问类型展示了 SQL Server 的一些新功能：

- 数据库镜像
- AlwaysOn 可读辅助路由
- 多活动结果集(Multiple Active Result Set，MARS)
- 快照隔离
- 查询通知
- XML 数据类型支持
- 用户定义的数据类型(User-Defined Data Type，UDT)
- 加密

- 执行异步操作
- 使用大型值类型
- 执行批量复制操作
- 表值参数
- 大型 CLR 用户定义类型
- 密码过期

可以在其他数据层(如 Microsoft Data Access Component，MDAC)中使用这些新功能中的一部分，但需要做更多的工作。MDAC 仍然存在，如果不需要 SQL Server 2008\2012 的一些新功能，可以使用 MDAC。如果开发基于 COM 的应用程序，那么应使用 SQL Native Client；如果开发托管代码应用程序(如使用 C#)，那么应考虑使用 SQL Server .NET Framework 数据提供程序(它非常健壮且包括 SQL Server 2008\2012 的功能)。

1.3.3　标准系统数据库

SQL Server 中的系统数据库很重要，大部分时候都不应修改它们。唯一例外是 model 数据库和 tempdb 数据库。model 数据库允许部署更改(如存储过程)到任何新创建的数据库，而更改 tempdb 数据库的原因则是为了帮助扩展数据库以承担更多的负载。下面将详细介绍标准系统数据库。

如果某些系统数据库被篡改或破坏，那么 SQL Server 可能无法启动。master 数据库包含了 SQL Server 保持联机所需的所有存储过程和表。

1. Resource 数据库

SQL Server 2005 添加了 Resource 数据库。这个数据库包含了 SQL Server 运行所需的所有只读的关键系统表、元数据以及存储过程。它不包含有关用户实例或数据库的任何信息，因为它只在安装新服务补丁时被写入。Resource 数据库包含其他数据库逻辑引用的所有物理表和存储过程。该数据库的默认位置为 C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Binn，每个实例只有一个 Resource 数据库。

路径中的 C:假定了一种标准设置。如果你机器的设置与此不同，那么可能需要改变此路径来进行匹配。另外，.MSSQLSERVER 是实例名。如果你的实例名也与此不同，那么在路径中使用你自己的实例名。

在 SQL Server 2000 中，当升级到新的服务补丁时，需要运行很多且很长的脚本以删除并重新创建系统对象。这个过程需要很长时间，并且新创建的环境不能回滚到安装服务补丁之前的版本。在 SQL Server 2012 中，升级到新服务补丁或快速修正时，将使用 Resource 数据库的副本覆盖旧数据库。这使得用户可以快速升级 SQL Server 目录，还可以回滚到前一个版本。

通过 Management Studio 无法看到 Resource 数据库，并且永远不应修改它，除非 Microsoft 产品支持服务(Microsoft Product Support Services，PSS)指导用户进行修改。在特定的单用户模式条件下，可以通过输入命令 USE MSSQLSystemResource 连接该数据库。通常，DBA 在连接到任何数据库的同时对它执行简单查询，而不必直接连接 Resource 数据库。Microsoft 提供了一些函数来实现这种访问。例如，如果在连接到任何数据库时运行下面的查询，将返回 Resource 数据库的版本及其最后一次升级的时间：

```
SELECT serverproperty('resourceversion') ResourceDBVersion,
serverproperty('resourcelastupdatedatetime') LastUpdateDate
```

不要将 Resource 数据库放在加密或压缩的驱动器中，这样做可能导致升级问题或性能问题。

2. master 数据库

master 数据库包含有关数据库的元数据(数据库配置和文件位置)、登录以及有关实例的配置信息。如果这个重要的数据库丢失，那么 SQL Server 将不能启动。例如，通过运行下列查询(它将返回有关服务器上的数据库的信息)，可以查看存储在 master 数据库中的一些元数据：

```
SELECT * FROM sys.databases
```

Resource 数据库和 master 数据库之间的主要区别在于 master 数据库保存用户实例特定的数据，而 Resource 数据库只保存运行用户实例所需的架构和存储过程，而不包含任何实例特定的数据。

尽量不要在 master 数据库中创建对象，如果在其中创建对象，那么可能需要更频繁地进行备份。

3. tempdb 数据库

tempdb 数据库类似于操作系统的分页文件，用于存储用户创建的临时对象、数据库引擎需要的临时对象和行版本信息。tempdb 数据库是在每次重启 SQL Server 时创建的。当 SQL Server 启动时，该数据库将重新创建为其原始大小。由于该数据库每次都重新创建，因此不能备份它。对 tempdb 数据库中的对象作数据更改可以减少登录。为 tempdb 数据库分配足够的空间非常重要，因为数据库应用中的很多操作都需要使用 tempdb 数据库。通常，应将 tempdb 数据库设置为在需要空间时自动扩展。一般来说，tempdb 数据库的大小不尽相同，但是应该知道数据库应用在峰值时使用多少临时空间，并保证在考虑到 15%~20%的扩展开销的情况下留出足够的空间。如果没有足够空间，用户可能接收如下错误信息之一：

- 1101 或 1105：连接到 SQL Server 的会话必须在 tempdb 数据库中分配空间。
- 3959：版本存储空间已满。
- 3967：版本存储空间必须压缩，因为 tempdb 数据库已满。

4. model 数据库

model 数据库是在 SQL Server 创建新数据库时充当模板的系统数据库。创建每个数据库时，SQL Server 将 model 数据库复制为新数据库。唯一的例外发生在还原或重新连接其他服务器上的数据库时。

如果表、存储过程或数据库选项应包括在服务器上创建的每个新的数据库中，那么通过在 model 数据库中创建该对象可以简化该过程。在创建新数据库时，model 被复制为新数据库，包括在 model 数据库中添加的特殊对象或数据库设置。如果在 model 数据库中添加你自己的对象，那么应该把 model 数据库包括在你的备份中，或是应维护包含更改的脚本。

5. msdb 数据库

msdb 是系统数据库，包含 SQL Server 代理、日志传送、SSIS 以及关系数据库引擎的备份和还原系统等使用的信息。该数据库存储了有关作业、操作员、警报以及作业历史的全部信息。因为包含这些重要的系统级数据，所以应定期对该数据库进行备份。

1.3.4　架构

架构可以对数据库对象进行分组。分组的目的可能是为了易于管理，这样可对架构中的所有对象应用安全策略。使用架构组织对象的另一个原因是使用者可以很容易地发现所需的对象。例如，可创建名为 HumanResources 的架构，并将雇员表和存储过程放入该架构。然后可对该架构应用安全策略，允许对其中包含的对象作适当的访问。

在引用对象时，应使用两部分名称。dbo 架构是数据库的默认架构。dbo 架构中的 Employee 表称为 dbo.Employee。表名在架构中必须是唯一的。也可在 HumanResources 架构中创建另一个名为 Employee 的表，它被称为 HumanResources.Employee。该表实际位于 SQL Server 2012 的 AdventureWorks 示例数据库中(所有的 SQL Server 2012 示例都必须从 wrox.com 单独下载和安装)。例如，使用两部分名称的示例查询如下所示：

```
SELECT BusinessEntityID, JobTitle
FROM HumanResources.Employee
```

在 SQL Server 2005 之前，两部分名称的第一部分是对象所有者的用户名。那种实现方式存在与维护有关的问题。如果拥有对象的用户要离开公司，就不能从 SQL Server 中删除该用户登录，除非确保已将该用户拥有的所有对象改为另一个所有者所有。引用该对象的所有代码必须改为引用这个新所有者。通过将所有关系与架构名分离，从 SQL Server 2005 到 2012 的各个版本消除了这一维护问题。

1.3.5　同义词

同义词是对象的别名或替换名，它在数据库对象和使用者之间创建一个抽象层。这个抽象层使得你可以改变一些物理实现，并将这些更改与使用者隔离开。下面的示例与链接服务器有关。可能表在另一个服务器上，该表需要连接到本地服务器上的表。使用 4 部分名称引用另一服务器上的对象，代码如下所示：

```
SELECT Column1, Column2
```

```
FROM LinkedServerName.DatabaseName.SchemaName.TableName
```

例如，可能为 LinkedServerName.DatabaseName.SchemaName.TableName 创建名为 SchemaName.SynonymName 的同义词。数据使用者将使用下列查询引用该对象：

```
SELECT Column1, Column2
FROM SchemaName.SynonymName
```

现在，这个抽象层允许将表的位置改为另一个服务器，使用不同的链接服务器名，甚至将数据复制到本地服务器来获得更好的性能，而不需要对引用该表的代码作任何更改。

同义词不能引用另一个同义词。object_id 函数返回同义词的 id 而非相关基对象的 id。如果需要列级别的抽象，可以使用视图。

1.3.6 动态管理对象

动态管理对象(Dynamic Management Object，DMO)和函数返回有关 SQL Server 实例和操作系统的信息。DMO 分为两类：动态管理视图(Dynamic Management View，DMV)和动态管理函数(Dynamic Management Function，DMF)。DMV 和 DMF 简化了对数据的访问，并提供了无法通过 SQL Server 2005 之前版本获取的新信息。DMO 可以提供各种信息，包括有关 I/O 子系统和 RAM 的数据以及有关 Service Broker 的信息。

无论何时启动实例，SQL Server 都会开始将服务器状态和诊断信息保存到 DMV 和 DMF 可以访问的内存中。当停止并启动实例时，从视图中清空这些信息，并开始收集新数据。可以像 SQL Server 中的任何其他表一样，用两部分的限定名来查询视图。例如，下列查询使用 sys.dm_exec_sessions DMV 来检索连接到实例的会话数量，并按登录名分组：

```
SELECT login_name, COUNT(session_id) as NumberSessions
FROM sys.dm_exec_sessions GROUP BY login_name
```

有些 DMF 是接受参数的函数。例如，下列代码使用 sys.dm_io_virtual_file_stats 动态管理函数来检索 AdventrueWorks 数据文件的 I/O 统计信息：

```
USE AdventureWorks
GO
SELECT * FROM
sys.dm_io_virtual_file_stats(DB_ID('AdventureWorks'),
FILE_ID('AdventureWorks_Data'))
```

SQL Server 2012 中提供了许多新的 DMV 和 DMF，利用这些视图可以更好地了解新增功能和现有的功能。下面列出了这些新的 DMV 和 DMF：

- AlwaysOn 可用性组动态管理视图和函数
- 与更改数据捕捉有关的动态管理视图
- 与更改跟踪有关的动态管理视图

- 与公共语言运行时(CLR)有关的动态管理视图
- 与数据库镜像有关的动态管理视图
- 与数据库有关的动态管理视图
- 与执行有关的动态管理视图和函数
- SQL Server 扩展事件动态管理视图
- FileStream 和 FileTable 动态管理视图
- 全文搜索和语义搜索动态管理视图和函数
- 与索引有关的动态管理视图和函数
- 与 I/O 有关的动态管理视图和函数
- 与对象有关的动态管理视图和函数
- 与查询通知有关的动态管理视图和函数
- 与复制有关的动态管理视图
- 与资源调控器有关的动态管理视图
- 与安全有关的动态管理视图和函数
- 与服务器有关的动态管理视图和函数
- 与 Service Broker 有关的动态管理视图
- 与空间数据有关的动态管理视图和函数
- 与 SQL Server OS 有关的动态管理视图
- 与事务有关的动态管理视图和函数

1.3.7　SQL Server 2012 数据类型

在 SQL Server 中，数据类型是创建表的基础。在创建表时，必须为表中的每列指派一种数据类型。本节将介绍 SQL Server 中最常用的一些数据类型。即使创建自定义数据类型，也必须基于一种标准的 SQL Server 数据类型。例如，可以使用如下语法创建一种自定义数据类型(Address)，但要注意，它基于 SQL Server 标准的 varchar 数据类型：

```
CREATE TYPE Address
FROM varchar(35) NOT NULL
```

如果在 SQL Server Management Studio 的表设计界面中更改一个大型表中某列的数据类型，那么该操作可能需要很长时间。可以通过在 Management Studio 界面中脚本化这种改变来观察其原因。Management Studio 创建一个辅助的临时表，采用像 tmpTableName 这样的名称，然后将数据复制到该表中。最后，界面删除旧表并用新的数据类型重命名新表。当然，此过程中还涉及其他一些用于处理表中索引和其他任何关系的步骤。

如果有一个包含数百万条记录的大型表，那么该过程可能需要花费 10 分钟，有时可能是数小时。为避免这种情况，可在查询窗口中使用简单的单行 T-SQL 语句来更改该列的数据类型。例如，要将 Employees 表中 JobTitle 列的数据类型改为 varchar(70)，可以使用如下语法：

```
ALTER TABLE HumanResources.Employee ALTER COLUMN JobTitle Varchar(70)
```

在转换为与当前数据不兼容的数据类型时，可能丢失重要数据。例如，如果要将包含一些数据(如 15.415)的 numeric 数据类型转换为 integer 数据类型，那么 15.415 这个数据将四舍五入为整数。

你可能想为 SQL Server 表编写报表，显示表中每列的数据类型。完成这项任务的方法有很多种，但下例演示的这种常用的方法是连接 sys.objects 表和 sys.columns 表。在下面的代码中，有两个函数可能不太为你所熟悉。函数 TYPE_NAME( )将数据类型 id 转换为适当的名称。要进行反向操作，可使用 TYPE_ID( )函数。需要注意的另一个函数是 SCHEMA_ID()，它用于返回架构的标识值。在需要编写有关 SQL Server 元数据的报表时，这是特别有用的。

```
USE AdventureWorks
GO
SELECT o.name AS ObjectName,
        c.name AS ColumnName,
        TYPE_NAME(c.user_type_id) as DataType
FROM sys.objects o
JOIN sys.columns c
ON o.object_id = c.object_id
WHERE o.name ='Department'
and o.Schema_ID = SCHEMA_ID('HumanResources')
```

该代码返回如下结果(注意，Name 是一种用户定义的数据类型)：

```
ObjectName        ColumnName         DataType
--------------------------------------------------
Department        DepartmentID       smallint
Department        Name               Name
Department        GroupName          Name
Department        ModifiedDate       datetime
```

1. 字符数据类型

字符数据类型包括 varchar、char、text 等。这些数据类型用于存储字符数据。varchar 和 char 类型的主要区别是数据填充。如果有个列名为 FirstName 且数据类型为 varchar(20)的表，同时将值 Brian 存储到 FirstName 列中，那么在物理上只存储 5 个字节。但如果在数据类型为 char(20)的列中存储相同的值，将使用全部 20 个字节。SQL 将插入拖尾空格来填满 20 个字符。

你可能想知道，如果要节省空间，那么为什么还使用 char 数据类型呢？这是因为使用 varchar 数据类型会稍增加一些系统开销。例如，如果要存储两字母形式的州名缩写，那么最好使用 char(2)列。尽管有些 DBA 认为应最大可能地节省空间，但一般来说，好的做法是在组织中找到合适的阈值，并指定低于该阈值的采用 char 数据类型，反之则采用 varchar 数据类型。一条可以采用的原则是，任何小于或等于 5 个字节的列都应存储为 char 数据类型，而不是 varchar 数据类型。如果超过这个长度，使用 varchar 数据类型的好处将超过其额外开销。

nvarchar 数据类型和 nchar 数据类型的工作方式与对等的 varchar 数据类型和 char 数据类型相同，但这两种数据类型可以处理国际性的 Unicode 字符。它们需要一些额外开销。以 Unicode 形式存储的数据为一个字符占两个字节。如果要将值 Brian 存储到 nvarchar 列，将使用 10 个字节；而如果将之存储为 nchar(20)，就需要使用 40 个字节。由于这些额外开销和增加的空间，应该避免使用 Unicode 列，除非确实有需要使用它们的业务或语言需求。提前考虑好将来是否需要在实例中使用 Unicode 列。如果将来没有这种需求，就应避免使用它们。

表 1-1 列出了这些类型，对其作了简单描述，并说明了要求的存储空间。

表 1-1　SQL Server 数据类型

数 据 类 型	描　　述	存 储 空 间
char(n)	n 为 1~8000 字符之间	n 字节
nchar(n)	n 为 1~4000 Unicode 字符之间	2×n 字节
nvarchar(max)	最多为 2^{30} - 1(1 073 741 823)Unicode 字符	2×字符数＋2 字节额外开销
text	最多为 2^{31} - 1(2 147 483 647)字符	每字符 1 字节＋2 字节额外开销
varchar(n)	n 为 1~8000 字符之间	每字符 1 字节＋2 字节额外开销
varchar(max)	最多为 2^{31} - 1(2 147 483 647)字符	每字符 1 字节＋2 字节额外开销

2. 精确数值数据类型

数值数据类型包括 bit、tinyint、smallint、int、bigint、numeric、decimal、money、float 以及 real。这些数据类型都用于存储不同类型的数值。第一种数据类型 bit 只存储 null、0 或 1，在大多数应用程序中被转换为 true 或 false。bit 数据类型非常适合用于开关标记，且只占据 1 字节空间。其他常见的数值数据类型如表 1-2 所示。

表 1-2　精确数值数据类型

数 据 类 型	描　　述	存 储 空 间
bit	0、1 或 null	1 字节(8 位)
tinyint	0～255 之间的整数	1 字节
smallint	- 32 768～32 767 之间的整数	2 字节
int	- 2 147 483 648～2 147 483 647 之间的整数	4 字节
bigint	- 9 223 372 036 854 775 808～9 223 372 036 854 775 807 之间的整数	8 字节
numeric(p,s)或 decimal(p,s)	- 1 038＋1～1 038 - 1 之间的数值	最多 17 字节
money	- 922 337 203 685 477.5808～922 337 203 685 477.5807	8 字节
smallmoney	- 214 748.3648～2 14 748.3647	4 字节

decimal 和 numeric 这样的数值数据类型可存储小数点右边或左边的变长位数。scale(表中的 s)是小数点右边的位数。precision(表中的 p)定义了总位数，包括小数点右边的位数。例如，14.88531 可为 numeric(7,5)或 decimal(7,5)。如果将 14.25 插入到 numeric(5,1)列中，该值将被舍入为 14.3。

3. 近似数值数据类型

这个分类中包括数据类型 float 和 real。它们用于表示浮点数据。但是，由于它们是近似的，因此不能精确地表示所有值。

float(n)中的 n 是用于存储该数尾数(mantissa)的位数。SQL Server 对此只使用两个值。如果指定位于 1~24 之间，就使用 24。如果指定 25~53 之间，就使用 53。当指定 float( )时(括号中为空)，默认为 53。

表 1-3 列出了近似数值数据类型，对其进行简单描述，并说明了要求的存储空间。

表 1-3　近似数值数据类型

数 据 类 型	描　　述	存 储 空 间
float[(n)]	－1.79E+308～－2.23E－308,0,2.23E－308～1.79E+308	n≤24－4 字节 n>24－8 字节
real( )	－3.40E+38～－1.18E－38,0,1.18E－38～3.40E+38	4 字节

real 的同义词为 float(24)。

4. 二进制数据类型

varbinary、binary、varbinary(max)等二进制数据类型用于存储二进制数据，如图形文件、Word 文档或 MP3 文件，值为十六进制的 0x0~0xf。image 数据类型可在数据页外部存储最多 2GB 的文件。image 数据类型的首选替代数据类型是 varbinary(max)，可保存超过 8KB 的二进制数据，其性能通常比 image 数据类型好。SQL Server 2012 的新功能是可以在操作系统文件中通过 FileStream 存储选项存储 varbinary(max)对象。这个选项将数据存储为文件，同时不受 varbinary(max)的 2GB 大小的限制。

表 1-4 列出了二进制数据类型，对其做了简单描述，并说明了要求的存储空间。

表 1-4　二进制数据类型

数 据 类 型	描　　述	存 储 空 间
binary(n)	n 为 1~8000 十六进制数字之间	n 字节
varbinary(n)	n 为 1~8000 十六进制数字之间	每字符 1 字节＋2 字节额外开销
varbinary(max)	最多为 2^{31}－1(2 147 483 647)十六进制数字	每字符 1 字节＋2 字节额外开销

5. 日期和时间数据类型

datetime 和 smalldatetime 数据类型用于存储日期和时间数据。smalldatetime 为 4 字节，存储 1900 年 1 月 1 日~2079 年 6 月 6 日之间的时间，并且只精确到最近的分钟。datetime 数据类型为 8 字节，存储 1753 年 1 月 1 日~9999 年 12 月 31 日之间的时间，并且精确到最近的 3.33 毫秒。

SQL Server 2012 有 4 种与日期相关的新数据类型：datetime2、dateoffset、date 和 time。通过 SQL Server 联机丛书可找到使用这些数据类型的示例。

datetime2 数据类型是 datetime 数据类型的扩展，有着更广的日期范围。时间总是用时、分钟、秒的形式来存储。可以定义末尾带有可变参数的 datetime2 数据类型——如 datetime2(3)。这个表达式中的 3 表示存储时秒的小数精度为 3 位，或 0.999。有效值为 0~9 之间，默认值为 3。

datetimeoffset 数据类型和 datetime2 数据类型一样，带有时区偏移量。时区偏移量最大为＋/－14 小时，包含了 UTC 偏移量，因此可以合理化不同时区捕捉的时间。

date 数据类型只存储日期，这是我们一直需要的一个功能。time 数据类型只存储时间，也支持 time(n)声明，因此可以控制小数秒的粒度。与 datetime2 和 datetimeoffset 一样，n 可为 0~7 之间。

表 1-5 列出了日期/时间数据类型，对其进行简单描述，并说明了要求的存储空间。

表 1-5　日期和时间数据类型

数 据 类 型	描　　述	存 储 空 间
date	1 年 1 月 1 日~9999 年 12 月 31 日	3 字节
datetime	1753 年 1 月 1 日~9999 年 12 月 31 日，精确到最近的 3.33 毫秒	8 字节
datetime2(n)	1 年 1 月 1 日~9999 年 12 月 31 日 0~7 之间的 n 指定小数秒	6~8 字节
datetimeoffset(n)	1 年 1 月 1 日~9999 年 12 月 31 日 0~7 之间的 n 指定小数秒＋/－偏移量	8~10 字节
smalldateTime	1900 年 1 月 1 日~2079 年 6 月 6 日，精确到 1 分钟	4 字节
time(n)	小时:分钟:秒.9999999 0~7 之间的 n 指定小数秒	3~5 字节

6. 其他系统数据类型

还有一些之前未见过的数据类型。表 1-6 列出了这些数据类型。

表 1-6　其他系统数据类型

数 据 类 型	描　　述	存 储 空 间
cursor	包含对游标的引用，只能用作变量或存储过程参数	不适用
hierarchyid	包含对层次结构中位置的引用	1~892 字节＋2 字节的额外开销

(续表)

数 据 类 型	描　　述	存 储 空 间
SQL_Variant	可能包含任何系统数据类型的值，除了 text、ntext、image、timestamp、xml、varchar(max)、nvarchar(max)、varbinary(max)以及用户定义的数据类型。最大尺寸为 8000 字节数据＋16 字节元数据	8016 字节
table	用于存储用于进一步处理的数据集。定义类似于 Create Table。主要用于返回表值函数的结果集，它们也可用于存储过程和批处理	取决于表定义和存储的行数
timestamp 或 rowversion	对于每个表来说是唯一的、自动存储的值。通常用于版本戳，该值在插入和每次更新时自动改变	8 字节
uniqueidentifier	可以包含全局唯一标识符(Globally Unique Identifier，GUID)。GUID 值可以从 Newsequentialid()函数获得。这个函数返回的值对所有计算机来说是唯一的。尽管存储为 16 位的二进制值，但仍显示为 char(36)	16 字节
XML	定义为 Unicode 形式	最多 2GB

cursor 数据类型不能用于 Create Table 语句中。

XML 数据类型存储 XML 文档或片段，在存储时使用的空间根据文档中使用 UTF-16 还是 UTF-8 决定，就像 nvarchar(max)一样。XML 数据类型使用特殊构造体进行搜索和索引。第 15 章将更详细地介绍这些内容。

7. CLR 集成

在 SQL Server 2012 中，还可使用公共语言运行时(Common Language Runtime，CLR)中称为 SQLCLR 的部分创建自己的数据类型、函数和存储过程。这让用户可以使用 Visual Basic 或 C#编写更复杂的数据类型以满足业务需求。这些类型被定义为基本的 CLR 语言中的类结构。

1.4　SQL Server 版本

SQL Server 2012 有很多版本，不同版本可用的功能差异也很大。可在工作站或服务器上安装的 SQL Server 版本也会因操作系统而不同。SQL Server 版本包括最低端的 SQL Express(速成版)和最高端的 Enterprise Edition(企业版)。

1.4.1　版本概览

SQL Server 2012 有 3 个主版本。另外还有一些额外的小版本，但是一般不建议在生产环境中使用它们，所以这里不做讨论。更多信息可参考网址 www.microsoft.com/sqlserver。图 1-1 和图 1-2 展示了 SQL Server 2012 的各种版本。

图 1-1

SQL Server 2012 提供了 3 个主要的 SKU：

- 企业版包含 SQL Server 2012 中的全部新功能，包括高可用性和性能更新，使 SQL Server 成为能够处理关键任务的数据库。这个版本中也包含全部 BI 功能。
- 商业智能(BI)版是 SQL Server 2012 中新引入的版本。BI 版提供了 SQL Server 2012 中全套强大的 BI 功能，包括 PowerPivot 和 Power View。这个版本的主要目标之一是使最终用户可以使用强大的 BI 功能。对于需要高级 BI 功能，但是不需要企业版的完整 OLTP 性能和可扩展性的项目，这是一个理想的版本。这个新的 BI 版包含了标准版，并且也提供了基本的 OLTP 功能。
- 标准版与现在一样，是为规模有限的部门使用设计的，提供了基本的数据库功能和基本的 BI 功能。SQL Server 2012 标准版中还新增了一些功能，如压缩。

图 1-2 总结了 SQL Server 2012 的各个版本的关键特征。

SQL Server Capabilities	SQL Server Editions		
	Standard	Business Intelligence	Enterprise
Maximum Number of Cores	16 cores	16 cores – DB OS Max – BI	OS Max
Basic OLTP	●	●	●
Basic Reporting & Analytics	●	●	●
Programmability & Developer Tools	●	●	●
Manageability (Management studio, policy based management)	●	●	●
Enterprise Data Management (Data quality, master data services)		●	●
Self-service Business Intelligence (Power View, PowerPivot for SPS)		●	●
Corporate Business Intelligence (Semantic model, advanced analytics)		●	●
Advanced Security (Advanced auditing, transparent data encryption)			●
Data Warehousing (Column store index, compression, partitioning)			●
Maximum Scalability and Performance			●
High Availability	Limited	Basic	●
AlwaysOn			●
Virtualization Licensing	1 VM	1 VM	Unlimited w/ SA

图 1-2

标准版中提供了基本功能。BI 版中提供了关键的企业 BI 功能。当想要使用高端数据仓库

功能及获得企业级的高可用性时，企业版是正确的选择。

图 1-2 详细列出了 SQL Server 2012 的各个版本以及相关功能。可以看到，标准版最多只能支持 16 核。对于 BI 版，数据库最多可以使用 20 核，BI 版最多可以使用操作系统设置的最大核数。企业版可以使用操作系统支持的最大核数。

BI 版和企业版都提供了 SQL Server 2012 的完整的高级 BI 功能，包括企业数据管理、自助式 BI 和公司 BI 功能。企业版还添加了关键任务和一级数据库功能，并使其实现最好的可扩展性、性能和高可用性。当通过 Microsoft Software Assurance 购买企业版时，客户可以实现无限的虚拟化，获得无限虚拟机的许可。

1.4.2 许可

SQL Server 2012 的许可模式发生了很大的变化。你如果不是使用通过 Microsoft Software Assurance 获取到的许可，这种变化可能会对你的环境产生影响。本节只是概述了这些变化，而没有深入探讨。更多细节可以咨询 Microsoft 客户经理。

SQL Server 2012 的定价和许可方案更符合客户购买数据库和 BI 产品的方式，为客户提供了多种好处，包括：

- SQL Server 2012 提供了市场领先的总体拥有成本(Total Cost of Ownership，TCO)：
  - 在主要供应商中，SQL Server 2012 仍然是明显值得信赖的领袖。这一点通过其定价模型、非企业版中包含的功能以及对各个版本均提供世界一流的支持得以充分体现。
  - 拥有 Software Assurance 的客户可以获得明显的好处，并且在过渡到新许可模型时会得到各种帮助。这些好处包括在许可有效期内能够进行升级和获得更有效的支持。
  - 拥有 Enterprise Agreement 的客户很容易过渡到 SQL Server 2012，并节省大量成本。
- SQL Server 2012 针对云做了优化：
  - SQL Server 2012 是对虚拟化支持程度最好的数据库，提供了扩展的虚拟化许可，允许灵活地为每个 VM 购买许可，并提供了对 Hyper-V 的出色支持。
  - 由于 SQL Server 2012 与 SQL Azure 更好地集成在一起，因此客户还能够支持涉及组织内部、私有云和公有云的混合场景。
- SQL Server 2012 的定价和许可模型允许客户随着自身规模的增长相应购买许可：
  - 新的精简版更符合数据库和 BI 需求。
  - 对于数据中心，许可模型更符合硬件能力。
  - 对于 BI，许可模型符合基于用户的访问，大多数客户都习惯以这种方式购买 BI。

CPU 核(不是处理器)许可

在 SQL Server 2012 版本中，Microsoft 转为采用逐核处理模型。不过不必担心，因为对于大多数服务器，费用应该没有变化。基于 CPU 核的许可(只适用于标准版和企业版)按“双核包”的方式销售。也就是说，对于 4 核 CPU，每个插槽需要两个这样的包。这些许可包的费用只有 SQL Server 2008 R2 CPU 许可的一半。但是，对于每个 CPU，必须至少购买 4 个核的许可。

例如：

- 对于两个插槽，每个插槽 2 个核，需要 4 个许可包(8 个核的许可)。
- 对于两个插槽，每个插槽 4 个核，需要 4 个许可包(8 个核的许可)。
- 对于两个插槽，每个插槽 6 个核，需要 6 个许可包(12 个核的许可)。
- 对于两个插槽，每个插槽 8 个核，需要 8 个许可包(16 个核的许可)。

虚拟化的 SQL Server 和基于主机的许可

当运行虚拟化的 SQL Server 时，必须为 VM 购买至少 4 个核的许可。如果 VM 中有 4 个以上的虚拟 CPU，就必须为分配给 VM 的每个虚拟 CPU 购买 1 个 CPU 核许可。

SQL Server 2012 仍然为拥有 Software Assurance 和 Enterprise Agreement 的那些客户提供了基于主机的许可。基于主机的许可的工作方式与以前一样：为主机购买足够的企业版 CPU 核许可，然后就可以在任意数量的虚拟机中运行 SQL Server。许多用户可能首选这种方式。没有 Software Assurance 或 Enterprise Agreement 的客户需要联系 Microsoft 代理或零售商，因为他们不能购买基于主机的许可。

可以看到，许可模型的变化很大。定价也有所改变，但是其类型取决于个人购买的产品，所以这里不做讨论。Microsoft 已经尽了最大努力让这个版本的价格对于大多数客户来说没有提高太多，所以对这一点不必担心。当然，购买前仍然应该仔细进行考虑，并与自己的 Microsoft 项目团队进行商讨。

1.5　小结

SQL Server 2012 的体系结构有了增强，从而改进了性能、提高了开发人员的效率和系统可用性，并降低了整体运营成本。将注意力集中到自己当前和将来的角色，并理解适用于自己的情况和可以满足组织需求的功能和版本，这是非常重要的。

第 2 章

SQL Server 2012 安装最佳实践

本章主要内容：

- 如何规划并成功地完成 SQL Server 2012 的安装
- 在完成安装后必须进行哪些配置
- 解决常见的安装问题

SQL Server 2012 的安装过程十分简单，只需要在安装媒体中执行安装向导，然后按照每一步的提示进行操作即可。在这个过程中，安装向导会替你做出几个重要的决策。本章将讨论这些决策，以及实现安全、稳定和可扩展的安装的合适配置。

安装过程主要有如下步骤：

(1) 规划系统。
(2) 准备硬件和软件。
(3) 安装操作系统和服务补丁。
(4) 建立 I/O 子系统。
(5) 安装 SQL Server 和服务补丁。
(6) 系统压力测试。
(7) 必要时执行安装后配置。
(8) 清理工作。

本章着重介绍规划系统和实际安装。

2.1 规划系统

在安装 SQL Server 前，第一步是进行合理的规划，这是成功安装 SQL Server 的基础。正如古语所说：不做规划就相当于准备失败。

在规划时，必须考虑下面列出的任务和要点：

- 当前工作负载的基准
- 估计工作负载的增长情况

- 最低硬件和软件需求
- 合适的存储系统大小和 I/O 需求
- SQL Server 版本
- SQL Server 排序规则、文件位置和 tempdb 大小
- 服务账户选择
- 数据库维护和备份计划
- 最小联机时间和响应时间服务等级
- 灾难恢复策略

在部署、升级或迁移 SQL Server 2012 实例时，必须考虑很多因素，上面只是列出了其中的一部分而已。下面将详细介绍其中的一些因素以及相关的最佳实践。

2.1.1 硬件选择

选择正确的硬件配置并不总是很简单。Microsoft 提供了支持 SQL Server 2012 安装的最低硬件需求，但它们只是表示“最低”需求，符合这些需求并不一定意味着配置是最合适的。理想情况下，需要提供超过最低需求的硬件以满足当前和将来的资源需求。

正因如此，需要为当前资源需求创建基准，并估计将来的需求。使硬件可以满足将来的需求不仅可以节省资金，还能够避免因为硬件升级而导致的停机时间。

为了保证平稳地进行安装，并获得可预测的性能，需要熟悉 Microsoft 提供的最低硬件需求，如表 2-1 所示。

表 2-1　SQL Server 2012 的最低硬件需求

硬　　件	需　　求
处理器	64 位安装 速度：1.4GHz 或更高 AMD Opteron、Athlon 64、Intel Pentium IV、支持 Intel EM64T 的 Xeon 32 位安装 速度：1.0GHz 或更高 Pentium III 兼容的处理器
内存	1GB(Express 版为 512MB)
存储器	数据库引擎和数据文件、复制、全文搜索以及数据质量服务：811MB Analysis Services 和数据文件：345MB Reporting Services 和报表管理器：304MB Integration Services：591MB 主数据服务：243MB 客户端组件(除了 SQL Server 联机丛书组件和 Integration Services 工具以外)：1823MB 用于查看和管理帮助内容的 SQL Server 联机丛书组件：375KB

1. 处理器

处理大量事务并且有大量并发连接的 SQL Server 2012 实例可以充分利用可用的处理能力。处理能力表现在处理器的时钟速度高、数量多。几个稍慢的处理器的性能表现要比单个快速的处理器好。例如，两个 1.6GHz 的处理器的速度要比单个 3.2GHz 的处理器更快。

新的处理器模型在一个物理插槽位置可以提供多个核心。这些多核处理器有许多优势，包括可以节省空间和电量消耗。多核处理器允许在同一个物理服务器内以命名实例或虚拟机的形式运行 SQL Server 2012 的多个实例。换句话说，在一个物理服务器上，只要硬件和许可允许，就可以运行任意数量的 SQL Server 2012 服务器。由于多个物理服务器可以合并到单个物理服务器中，数据中心的占用空间得以显著减少。这种合并也使得电费大大降低，因为连接到电网的服务器数减少了。

SQL Server 2012 采用了基于核心的许可模型。在这种模型中，多核处理器的每个核心都必须获得许可。这种改变对于物理服务器和虚拟机都是适用的。关于 SQL Server 2012 许可模型的更多信息，请参阅第 1 章的 1.4.2 节“许可”。

2. 内存

内存是让 SQL Server 2012 实现最佳性能的重要资源。设计良好的数据库系统会尽可能地从内存缓冲区缓存的数据页中读取数据，从而合理利用可用内存。

SQL Server 实例和操作系统都需要使用内存。应该尽量避免在 SQL Server 实例所在的 Windows 服务器上安装内存密集型应用程序。

在决定需要多少内存时，一个不错的起点是考虑 SQL Server 实例中托管的每个数据库的数据页数，以及查询执行统计信息，例如典型的工作负载使用的最小、最大和平均内存。目标应该是让 SQL Server 把尽可能多的数据页保存在缓存中，将尽可能多的执行计划保存在内存中，以避免从磁盘读取数据页以及编译执行计划，这些都是开销很高的操作。

另外，还需要知道具体 SQL Server 版本对内存的限制。例如，SQL Server 2012 企业版支持多达 2TB 的 RAM，标准版支持 64GB 的 RAM，而 Express 版支持 1GB 的 RAM。

3. 存储系统

SQL Server 2012 实例的存储系统需要特别进行考虑，因为缓慢的存储系统可能导致数据库性能严重下降。当为 SQL Server 2012 数据库规划存储系统时，要考虑自己对可用性、可靠性、吞吐量和可扩展性的需求。

为测试和验证存储系统的性能，需要收集一些重要的指标信息，例如每秒最大 I/O 请求数(IOPS)、吞吐量(MBPS)和 I/O 延迟。表 2-2 列出了这 3 个关键指标，并对它们做了简要描述。

表 2-2　关键存储指标

指　　标	描　　述
每秒请求数(IOPS)	存储系统在 1 秒内可以处理的并发请求数。这个数字越高越好。根据具体的配置和制造商，对于单个 15k rpm 的 SAS 驱动器，通常应该为 150~250 IOPS；对于企业 SSD 和 SAN，通常应该为 1000~1 000 000 IOPS
吞吐量(MBPS)	存储系统在 1 秒内可以读写的数据大小。这个数字越高越好
I/O 延迟(ms)	I/O 操作之间的时间延迟。这个数字最好为 0 或接近为 0

通过使用一些免费的工具，如 SQLIO、SQLIOSim、IOMeter 和 CrystalDiskMark，可以收集这些关键的指标。关于如何使用这些工具的介绍不在本章讨论范围内，不过可以在以下网址找到介绍它们的文档：http://msdn.microsoft.com/en-us/library/cc966412.aspx。

SQL Server 安装中主要采用两种类型的存储系统：DAS 和 SAN。

直连式存储(Direct Attached Storage，DAS)

直连式存储理解起来最简单。在这类存储系统中，磁盘驱动器位于服务器机箱内，直接连接到磁盘控制器。它们也可以位于外部，通过缆线直接连接到主机总线适配器(Host Bus Adapter，HBA)上。并不需要使用额外的设备，例如交换机。

DAS 的主要优势在于易于实施且维护成本低，主要缺点在于扩展性有限。虽然在近年来，DAS 存储系统开始具有一些原本只有高端 SAN 存储单元才有的功能，但是一些局限性依然是存在的，例如可以扩展到和管理的磁盘驱动器数和卷大小，可以连接到的服务器数，以及存储单位和服务器之间的距离。

服务器连接和距离是 DAS 和 SAN 的最大区别。DAS 要求存储单位与服务器之间存在直接的物理连接，这就限制了可以同时连接到的服务器数，以及存储单位和服务器之间的距离(通常只有几英尺)。

存储区域网络(Storage Area Network，SAN)

存储区域网络是一种专用的网络，将作为直连式存储卷提供给服务器使用的存储设备相互连接起来。这种存储设备网络的连接方式有两种：通过叫做 fabric 交换机的高速专用光纤通道(Fibre Channel，FC)，或者通过使用常规的以太网交换机的 iSCSI 协议。

SAN 的主要优势之一是通过使用专用的广域网(WAN)和 TCP/IP 路由，可以跨越大片地理区域。这就允许组织在进行灾难恢复时，在相隔遥远的数据中心之间复制数据，以及实现其他一些功能。

另外，SAN 为关键任务数据系统提供了最高的可靠性和可扩展性。与 DAS 相比，合理架构的 SAN 可以提供好得多的吞吐量，并且可以降低 I/O 延迟。而且 SAN 可以扩展，从而处理比 DAS 多得多的磁盘阵列。

SAN 的主要缺点在于成本更高，并且实现和维护的难度更大。

选择合适的存储系统

在安装 SQL Server 时选择的存储系统类型取决于自己的需求。从上面的简单比较中可以知道，DAS 的成本低一些，并且配置和维护都要比 SAN 简单；但是，SAN 在性能、可用性和可扩展性方面更好。

在选择存储系统时，一个关键的考虑因素是存储系统中使用的磁盘技术，以及这些磁盘驱动器是如何排列到一起的。DAS 和 SAN 都使用磁盘驱动器的阵列，并且通常把它们配置为存储池，从而可以把它们作为单个实体提供给服务器使用。

接下来将介绍不同的磁盘驱动器技术，以及如何通过 RAID 级别使它们形成存储池。

磁盘驱动器

如前所述，支持 IO 需求所需的吞吐量是重要的考虑因素之一。为了满足较大的吞吐量需求，经常需要把读写操作分散到大量转速快的磁盘驱动器上。

分散 IO 操作意味着在群组到一起的每个磁盘驱动器上存储少量的数据。在这种分布式存

储中，没有哪个磁盘驱动器包含完整的数据。因此，一次磁盘失败意味着全部数据都会丢失。这就是在做出关于存储系统的决策时，一定要考虑到可靠性的原因。为了避免由于磁盘失败而导致的数据丢失，可以采用一种叫做数据阵列或 RAID 的特殊方法来组织磁盘，以同时满足吞吐量和可靠性需求。选择正确的磁盘 RAID 级别是关键决策，这会影响到服务器的整体性能。表 2-3 描述了 SQL Server 环境中最常用的磁盘 RAID 级别。

表 2-3　常用的 RAID 级别

RAID 级别	描　　述
RAID 0	也叫做条带集或条带卷。将两个或更多个磁盘合在一起，形成单个较大的卷。不能容错。读写快速
RAID 1	也叫做镜像驱动器。将相同的数据写到两个驱动器中。即使其中一个磁盘失败，也不会丢失数据。写操作较慢。只能使用原始存储空间的一半
RAID 1+0	也叫做 RAID 10。条带集中的镜像集。写操作的性能较好，能够容错。只能使用原始存储空间的一半
RAID 0+1	镜像集中的条带集。容错性比 RAID 1+0 稍差。写操作的性能较好
RAID 5	能够容忍其中一个磁盘失败。写操作被分布到各个磁盘中。读操作较快，写操作较慢。部分原始存储空间将无法使用
RAID 6	能够容忍两个磁盘失败。读操作较快，写操作比 RAID 5 更慢，因为奇偶校验计算增加了开销。原始存储空间的丢失情况与 RAID 5 类似

近年来，由于价格的降低和可靠性的提高，一种更快速的磁盘驱动器技术变得越来越流行，即固态磁盘驱动器(Solid State Drive，SSD)。这种磁盘内部没有移动零件。SSD 的读写吞吐量要比转动式磁盘驱动器好 100 倍。SQL Server 可以从更快的读写操作中受益，尤其是那些对 IO 要求高的数据库。

过去几年中，SSD 的采用量得以增长，部分原因在于其可靠性提高，同时价格却比原来低。有几个 SAN 存储系统供应商也提供 SSD 驱动器阵列。

虽然 SSD 磁盘驱动器比转动式磁盘驱动器更加可靠，但是仍然采用 RAID 技术保护 SSD。SSD 磁盘驱动器仍然可能受电子元件失败和损坏的影响。

在选择存储系统时，特别是倾向于使用 DAS 时，另一个要考虑的关键因素是磁盘控制器。在接下来的内容中，你将了解可以改进 DAS 性能和可靠性的磁盘驱动器特征。

磁盘控制器

磁盘控制器是关键硬件，当使用直连式磁盘驱动器时，需要经过仔细考虑再做出选择。磁盘控制器也有吞吐量限制，所以可能导致严重的 IO 瓶颈。

快速的磁盘驱动器还不足以保证快速的存储系统。如果不正确地进行配置，磁盘控制器可能会增加额外的开销。多数磁盘控制器都提供了设置，允许根据具体的工作负载进行配置。例如，可以针对高事务系统配置磁盘控制器，使其针对大量的写操作进行优化。还可以针对大量的读操作优化专门用于报表数据库系统(如数据仓库)和操作型数据存储区的磁盘控制器。

关于磁盘控制器的另外一个重要的考虑是写缓存。尽管这个功能对于改善写操作的性能是很方便的，但一些意外的情况还是可能发生，比如数据丢失或数据库损坏。

磁盘控制器通过把数据临时存储到它们的缓存，并最终以批处理的形式把数据刷新到磁盘来提高写操作的性能。当数据保存到磁盘控制器的缓存中以后，SQL Server 就认为事务已经提交。而实际上，数据还没有提交到磁盘，而只是存在于内存空间中。如果在磁盘控制器把缓存的数据提交到磁盘之前，服务器意外宕机，那么这些数据将无法记录到数据库的事务日志中，从而导致数据丢失。

为了避免潜在的数据丢失，关键数据库环境应该考虑使用带有备用电源(如 UPS)和内部磁盘控制器电池的企业级磁盘控制器。

2.1.2 软件和安装选择

规划的下一步是确保正确设置几个重要的配置选项，如数据库文件的合适位置以及建立合适的服务账户。

1. 排序规则

SQL Server 排序规则指定了一组用于存储、排序和比较字符的规则。排序规则十分重要，因为它指定了使用的代码页。不同的代码页支持不同的字符，并且在排序和比较字符串时表现不同的行为。改变实例的排序规则是很复杂的，所以错误地设置排序规则可能迫使你重新安装 SQL Server 实例。

你需要理解组织和客户对数据的区域、排序、大小写和发音敏感度的需求，以确定使用哪种排序规则。Windows Server 使用的代码页可以在“控制面板”|“区域和语言选项”中找到。为 Windows Server 选择的代码页并不一定就是 SQL Server 实例需要的代码页。

下面介绍了两种常用的排序规则：SQL Server 排序规则和 Windows 排序规则。

SQL Server 排序规则

SQL Server 排序规则影响用于在 char、varchar 以及 text 列中存储数据的代码页，还影响对这些数据类型如何进行比较和排序。例如，如果要对有着区分大小写排序规则的数据库创建下列 SELECT 语句，就不会返回名为 Jose(即混合大小写形式)的雇员。

```
  SELECT FROM Employees
WHERE EmployeeFirstName='JOSE'

Results: <none>

SELECT FROM Employees
WHERE EmployeeFirstName='Jose'

Results: Jose
```

Windows 排序规则

Windows 排序规则使用基于为操作系统选择的 Windows 区域的规则。其默认行为是比较和排序都遵循相关语言的字典采用的规则。可指定区分二进制、大小写、发音、假名以及宽度。关键是 Windows 排序规则确保单字节和双字节字符集都以相同的方式排序和比较。

2. 区分大小写

排序规则可区分大小写，也可不区分。区分大小写意味着 U 和 u 是不同的。排序规则应用的区域中的所有一切(指 master、model、resource、tempdb 以及 msdb 数据库)都是如此。这也适用于这些数据库中的所有数据。这里的要点是，考虑这些数据库中的数据实际是什么。这包括所有系统表中的数据，意味着对象名也要区分大小写。

3. 排序顺序

选择的排序规则也将影响排序。二进制顺序(如 Latin1_General_BIN)根据字符的位值进行排序；它们区分大小写。考虑下面的 SELECT 语句——用于包含雇员姓名 Mary、Tom、mary、tom 等的表，如图 2-1 所示。如果选择一种字典排序顺序(例如 Latin1_General_CS_AI)，那么这条语句将得到如图 2-2 所示的结果。

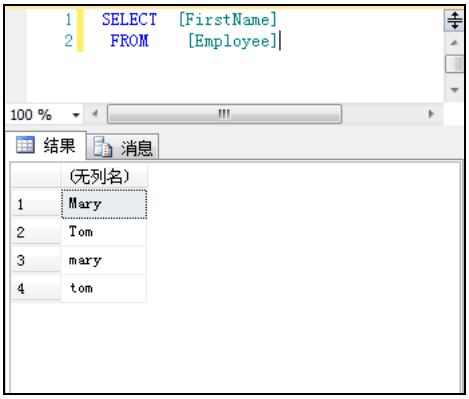

图 2-1

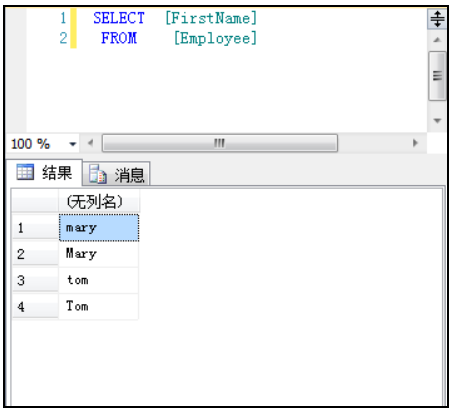

图 2-2

4. 服务账户

服务账户是安全模型的重要组成部分。选择服务账户时，要考虑到最小特权原则。服务账户应该只拥有执行操作需要的最少权限。对于每个服务，应该使用单独的服务账户以单独跟踪各个服务执行的操作。应该为服务账户设置强密码。有下列一些服务账户可供选择：

- **Windows 账户或域账户**：这是创建的活动目录或 Windows 账户，对于需要网络访问的 SQL Server 服务来说是首选的账户类型。
- **本地系统账户**：这是一种有很高权限的账户，不应用于运行服务，因为是作为网络中的一台计算机，没有密码。使用本地系统账户的被破坏进程也会破坏数据库系统。
- **本地服务账户**：这是一种预配置的特殊账户，和“用户”组成员有着相同的权限。网络访问将通过不需要凭据的空会话进行。这是一类不受支持的账户。
- **网络服务账户**：这个账户与本地服务账户相同，但允许网络访问，被看做计算机账户。不要将这类账户用作 SQL Server 或 SQL 代理服务账户。
- **本地服务器账户**：这是创建的本地 Windows 账户。对于不需要网络访问的服务来说，这是最安全的方法。

对于生产系统，应该使用专用的 Windows 账户或域账户。一些组织选择为所有的 SQL Server

实例创建单个域账户，而其他组织选择为每个服务创建单独的账户。只要服务账户使用了强密码并得到保护，使用哪种方法都可以。

2.2 安装 SQL Server

本节介绍不同类型的安装：全新安装、并列安装和升级安装。你将学习如何使用图形用户界面(GUI)、命令提示符、配置文件和 PowerShell 脚本进行自动和手动安装。第 3 章中将介绍有关升级安装的详细内容。

2.2.1 全新安装

如果服务器上没有其他 SQL Server 组件且具备干净的系统环境，就将执行全新安装。检查目录和注册表，以确保是干净的系统环境，没有以前 SQL Server 安装的残留物。

2.2.2 并列安装

SQL Server 还支持并列安装。当服务器上有多个 SQL Server 实例时，就会发生并列安装。SQL Server 2012 支持在一台服务器上有数据库引擎、Reporting Services 和 Analysis Services 的多个实例，并且还可以与以前的 SQL Server 版本并列运行。如果现有实例为默认实例，那么新安装必须为命名实例，因为每台服务器上只能有一个默认实例。

并列安装最大的问题是内存争用。要确保正确配置了内存，这样每个实例就不会试图获取全部物理内存。如果不同实例的数据库文件共享相同的存储资源，那么 IO 争用也可能成为一个问题。

2.2.3 升级安装

如果服务器上已有 SQL Server 组件，就可升级现有实例。在这种情况下，是在现有实例上安装 SQL Server，也称为就地安装。为了从以前的版本升级到 SQL Server 2012，需要在 SQL Server 安装中心中启动升级向导，然后使用“安装”选项卡下的“从 SQL Server 2005、SQL Server 2008 或 SQL Server 2008 R2 升级”。

2.2.4 自动安装

SQL Server 2012 允许通过命令行参数或配置文件执行自动安装。自动安装允许在多个服务器上以完全相同的配置安装 SQL Server，安装过程基本上不需要用户交互。屏幕上的所有选项和对话框的响应都是使用配置文件中存储的信息或命令行参数自动选择的。

使用命令行参数自动安装

使用命令行安装新的 SQL Server 2012 的步骤如下：

(1) 以提升的管理员权限启动“命令提示”窗口。方法是右击“命令提示”窗口的可执行程序，然后从上下文菜单中选择“以管理员身份运行”。“命令提示”窗口将会打开。

(2) 在命令行中，输入以下命令，然后按 Enter 键：

```
D:\setup.exe /ACTION=install /QS /INSTANCENAME="MSSQLSERVER" /
IACCEPTSQLSERVERLICENSETERMS=1
/FEATURES=SQLENGINE,SSMS
/SQLSYSADMINACCOUNTS="YourDomain\Administrators"
```

安装路径因安装媒体而异。而且根据要安装的功能，参数也可能发生变化。另外，还要将/SQLSYSADMINACCOUNTS 的值设为有效的域名和用户账户。

这个命令行脚本自动安装 SQL Server 数据库引擎和 SQL Server 基本管理工具。表 2-4 描述了前面脚本中使用的每个命令行参数。

表 2-4　命令行参数

参　　数	描　　述
/ACTION	指定要执行的动作。在本例中为全新安装
/QS	指定安装程序运行并显示安装进度，但是不接受输入，也不显示错误消息
/INSTANCENAME	指定必需的实例名
/IACCEPTSQLSERVERLICENSETERMS	使用/Q 或/QS 时需要使用此参数确认接受许可条款
/FEATURES	用于指定安装的功能的必需参数
/SQLSYSADMINACCOUNTS	用于指定 sysadmin 角色的成员的必需参数

关于命令行参数的完整列表，请访问以下网址：http://msdn.microsoft.com/en-us/library/ms144259(v=sql.110).aspx。

使用配置文件自动安装

在默认情况下，SQL Server 2012 安装程序会创建一个配置文件，用于记录在安装过程中指定的选项和参数值。这个配置文件可以为验证和审核提供帮助，对于使用相同配置部署额外的 SQL Server 安装特别有用。

创建配置文件的步骤如下：

(1) 从 SQL Server 2012 安装媒体中启动 Setup.exe。SQL Server 的安装程序将会启动。

(2) 为 SQL Server 2012 安装指定选项和参数。在完成“安装向导”的过程中，指定的所有选项和值将会记录在配置文件中。

(3) 按照“安装向导”的提示进行设置，直到进入“准备安装”屏幕，如图 2-3 所示。

图 2-3

在图 2-3 中，注意 ConfigurationFile.ini 文件的路径。到现在，配置文件已被创建，在前面的向导页面中指定的所有选项和参数值都已被记录。

(4) 打开 Windows 资源管理器，导航到配置文件所在的文件夹。单击“取消”按钮，停止 SQL Server 2012 安装向导。

(5) 找到 ConfigurationFile.ini 文件，将其复制到自动安装过程中可以引用的文件夹中。例如，使用共享文件夹\\fileserver\myshare。

(6) 打开 ConfigurationFile.ini 文件并进行如下修改，以便为自动安装准备好文件：

- 设置 QUIET="True"
- 设置 SQLSYSADMINACCOUNTS="YourDomain\Administrators"
- 设置 IACCEPTSQLSERVERLICENSETERMS="True"
- 删除 ADDCURRENTUSERASSQLADMIN
- 删除 UIMODE

在为自动安装定制好配置文件后，使用命令提示符执行 Setup.exe，指定配置文件的路径。下面的命令行脚本演示了这种语法：

```
D:\Setup.exe /ConfigurationFile=\\fileserver\myshare\ConfigurationFile.ini
```

使用 PowerShell 进行脚本安装

还可以使用 PowerShell 执行自动安装。编写简单的 PowerShell 脚本，通过其命令行界面执行 SQL Server 2012 安装程序。例如，可以像下面这样，在其命令行中执行刚才使用过的命令行脚本：

```
$cmd = "d:\setup.exe /ACTION=install /Q /INSTANCENAME="MSSQLSERVER" /
IACCEPTSQLSERVERLICENSETERMS=1
/FEATURES=SQLENGINE,SSMS
```

```
/SQLSYSADMINACCOUNTS="YourDomain\Administrators";
Invoke-Expression -command $cmd | out-null;
```

针对大型的 SQL Server 2012 部署，可以编写更加复杂的 PowerShell 脚本。常见的一种方法是使用接受执行自动安装所需的安装参数的 PowerShell 函数。然后，这些 PowerShell 函数将以批处理的形式执行，或者在遍历具有对应参数的服务器名列表的进程中执行。

例如，可以在 PowerShell 脚本文件中保存 PowerShell 函数，然后用安装参数来调用这个函数以执行大规模的 SQL Server 2012 自动部署。程序清单 2-1 给出了一个 PowerShell 函数的示例，它可以用于 SQL Server 2012 自动安装。

程序清单 2-1　Install-Sql2012.ps1

```
Function Install-Sql2012
{
  param
  (
    [Parameter(Position=0,Mandatory=$false)][string] $Path,
    [Parameter(Position=1,Mandatory=$false)][string] $InstanceName =
    "MSSQLSERVER",
    [Parameter(Position=2,Mandatory=$false)][string] $ServiceAccount,
    [Parameter(Position=3,Mandatory=$false)][string] $ServicePassword,
    [Parameter(Position=4,Mandatory=$false)][string] $SaPassword,
    [Parameter(Position=5,Mandatory=$false)][string] $LicenseKey,
    [Parameter(Position=6,Mandatory=$false)][string] $SqlCollation =
  "SQL_Latin1_General_CP1_CI_AS",
    [Parameter(Position=7,Mandatory=$false)][switch] $NoTcp,
    [Parameter(Position=8,Mandatory=$false)][switch] $NoNamedPipes
  )
  #Build the setup command using the install mode
  if ($Path -eq $null -or $Path -eq "")
  {
  #No path means that the setup is in the same folder
  $command = 'setup.exe /Action="Install"'
  }
  else
  {
  #Ensure that the path ends with a backslash
  if(!$Path.EndsWith("\"))
  {
    $Path += "\"
  }
  $command = $path + 'setup.exe /Action="Install"'
  }
  #Accept the license agreement - required for command line installs
  $command += ' /IACCEPTSQLSERVERLICENSETERMS'
  #Use the QuietSimple mode (progress bar, but not interactive)
  $command += ' /QS'
  #Set the features to be installed
  $command += ' /FEATURES=SQLENGINE,CONN,BC,SSMS,ADV_SSMS'
```

```
#Set the Instance Name
$command += (' /INSTANCENAME="{0}"' -f $InstanceName)
#Set License Key only if a value was provided,
#else install Evaluation edition
if ($LicenseKey -ne $null -and $LicenseKey -ne "")
{
  $command += (' /PID="{0}"' -f $LicenseKey)
}
#Check to see if a service account was specified
if ($ServiceAccount -ne $null -and $ServiceAccount -ne "")
{
#Set the database engine service account
$command += (' /SQLSVCACCOUNT="{0}" /SQLSVCPASSWORD="{1}"
/SQLSVCSTARTUPTYPE="Automatic"' -f
$ServiceAccount, $ServicePassword)
#Set the SQL Agent service account
$command += (' /AGTSVCACCOUNT="{0}" /AGTSVCPASSWORD="{1}"
  /AGTSVCSTARTUPTYPE="Automatic"' -f
$ServiceAccount, $ServicePassword)
}
else
{
#Set the database engine service account to Local System
$command += ' /SQLSVCACCOUNT="NT AUTHORITY\SYSTEM"
/SQLSVCSTARTUPTYPE="Automatic"'
#Set the SQL Agent service account to Local System
$command += ' /AGTSVCACCOUNT="NT AUTHORITY\SYSTEM"
  /AGTSVCSTARTUPTYPE="Automatic"'
}
#Set the server in SQL authentication mode if SA password was provided
if ($SaPassword -ne $null -and $SaPassword -ne "")
{
$command += (' /SECURITYMODE="SQL" /SAPWD="{0}"' -f $SaPassword)
}
#Add current user as SysAdmin
$command += (' /SQLSYSADMINACCOUNTS="{0}"' -f
  [Security.Principal.WindowsIdentity]::GetCurrent().Name)
#Set the database collation
$command += (' /SQLCOLLATION="{0}"' -f $SqlCollation)
#Enable/Disable the TCP Protocol
if ($NoTcp)
{
  $command += ' /TCPENABLED="0"'
}
else
{
  $command += ' /TCPENABLED="1"'
}
#Enable/Disable the Named Pipes Protocol
if ($NoNamedPipes)
```

```
{
  $command += ' /NPENABLED="0"'
}
else
{
  $command += ' /NPENABLED="1"'
}
if ($PSBoundParameters['Debug'])
{
  Write-Output $command
}
else
{
  Invoke-Expression $command
}
}
```

从 Wrox 的网站下载了程序清单 2-1 的代码后，将其保存到一个文件夹中，例如 c:\scripts。因为这是从 Internet 上下载的文件，所以可能需要右击该文件并选择“属性”，在打开的对话框中选择“解除锁定”命令。下载完该文件后，按照以下步骤执行此函数：

(1) 使用提升的管理员权限启动 PowerShell 命令行。方法是右击 PowerShell 可执行文件，然后选择“以管理员身份运行”。PowerPoint 命令行将会打开。

(2) 确认自己可以运行和加载未签名的 PowerShell 脚本和文件。在 PowerShell 命令行中，输入 get-executionpolicy 来确认当前的执行策略。如果执行策略不是 RemoteSigned，就需要执行下面的命令以将其设为这个值：

```
Set-ExecutionPolicy RemoteSigned
```

(3) 接下来，执行下面的命令来加载脚本文件中的 PowerShell 函数：

```
. c:\scripts\Install-Sql2012.ps1
```

注意脚本文件路径前面的点号和空格。点号和空格是使用点号访问脚本文件所必需的。

(4) 执行下面的命令来确认函数已被加载：

```
get-command Install-Sql2012
```

执行命令后将返回单独的一行结果，其中显示了 CommandType 为 Function，Name 为 Install-Sql2012。

(5) 现在，可以调用刚才加载的 PowerShell 函数了。调用 Install-Sql2012 的命令如下：

```
Install-Sql2012 -Param1 Param1Value -Param2 Param2Value ..
```

(6) 例如，下面的命令调用了 Install-Sql2012 函数，并设置了 SQL Server 服务账户、密码以及实例名，还启动了 SQL Server 2012 安装：

```
Install-Sql2012 -Path d:\ -ServiceAccount "winserver\Administrator" -
ServicePassword "P@ssword"
-SaPassword "P@ssword"
-InstanceName "MyInstanceName"
```

SQL Server 2012 的安装路径可能随安装媒体的不同发生变化。

图 2-4 显示了 PowerShell 命令行窗口，以及调用 Install-Sql2012 函数的必要步骤。

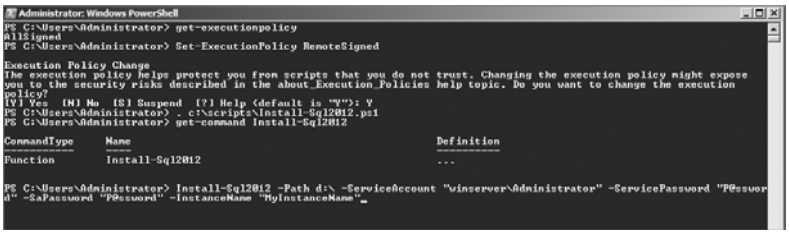

图 2-4

在 Codeplex.com 上可以下载由社区创建的项目 SPADE，该项目使用 PowerShell 自动完成 SQL Server 的安装。

2.2.5 手动安装

最简单、最常见的 SQL Server 部署方式是使用安装向导提供的 GUI 进行手动安装。手动安装需要用户频繁参与，以提供完成 SQL Server 2012 安装所需的信息和参数值。

执行以下步骤来完成 SQL Server 2012 的手动安装：

(1) 从 SQL Server 2012 安装媒体中启动 Setup.exe。“SQL Server 安装中心”将会打开，如图 2-5 所示。

图 2-5

(2) 单击左侧的“安装”选项卡，然后单击右侧的第一个选项，其标题为“全新 SQL Server 独立安装或向现有安装添加功能”。SQL Server 2012 安装向导将会启动。

(3) 安装程序支持规则将会运行，以识别在安装程序支持文件的安装过程中可能发生的问题。完成这个步骤后，单击“确定”按钮。“安装安装程序文件”进程将会启动。

(4) 在安装程序文件安装完成后，还需要检查另外一组安装支持文件。单击“下一步”按钮继续。

(5) 对于某些安装媒体和许可协议，可能需要选择 SQL Server 的版本，并在下一个屏幕中输入产品密钥。单击“确定”按钮继续。

(6) “许可条款”屏幕将会打开。接受许可条款，然后单击“下一步”。

(7) “设置角色”屏幕将会打开。选择“SQL Server 功能安装”选项，然后单击“下一步”。

(8) “功能选择”屏幕将会打开。选中“数据库引擎服务”和“管理工具-基本”选项，然后单击“下一步”。图 2-6 显示了 SQL Server 2012 中可以安装的功能列表。

(9) “安装规则”屏幕将会打开，以判断是否存在可能阻止安装的问题。单击“下一步”。

(10) “实例配置”屏幕将会打开。在这个屏幕中，可以决定将实例作为默认实例或命名实例来安装。还可以提供实例 ID 以及改变默认根目录。单击“下一步”。

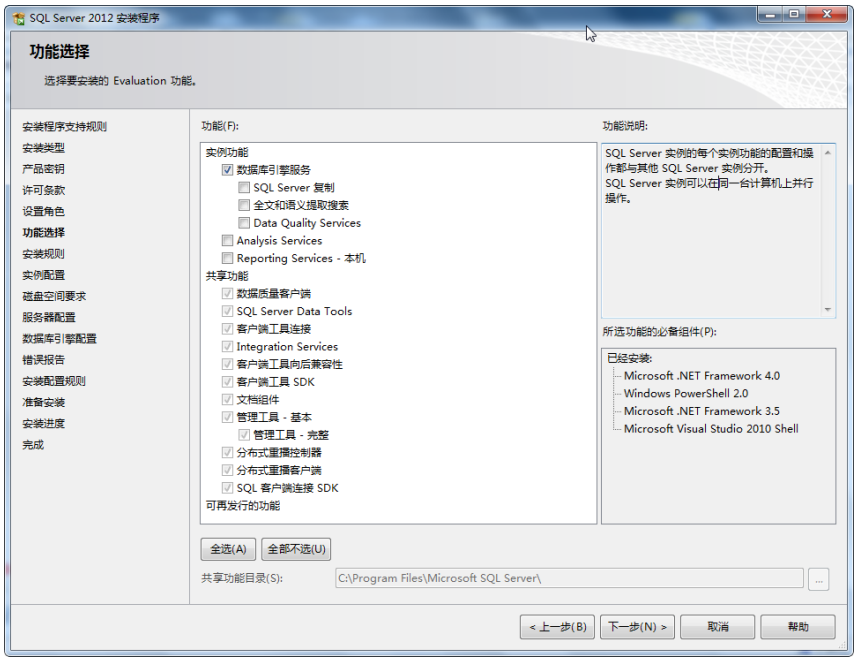

图 2-6

(11) “磁盘空间要求”屏幕将会打开，显示所选功能占用空间的摘要。单击“下一步”。

(12) “服务器配置”屏幕将会打开。提供运行 SQL Server 数据库引擎、SQL Server 代理和 SQL Server Browser 的服务账户，以及 SQL Server 2012 使用的排序规则。单击“下一步”。图 2-7 显示了“服务器配置”屏幕。

图 2-7

作为一种安全最佳实践，在选择服务账户时一定要考虑遵守最小特权原则。只要有可能，就应该避免分配在域或服务器中拥有提升权限的服务账户。

(13) “数据库引擎配置”屏幕将会打开。在这里指定身份验证模式、SQL Server 管理员和默认的数据目录，并启用 FILESTREAM。在这个屏幕中至少指定一个拥有 SQL Server 管理员权限的账户是很重要的，而且需要在这个屏幕中将身份验证模式指定为“Windows 身份验证模式”或“混合模式”。如果选择“Windows 身份验证模式”，那么只有经过身份验证的 Windows 账户可以登录；如果选择“混合模式”，那么 Windows 账户和 SQL Server 账户都可以登录。单击“下一步”。

不要将 SA 账户的密码留空。一定要为之指定强密码，并在安装完成后禁用该账户，以避免针对这个众所周知的账户的安全攻击。

(14) “错误报告”屏幕将会打开。单击复选框，选择将错误报告发送给 Microsoft。单击“下一步”。

(15) “安装配置规则”屏幕将会打开。当“安装配置规则”完成扫描后，单击“下一步”。

(16) “准备安装”屏幕将会打开。此时，安装向导已经收集了所有必要的信息，并且在启动安装过程之前把它们显示出来，以便进行检查。单击“安装”按钮启动安装。

这就完成了 SQL Server 2012 的手动安装过程。

Microsoft 专门为 SQL Server 2012 提供了一些示例数据库，供用户下载使用。从 Codeplex.com 上可以免费下载这些示例数据库和示例项目文件，网址为 http://msftdbprodsamples.codeplex.com/。

2.3　安装 Analysis Services

SQL Server Analysis Services 的安装过程很简单。可以随其他 SQL Server 2012 功能和服务一起安装，也可以单独安装，如图 2-8 所示。

图 2-8

SQL Server 2012 中的新增之处是允许以两种方式安装 Analysis Services：

- 多维和数据挖掘模式(UDM 模式)
- 表格模式

这两个模式是在 SQL Server 2012 安装程序的“Analysis Services 配置”屏幕中选择的，如图 2-9 所示。

图 2-9

下面将简要描述 Analysis Services 的这两种安装模式。

2.3.1 多维和数据挖掘模式(UDM 模式)

Analysis Services 的多维和数据挖掘模式(UDM 模式)安装自 SQL Server 2005 以来就可用的传统的 Analysis Services。这个引擎基于统一维度模型(Unified Dimensional Mode，UDM)。

UDM 作为一个或更多个数据源之间的中间层，可以合并所有的业务规则。作为多维模型的中心，UDM 使用户能够查询、聚合、深化 Analysis Services 数据库以及对 Analysis Services 数据库执行切片和切块操作，并且操作的响应时间十分快速。

Analysis Services 的 UDM 模式支持接下来将要描述的 MOLAP、ROLAP 和 HOLAP 存储和处理模式。

1. MOLAP(多维 OLAP)

在 MOLAP 存储模式中，数据存储和聚合在多维数据库的一个或更多个分区中。这种存储模式可以最大化查询性能。使用 MOLAP 存储模式存储的数据是自上一次处理后的数据。

2. ROLAP(关系 OLAP)

在 ROLAP 存储模式中，数据没有存储到 Analysis Services 数据库中。如果查询缓存不能满足查询，那么查询将使用内部索引视图从关系型数据源检索数据。在这种模式下，查询时间比 MOLAP 更慢，但是数据则更新，与源事务系统是同步的。

3. HOLAP(混合 OLAP)

在 HOLAP 存储模式下，多维数据库的一部分数据存储在 MOLAP 中，一部分则直接从其关系型数据源检索。访问聚合程度高的数据的查询由其多维存储满足，这与 MOLAP 模式一样。深化查询则直接从其关系型数据源检索，这与 ROLAP 一样。

2.3.2 表格模式

Analysis Services 的表格模式基于如下新的数据库引擎：VertiPaq 引擎。VertiPaq 是一种基于列的数据库，由于对存储离散值的需求降低，并且采用了更先进的压缩算法，因此能够提供很高程度的压缩。

压缩大量数据的能力使 VertiPaq 引擎能够以比传统的基于磁盘的 Analysis Services 引擎更快的速度存储、检索和操纵 RAM 中的数据。

表格模式支持一种新的语义模型，叫做商业智能语义模型(Business Intelligence Semantic Model，BISM)。其开发环境与 Excel 的 PowerPivot 加载项类似。Excel 的 PowerPivot 加载项运行的是 Analysis Services 表格模式使用的 VertiPaq 引擎的简化版本。

与传统的 Analysis UDM 引擎类似，Analysis Services 表格模式支持从内存的存储区中查询，直接从关系型数据源查询，或者混合使用两者。这些查询模式选项可以通过 BISM 模型属性选择。4 种可用的查询模式选项包括：

- Vertipaq
- Direct Query

- InMemorywithDirectQuery
- DirectQueryWithInMemory

接下来就介绍这 4 种查询模式选项。

1. Vertipaq 查询模式

在 Vertipaq 查询模式下，数据存储在内存的存储数据集中，并从这些存储数据集中进行查询。磁盘 IO 的延迟开销被最小化甚至消除，所以这种模式几乎没有延迟。因为访问的是位于内存中的数据集，所以复杂的计算和排序操作几乎是立刻完成的。

Vertipaq 查询模式与传统的 Analysis Services 引擎的 MOLAP 存储相似，它们都保存时间点(Point-In-Time，PIT)数据。如果关系型数据源中的数据发生变化，就需要使用新的数据集刷新内存中的存储区。

2. Direct Query 查询模式

Direct Query 模式也叫做穿透模式。在这种模式下，查询由其数据源处理，通常是关系数据库。与 Vertipaq 查询模式相比，Direct Query 查询模式的优势在于能够提供大型数据卷的实时数据集，而这些数据卷是不适合放到内存中的。

3. InMemoryWithDirectQueryMode 查询模式

在这种查询模式下，查询默认使用存储在缓存中的数据集，除非客户端的连接字符串另有指定。这种模式支持客户端切换为使用实时数据。

4. DirectQueryWithInMemoryMode 查询模式

与 InMemoryWithDirectQueryMode 模式相反，除非客户端的连接字符串另有指定，否则查询默认使用关系型数据源。这种模式支持客户端切换为使用缓存数据。

2.4　安装 PowerPivot for SharePoint

PowerPivot for SharePoint 模式是自 SQL Server 2008 R2 以来就可用的一种角色选项。PowerPivot for SharePoint 选项安装新的 Analysis Services Vertipaq 引擎的一个版本，以支持发布到 SharePoint 2010 的 PowerPivot 工作簿的服务器端处理和管理。

要将 PowerPivot for SharePoint 安装到一台服务器上，该服务器必须连接到域，并且该服务器上必须安装了 PowerPivot 2010 Enterprise with Service Pack 1，实例名 PowerPivot 必须可用。

安装 PowerPivot for SharePoint 的步骤如下：

(1) 从 SQL Server 2012 安装媒体中启动 Setup.exe。安装中心将会打开。

(2) 单击左侧的“安装”选项卡，然后单击右侧的第一个选项“全新 SQL Server 独立安装或向现有安装添加功能”。SQL Server 2012 安装向导将会打开。

(3) 安装程序支持规则将会运行，以识别在安装程序支持文件的安装过程中可能发生的问题。完成这个步骤后，单击“确定”按钮。“安装安装程序文件”进程将会启动。

(4) 在安装程序文件安装完成后，还需要检查另外一组安装支持文件。单击“下一步”按钮继续。

(5) 对于某些安装媒体和许可协议，可能需要选择 SQL Server 的版本，并在下一个屏幕中输入产品密钥。单击“确定”按钮继续。

(6) “许可条款”屏幕将会打开。接受许可条款，然后单击“下一步”。

(7) “设置角色”屏幕将会打开。选择“SQL Server PowerPivot for SharePoint”选项。单击该选项下方的复选框，还可以在这次安装中安装 SQL Server 数据库服务的实例，如图 2-10 所示。

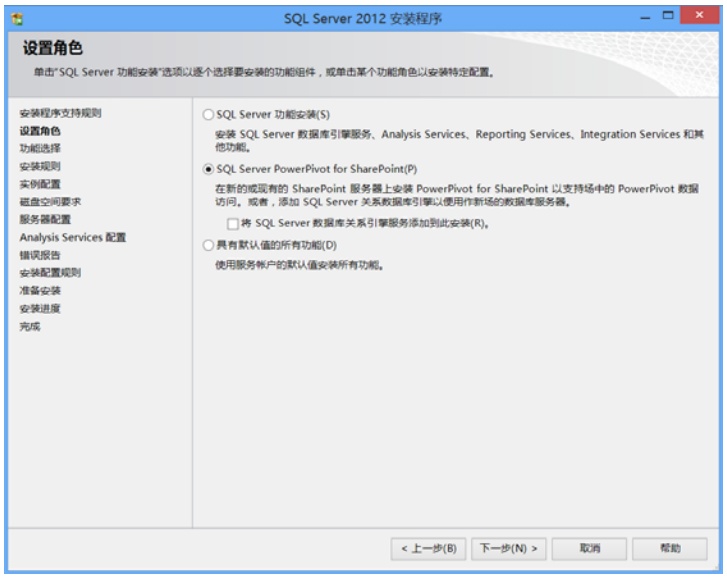

图 2-10

(8) 单击“下一步”。“功能选择”屏幕将会打开。其中的选项已被预先选中并显示出来。

(9) 单击“下一步”。“安装规则”屏幕将会打开，以判断是否存在可能阻止安装的问题。单击“下一步”。

(10) “实例配置”屏幕将会打开。实例名是不能修改的，只能是 PowerPivot。能够修改的只有实例 ID。单击“下一步”。

(11) “磁盘空间要求”屏幕将会打开，显示所选功能占用空间的摘要。单击“下一步”。

(12) “服务器配置”屏幕将会打开。在这个屏幕中提供运行 Analysis Services 引擎的服务账户。必须使用域账户。单击“下一步”。

(13)“Analysis Services 配置”屏幕将会打开。在这个屏幕中添加需要 Analysis Services 实例的管理员权限的域账户。

(14) 单击“下一步”，直到进入“准备安装”屏幕。检查安装摘要页面，然后单击“安装”。

这样就最终完成了 PowerPivot for SharePoint 的安装。

2.5 系统压力测试

在将系统投入正常使用前，应对其进行压力测试。通常，已投入生产运行数月或数年的服务器总是存在问题，而这些问题在部署服务器时就已存在。许多故障在服务器负载较轻时不会

显现，而当服务器负载较高时就会立即突现。

有几个免费的工具可以对数据库服务器进行压力测试，以确保存储系统已经准备好处理必要的 IO 工作负载、内存压力和对 CPU 处理能力的需求。下面列出了几个这样的工具：

- SQLIOSim：Microsoft 设计的免费工具，用于生成类似的 SQL Server IO 读写模式。这个工具对于测试 IO 密集型操作(如 DBCC CHECKDB 和批量插入、删除和更新操作)意义重大。SQLIOSim 替换了 SQLIOStress，从以下网址可以下载 SQLIOSim：http://support.microsoft.com/kb/231619。
- IOMeter：另外一个运行压力测试的免费工具，能够模拟并发应用程序工作负载。
- Prime95：这个免费工具被设计用于找出梅森素数，这是一种 CPU 和 RAM 密集型操作。可以定制该工具，对 CPU 和内存的工作负载进行长时间的压力测试。

在网上还可以搜索找到其他一些免费和付费的应用程序来执行最初的压力测试。一些服务器和服务器组件制造商也提供了工具来对设备进行基准测试和压力测试。

2.6　安装后的配置

在安装了 SQL Server 2012 以后，需要配置额外的一些设置，并完成必要的任务以得到一台能够投入生产的服务器。其中，一些设置用于调优 SQL Server 实例来获得最佳性能，如最大服务器内存、并行度阈值和网络数据包大小。其他一些设置和任务则用于保护、审核和监控 SQL Server 实例，如改变默认端口、登录审核和禁用 SA 账户。

2.6.1　配置 SQL Server 设置以实现高性能

SQL Server 2012 提供了一些可以针对特定的环境和工作负载模式进行优化的系统设置。接下来就讨论一些最重要的性能设置。

1. 内存

最大和最小服务器内存是两个重要的服务器属性设置。SQL Server 的默认配置是最小内存为 0MB，最大内存为 2 147 483 647MB(2TB)，如图 2-11 所示。

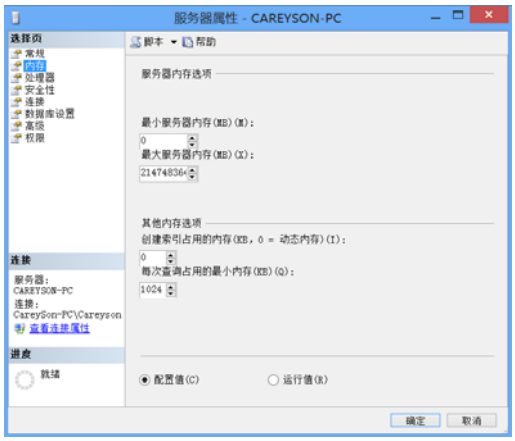

图 2-11

保留这两个设置的默认值的后果有时候会被误解，并且经常会被忽视。最小服务器内存设置指定了在分配后，SQL Server 不会返回给操作系统的内存量。换句话说，即使不再需要这部分最小内存，SQL Server 也仍然会保持占有它们。

一个常见的误解是 SQL Server 在启动后会立即分配达到最小内存量的内存。实际上，只有在收到请求时，SQL Server 才会分配内存，分配的内存可能会、也可能不会达到指定的最小服务器内存值。

最小服务器内存设置一般不用改变，除非操作系统不断为共享相同内存空间的其他应用程序请求内存资源。应该避免向操作系统系统释放太多内存，因为这可能导致 SQL Server 实例缺少足够的内存资源。

另一方面，最大服务器内存为 SQL Server 实例可以分配的内存量设置了最大限制。把这个值设置得太高可能会使操作系统没有足够的内存资源可用。最大服务器内存值不应等于或超过总的可用服务器内存，至少应该比总的服务器内存小 1GB。

2. 网络数据包大小

SQL Server 2012 的默认网络数据包大小为 4096 字节。将数据包的大小设置得比默认值高可以改进需要执行大量批量操作和传输大量数据的 SQL Server 实例的性能。

如果服务器的硬件和网络基础设施支持并启用了 Jumbo Frames，那么最好把网络数据包的大小增加为 8192。

3. 即时文件初始化

每当数据库文件被创建或需要增长时，操作系统就会用 0 填充数据库文件，然后新的空间才可被写入。由于在 0 填充完成之前，所有的写操作都会被阻塞，因此这个操作的开销可能很大。为了避免这类阻塞和等待，可以启用即时文件初始化。具体方法是将 SQL Server 服务账户添加到服务器的“安全性设置”的“本地策略”的“用户权限管理”下的“执行大量维护任务”策略的用户列表中。

2.6.2 tempdb

tempdb 是最重要的系统数据库之一，需要特别考虑和计划。与过去相比，tempdb 承担了更多的责任。tempdb 以前只用于内部进程，如建立索引和存储表变量，以及作为编程人员的临时存储空间。下面列出了 tempdb 的部分用途：

- 用触发器批量加载
- 公共表表达式
- DBCC 操作
- 事件通知
- 索引重建，包括 SORT_IN_TEMPDB、分区索引排序以及联机索引操作
- 大型对象类型变量和参数
- 多活动结果集操作
- 查询通知
- 行版本控制

- Service Broker

对于大量使用 tempdb 的环境，创建额外的 tempdb 文件可以显著提升性能。根据工作负载，可以考虑创建与每个逻辑 CPU 成正比的大量 tempdb 文件，使 SQL Server 计划程序工作线程可以松散对齐到某个文件。一般来说，可接受的 tempdb 文件与逻辑 CPU 的比率在 1:2 到 1:4 之间。在极端情况下，可能想要为每个 CPU 创建 tempdb 文件(1:1)。确定应该创建多少个 tempdb 文件的唯一方式是进行测试。

tempdb 的位置十分重要。tempdb 文件应该与数据库文件和日志文件分隔开，以避免出现 IO 争用。如果使用了多个 tempdb 文件，那么可以考虑将每个 tempdb 文件隔离到自己的 LUN 和物理磁盘上。

大小合适的 tempdb 对于优化整体性能十分关键。考虑将 tempdb 的初始大小设为不同于默认值的值，以避免昂贵的文件增长。预先分配的空间取决于预期的工作负载和在 SQL Server 实例中启用的功能。

估计 tempdb 初始大小的一种好方法是分析在典型的工作负载中执行的查询的查询计划。在查询计划中，查询操作符(如排序、哈希匹配和后台打印)能够为计算大小需求提供重要的信息。为了估计每个操作符所需的空间，可以查看操作符报告的行数和行大小。为了计算所需的空间，可以将实际(或估计)的行数乘以估计的行大小。虽然这种方法并不精确，但是却可以提供不错的参考。只有经验和测试才能保证更精确的大小设置。

model 和用户数据库

model 数据库是最常被忽视的系统数据库，是所有用户数据库的模板。换句话说，model 数据库的所有数据库设置都会被 SQL Server 数据库实例中每个新创建的数据库继承。

设置 model 数据库的初始大小、自动增长和恢复模型设置确保了新创建的所有用户数据库都会被恰当地配置，从而优化性能。

设置的初始数据库大小应该足以处理在足够长的时间段内预计发生的事务量。关键是要避免频繁地增加数据库的大小。当最初分配的空间不能满足数据库的需要时，通过启用自动增长数据库设置可以让其自动增加大小。尽管应启用自动增长功能，但该功能很昂贵且耗时。可将自动增长作为一种紧急情况下的操作。如果启用了自动增长，就应选择足够大的文件大小增量，以避免频繁进行自动增长操作。

绝不应该打开自动收缩数据库设置或对数据库计划收缩操作。数据库收缩操作会导致大量等待和阻塞，消耗大量 CPU、内存和 IO 资源，并增加碎片量。

2.6.3　针对安全配置 SQL Server 设置

通过优化 SQL Server 2012 提供的一些系统设置，可以得到可控程度更高、更加安全的环境。接下来将讨论一些最重要的安全设置。

1. SA 账户

SysAdmin(SA)账户是默认的系统账户，在 SQL Server 中拥有顶级权限。这是一个广为人知

的账户，所以是大量攻击的目标。为了避免遭受这种攻击，应该为 SA 账户指定只有你自己知道的强密码，并且从不使用该密码。将这个密码束之高阁，然后禁用 SA 账户。

2. TCP/IP 端口

SQL Server 默认使用 TCP/IP 端口 1433 来与客户端通信，而 SQL Server 命名实例则在服务启动时被动态分配 TCP/IP 端口。为了预防黑客和进行防火墙配置，可能需要修改默认端口，并控制 SQL Server 命名实例在通信时使用的端口号。

SQL Server 2012 中包含了叫做“SQL Server 配置管理器”的工具(本节稍后将更详细介绍)，用于管理 SQL Server 服务和相关的网络配置。在“开始”菜单的“Microsoft SQL Server 2012”|“配置工具”文件夹下可以找到 SQL Server 配置管理器。图 2-12 显示了 SQL Server 配置管理器的“TCP/IP 属性”对话框，在这里可以修改默认的 1433 端口。

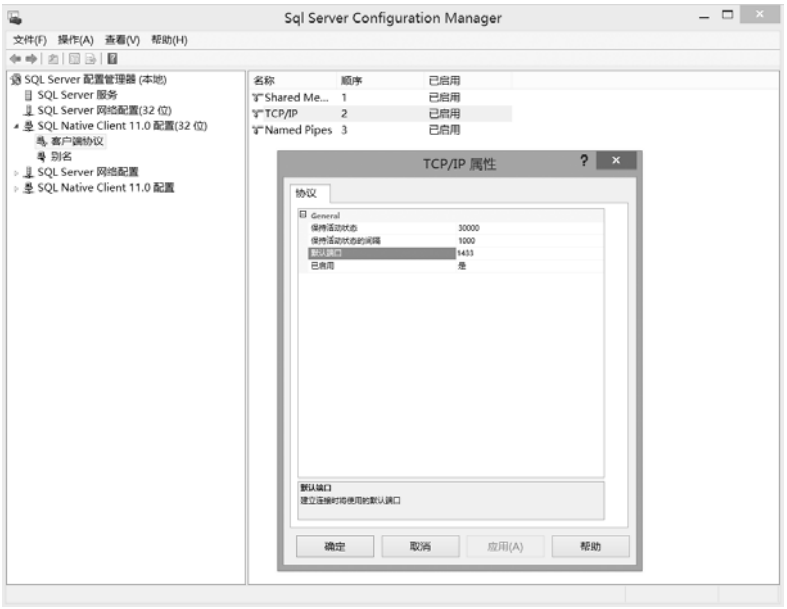

图 2-12

3. 服务补丁和更新

在全新安装 SQL Server 实例后，必须检查可用更新。SQL Server 更新的形式可能是修补程序、累计更新和服务补丁。在应该更新之前，应仔细检查所有更新，以免它们对应用程序产生负面影响。被标记为“关键”的安全补丁是一定要安装的，它们可以防止数据库系统遭受已知的威胁、蠕虫和漏洞的损害。在 SQL Server 生产实例中不要启用自动更新。在把更新应用到生产实例之前，应该在受控的测试环境中测试所有的更新。

4. 其他的 SQL Server 设置

通过 SQL Server Management Studio 和 sp_configure 系统存储过程，还可以配置其他一些 SQL Server 设置和属性。

关于 sp_configure 系统存储过程可以配置的所有 SQL Server 配置选项的完整列表及描述，

可以访问以下网址：http://msdn.microsoft.com/en-us/library/ms188787(v=sql.110).aspx。

第 4 章将详细介绍 SQL Server 配置。

2.6.4　Best Practices Analyzer(BPA)

Microsoft SQL Server 2012 Best Practices Analyzer(BPA)是一个免费的诊断工具，可以收集关于已安装 SQL Server 2012 实例的信息，判断配置是否符合推荐的最佳实践，列出不符合推荐的最佳实践的设置，指出潜在的问题，并推荐相应的解决方案。

BPA 可以帮助识别 SQL Server 2012 安装的潜在问题。在服务器进入生产环境以前就捕获配置问题是最好的，这可以防止出现意外的宕机。

从以下网址可以下载 SQL Server 2012 的 Best Practices Analyzer：http://technet.microsoft.com/en-us/sqlserver/bb671430。

2.6.5　SQL Server 配置管理器

SQL Server 配置管理器用于指定 SQL Server 服务选项，以及这些服务在 Windows 启动后是自动启动还是手动启动。SQL Server 配置管理器允许配置服务账户、网络协议和 SQL Server 监听的端口等服务设置。

SQL Server 配置管理器可以在“开始”|“所有程序”|“Microsoft SQL Server 2012”|“配置工具”文件夹下找到。另外，也可以在“计算机管理”控制台的“服务和应用程序”下找到。

2.6.6　备份

完成 SQL Server 安装后，就应该检查备份计划。必须为系统和用户数据库定义备份计划和备份的存储位置。另外，如果使用了加密，那么还要备份加密密钥。

总是应该在共享网络驱动器或备份设备上创建备份文件，而不应该在被备份的服务器上进行备份。可以考虑保持备份的冗余副本，并保证备份是安全的，在发生灾难情况时可以立即使用。

数据库应该以完全或增量方式备份。取决于数据库恢复模式日志，备份也应该是备份计划的一部分，以便在必要时可以从日志中恢复。更多细节请参阅第 17 章。

定义备份保持策略以避免存储不必要的历史备份。定期还原备份，保证它们在任意时刻都能够成功还原，以免在发生灾难时出现不能还原的情况。记住，好的备份与恢复同等重要。

2.7　卸载 SQL Server

在一些情况下，由于出现了一些问题(例如与新版本不兼容)，或者要进行许可合并，需要彻底卸载 SQL Server 实例。使用“控制面板”的“程序和功能”选项可以卸载 SQL Server。在卸载过程中，可以选择删除为特定实例安装的全部或部分功能。如果安装了多个实例，卸载进程会提示选择想要删除的实例。

在安装程序中安装的某些额外的组件和需求可能不会被卸载，此时需要单独卸载它们。

2.7.1 卸载 Reporting Services

在卸载 Reporting Services 时，需要手动清除一些项，本节将介绍这些项。但在卸载前需要收集一些信息。确保知道 Reporting Services 实例使用了哪些数据库。可通过 Reporting Services 配置工具获得这些信息。通过运行 SQL Server 配置管理器以了解 Reporting Services 实例安装在哪个目录下。还需要知道 Reporting Services 统计和日志文件使用哪个目录。

卸载 Reporting Services 并不会删除 ReportServer 数据库。必须手动删除它们，否则新的 Reporting Services 实例就可以重用它们。

2.7.2 卸载 Analysis Services

卸载 Analysis Services 也需要做一些手动清除工作。在卸载前总是要收集一些信息。通过运行 SQL Server 配置管理器以确定 Analysis Services 实例安装在哪个目录下。

尽管正常的卸载过程未保留任何数据库，但保留了所有 Analysis Services 日志文件。默认位置是 Analysis Services 安装目录或在前面看到的其他位置。要删除它们，只要删除适当的目录即可。

2.7.3 卸载 SQL Server 数据库引擎

和其他服务一样，在卸载 SQL Server 数据库引擎时，不会删除日志文件。要删除它们，只须删除适当的目录。可能需要单独删除 MS SQL Server Native Client，并且可能发现一些目录会被保留下来，需要手动删除。

如果机器上没有其他 SQL Server 实例，就不必单独删除 100 目录，而是可以删除 Program Files 下的整个 MS SQL Server 目录。.NET Framework 也仍会留在机器上。要删除它，可通过“控制面板”的“程序和功能”选项，但要确保没有其他应用程序正在使用它。

2.8 故障排除失败安装

安装失败最常见的原因是安装程序支持规则和安装规则失败。在安装时，会检查一系列规则以识别可能阻止 SQL Server 成功安装的问题。当检测到规则失败时，必须进行纠正，然后才能继续安装。在手动安装过程中，总是会提供规则错误报告的链接和描述，并且会生成错误日志文件以供以后查看。

发生失败时，总是会生成详尽的报告，它们为确认问题的根源提供了宝贵的信息。很多时候，通过安装缺少的功能或应用程序就可以解决这些失败。

从%Program Files%\Microsoft SQL Server\110\Setup Bootstrap\Log 文件夹中可以找到错误报告。每次安装尝试都会生成带有时间戳的文件夹，详细的安装信息将保存到该文件夹下的日志文件中。该日志文件可以帮助对任何错误进行故障排除。关于安装过程中生成的日志文件的完整列表和描述，可以访问网址：http://msdn.microsoft.com/en-us/library/ms143702(v=sql.110).aspx。

2.9　小结

可以看到，SQL Server 2012 的安装过程一般很简单，并且只需要很少用户参与就可以完成。在安装前进行规划是成功部署的关键。成功的 SQL Server 2012 安装始于良好的规划和对需求的准确定义。这些需求应该定义了硬件和软件需求、安装条件、身份验证、排序规则、服务账户、文件位置等。

完成安装程序向导并不意味着 SQL Server 2012 安装就结束了。在安装后还需要执行一些任务，可能需要修改多个默认的配置设置，例如最大内存、并行度阈值、TCP/IP 端口、补丁等。可以使用 SQL Server Management Studio 和 SQL Server 配置管理器来修改这些默认配置选项。还需要对数据库服务器进行压力测试，以免在高负载下出现意外行为。

如果要进行升级，第 3 章将提供一些很好的建议。

第 3 章

升级到 SQL Server 2012 的最佳实践

本章主要内容：

- 升级计划
- 停止支持和弃用的功能
- 选择更新方法
- 升级后的注意事项

第 2 章介绍了执行 SQL Server 2012 全新安装的过程，而本章将讨论升级以前的 SQL Server 版本。成功升级的最佳策略是提前做好计划和准备。首先将说明升级到 SQL Server 2012 的原因，然后讨论各种升级策略的优缺点，并学习可帮助在升级过程中减小风险的各种工具。接着将介绍 SQL Server 2012 的行为变化以及在升级前需要知道的不再支持的功能。然后探讨了在升级后可能遇到的一些问题。相信到本章结束时，读者将具备成功升级到 SQL Server 2012 所需要掌握的一切技能。

3.1 升级到 SQL Server 2012 的原因

本书将介绍 SQL Server 2012 中的重要改进。在开发周期中，要考虑到这个产品的 3 个核心关注点：对关键任务的信心、对技术有突破性见解和使客户能够方便地使用云服务。随着 SQL Server 2012 的发布，Microsoft 在可扩展性、可靠性、可用性以及安全性方面新增了大量功能。这些新功能带来的好处如下：

- 通过 AlwaysOn 实现更有效的高可用性和灾难恢复
- 支持列存储索引
- 内置的加密功能
- 通过使用 Windows Server Core 而减少操作系统补丁

- 新的压缩功能可以提高 I/O 操作的速度
- 组有了默认架构
- 所有版本都能使用 SQL Server 审核功能
- 被包含的数据库身份验证
- 用户定义的服务器角色

3.1.1 减少风险——微软的贡献

和 SQL Server 之前的所有版本一样，SQL Server 团队采取了额外的措施来确保 SQL Server 2012 的高品质。软件工程周期中的特定措施超出了本书的范围，但本节还是会介绍有关每日构建过程的被认为是公共常识的几点内容。

如今，每日构建过程生产的 x86、x64 和安腾版本的 SQL Server 2012 代码经过了大量测试。这个过程不仅用于新版本的开发，还用于开发 SQL Server 2012 服务补丁。这些测试是内部构建测试、客户工作负载和可信任计算过程的汇总。微软研究院(Microsoft Research)不断地为 Microsoft 的产品带来创新。在软件开发方面，微软研究院团队主要致力于软件工程和测试过程。他们改进测试装置，并在多个方面进行增强，包括威胁建模、测试效率以及渗透分析。

另外，许多客户工作负载也是软件测试过程的一部分。这些工作负载都通过一些程序(如客户重放计划)和不同的实验室(包括 SQL Server 2012 兼容性实验室)获得。

每日构建将针对这些收集的信息进行测试，而结果可获得性能度量值、安全度量值以及故障信息。然后故障被归档、分配、排定优先级并跟踪，直至最后被解决。一旦修正了故障，其代码就会作为软件工程过程的一部分进入安全性测试。这些都发生在代码嵌入软件树而进入下一次测试循环之前。这种严格的开发和质量保证过程帮助确保交互的产品十分可靠，为生产环境做好了准备。那句老话，“等到有了第一个服务补丁后再进行升级”，对于 SQL Server 2012 是不适用的。

3.1.2 独立软件厂商和 SQL 社区的贡献

从 SQL Server 2005 开始到 SQL Server 2012，都采纳了社区技术预览(Community Technology Preview，CTP)的概念。除了发布候选版本(Release Candidate，RC)以外，November 2010 CTP 是几个这种版本中的第一个。对代码(或构建)采用这种时间点快照的决策导致了超过数十万次的 CTP 和 RC 下载，为独立软件供应商(ISV)和 SQL 社区测试提供了对更新代码的史无前例的访问。在本书创作时，Microsoft 已经发布了 10 个案例分析，详细说明了 SQL Server 2012 的成功实现。这种对测试代码的访问作为一种标识额外故障、执行额外软件修正测试和根据社区反馈进行额外改进的手段。

3.2 升级到 SQL Server 2012

第 2 章已介绍过安装指导，因此本节主要介绍升级策略和有关 SQL Server 2012 数据库组件的考虑事项。平滑升级要求有好的计划。在设计升级计划时，需要将升级过程分为独立的任务。这个计划应包括升级前任务、升级任务和升级后任务几个部分。

- 升级前任务应考虑 SQL Server 2012 最低的硬件和和软件需求。应有访问服务器的应用程序清单、数据库排序规则需求、服务器依赖性以及遗留系统需求(如数据访问方法)信息。列表中应包括数据库一致性检查和所有数据库的备份。计划还应考虑测试升级过程和应用程序。你应对向后兼容性问题十分了解和有确定的替代或解决方案。还应使用本章后面要介绍的 SQL Server 2012 升级顾问来帮助找出并解决问题。
- 升级执行过程应是平滑执行有良好文档的预演计划的过程。为了重申此步骤的重要性，务必在执行升级过程前备份所有数据库。
- 升级后任务应包括检查升级过程、使系统恢复到联机状态、监控和测试系统。在向用户社区发布系统前还需要执行一些特定的数据库维护，本章后面将介绍这些步骤及其他推荐步骤。我们推荐在升级之后在向后兼容模式下运行数据库，以便最小化环境变化。数据库兼容模式的更新连同启用 SQL Server 2012 新功能应作为后续升级过程的一部分。

在确定升级策略方面，本节将介绍两种升级方法：本地升级和并列迁移。

3.2.1　本地升级

本地服务器升级是这两种方法中较容易的一种，但是风险也要大一些。这是一种要么完全成功、要么完全失败的升级方法，这意味着一旦启动升级，就没有简单的回滚过程。这种升级要求有更好的前期测试，以避免使用复杂的回退计划。这种方法的好处是不用担心用户和登录名保持同步的问题，对于应用程序也不要求数据库连接改变。另外，SQL Sever Agent 作业将在升级过程中迁移。

这里根据图 3-1 介绍如下本地升级场景：

(1) 首先，在系统上安装必要的文件。在升级到 SQL Server 2012 之前，服务器至少需要：

- .NET Framework 4.0
- Windows PowerShell 2.0
- .NET 3.5 with Service Pack 1
- SQL Server 2005、SQL Server 2008 或 SQL Server 2008 R2 的当前实例

(2) 下一步是运行系统配置检查器(System Configuration Checker，SCC)。SCC 检查目的计算机是否存在阻碍升级完成的问题，比如不满足最低的硬件或软件要求。如果有这种问题，就退出安装程序并卸载 SQL Server 2012 组件。

(3) 一旦通过验证，SQL Server 安装程序就可以在磁盘上部署 2012 文件和向后兼容性支持文件，同时 SQL Server 2008(或 2005)仍可用。但并不推荐计划升级服务器时仍使用户联机。安装程序通过停止现有 SQL Server 服务，可使服务器脱机。基于 SQL Server 2012 的服务假定有 master 数据库的控制权和服务器身份。这时，SQL Server 服务会接管数据库并开始更新它们，同时不允许用户返回到环境中。当数据请求在只局部更新的数据库中发生时，将会更新、处理与请求相关的数据，然后返回给用户。

(4) 最后，卸载旧的二进制文件。这一步骤仅在服务器上没有遗留的 SQL Server 2005 或 2008 实例时才执行。最后，SQL Server Agent 作业得到迁移。

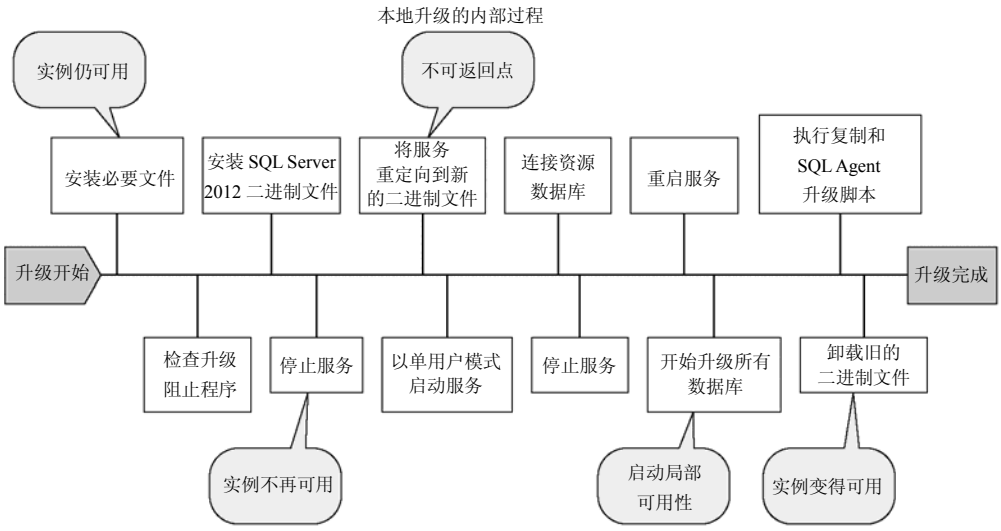

图 3-1

本地升级的优点如下：

- 快速、方便、自动(最适合小型系统)
- 不需要额外的硬件
- 应用程序保持相同的实例名
- 自动保留 SQL Server 2008(或 2005)的功能

本地升级的缺点如下：

- 导致宕机时间，因为整个服务器在升级时处于脱机状态
- 不支持组件级升级
- 复杂的回滚策略
- 必须为 SQL Server 实例解决向后兼容问题
- 不是所有 SQL Server 组件都可以本地升级
- 大型数据库需要较长的回滚时间

另外，如果想在升级过程中更改版本，那么有一些必须了解的限制。SQL Server 2005 和 2008 企业版、开发人员版、标准版以及工作组版可升级到 SQL Server 2012 的不同版本。但是，SQL Server 2005 和 2008 速成版只能升级到 SQL Server 2012 速成版。如果对此内容感兴趣，可查看 SQL Server 2012 联机丛书(BOL)中“升级到 SQL Server 2012”下的主题“版本升级”。

3.2.2 并列升级

在并列升级中，SQL Server 2012 可以作为单独的实例与 SQL Server 2008(或 2005)一起安装，或是在一台不同的服务器上安装。这个过程实质上就是在全新安装后进行数据库迁移。在硬件更新或迁移到新平台(如安腾或 x64)的过程中，可选择这种方法。由于回退场景中会多次涉及备份和还原，因此如果有相当大的数据库，那么这种方法显然是较好的选择。

作为这种方法的一部分，可从原始服务器备份数据库，然后将它们还原到 SQL Server 2012 实例。另一种选择是手动从旧实例中分离数据库，然后使用日志传送或数据库镜像将它们重新

附加到新实例。也可利用复制数据库向导将数据库迁移到新服务器。尽管这种方法提供了最好的恢复场景，但和本地升级相比，还有一些额外要求，如维护原服务器名称、注意应用程序连接性，使用户和登录名保持同步。

支持并列升级的理由如下：

- 更细粒度地控制组件级升级过程(数据库、Analysis Services 及其他)
- 可以并列运行 SQL Server 以进行测试和验证
- 可以收集升级的实时矩阵(中断窗口)
- 回滚策略，使得原服务器仍保持不变
- 最适合于超大型数据库，因为还原时间可调整

反对并列升级的理由如下：

- 不保留 SQL Server 2008(或 2005)的功能
- 连接应用程序时实例名的问题

3.2.3　本地升级与并列升级的考虑事项

在选择升级策略前，应考虑许多因素。升级策略应包括组件级升级需求、失败时的回滚能力、数据库大小及局部升级的需求。对于大部分人来说，最主要的考虑包括是否可以升级到新硬件、是否有助于改变策略(如服务器合并)，以及是否能为升级提供较小的服务器中断窗口。表 3-1 对这两种升级方法做了汇总。

表 3-1　比较本地升级和并列升级

过　　程	本 地 升 级	并 列 升 级
结果实例数	1 个	2 个
数据文件传输	自动	手动
SQL Server 实例配置	自动	手动
支持升级实用程序	SQL Server 安装	各种迁移和数据传输方法

3.3　升级前的操作步骤和可用工具

现在你已经知道了升级到 SQL Server 2012 的理由和选项，接下来就可以选择升级工具来辅助升级过程，并执行升级前所需的步骤。在升级前，可以采取一些预防措施来避免常见的升级问题，例如磁盘空间不足或者在升级过程中执行启动存储过程。有许多工具可以帮助识别环境中的潜在升级问题。最常用的两个工具是 SQL Server 2012 升级顾问和 SQL Server 2012 升级助手。这些工具都可以提供升级前分析，并帮助你确信升级可以成功完成。升级助手使用负载测试来测试升级后的应用程序行为，而升级顾问则对数据库执行本地分析以找出潜在的兼容性问题。

3.3.1　升级前的步骤

在进行升级前，需要执行一些步骤。这些预防措施可以防止升级过程中出现令人不愉快的

问题。

- 在升级过程中，将数据文件和日志文件设为自动增长。
- 禁用所有的启动存储过程，因为升级过程会在被升级的 SQL Server 实例上停止和启动服务。
- 在升级到 SQL Server 2012 前，使用 sp_dropextendedproc 和 sp_addextendedproc 存储过程重新注册没有用完整路径名注册的扩展存储过程。

tempdb 负责管理临时对象、行版本控制和联机索引重建。

- 在升级到 SQL Server 2012 前禁用所有跟踪标志。跟踪标志功能在 SQL Server 2012 中发生变化或根本不存在的可能性是存在的。在升级后，应该与 Microsoft 支持团队协作，确定哪些跟踪标志仍然需要(或者全部不再需要)。
- 迁移到数据库邮件。对 SQL 邮件的支持已被停止。

3.3.2 升级前的工具

升级过程可能是一项令人望而生畏的任务。通过在进行升级前检查实例，可以降低升级失败或在升级后出现异常行为的风险。在准备进行升级时，有两个工具可以考虑：SQL Server 2012 升级顾问和 SQL Server 2012 升级助手。

1. SQL Server 2012 升级顾问

升级顾问检查的规则代表可能对升级到 SQL Server 2012 的过程产生影响的条件、情形或已知的错误。如果想吸取别人的升级经验，那么 SQL Server 2012 升级顾问是个不错的选择。该工具基于之前采用者的反馈和内部实验室测试的反馈。SQL Server 2012 升级顾问可以作为 Microsoft SQL Server 2012 功能包的一部分，从 www.microsoft.com/downloads/en/details.aspx?id=26726 免费下载。另外也包含在所有版本的 SQL Server 2012 安装程序中。这个工具的目的是要标识已知的升级问题，并对在每个服务器组件上标识出的问题提供解决方法或修正措施。Microsoft 努力开发这个工具以帮助减少 SQL Server 2005 和 SQL Server 2008 用户升级到 SQL Server 2012 的风险。因此，不管运行的是 Analysis Services、Integration Services、Reporting Services 组件还是这些组件的组合，升级顾问工具都可以提供帮助。

安装 SQL Server 2012 升级顾问

升级顾问是个相对易于使用的工具。可以从安装盘默认屏幕的“准备”部分找到，也可以从 www.microsoft.com/download/en/details.aspx?id=26726 下载。升级顾问的欢迎界面如图 3-2 所示。选择“检查更新”选项，因为可以在线获得此工具的升级版本。

这个工具会不断更新，从而反映之前进行升级的 DBA 所获得的经验。安装它需要有.NET 4.0，可以从 Windows 更新服务或 MSDN 下载.NET 4.0。另外，在安装升级顾问前，必须安装 Microsoft SQL Server Transact-SQL ScriptDom，其网址为 www.microsoft.com/download/en/details.aspx? id=26726。也可选择安装该工具的单个实例和版本，在企业中测试所有服务器。该选项通过对服务器的只读访问来支持零空间占用(zero-footprint)询问。

图 3-2

该工具是读操作密集型的，应在测试服务器上进行测试以评估对系统的潜在影响。

安装过程非常简单，唯一要做的是选择该工具的安装位置。默认安装路径是 C:\Program Files(x86)\Microsoft SQL Server Upgrade Advisor。

使用升级顾问

在安装后，升级顾问将提供两个选项：升级顾问分析向导和升级顾问报表查看器。选择升级顾问分析向导，如图 3-3 所示，只需要选择服务器和组件作升级分析，或单击“检测”按钮启动检测过程，就可以检测系统中安装的组件。

图 3-3

选择要测试的组件之后，接下来选择要进行升级评估的数据库，如图 3-4 所示。这个过程

中最有利的部分是可选择分析跟踪文件和 SQL 批处理文件来帮助进行全面分析。也就是说，通过在评估过程中添加这些文件，升级顾问不仅评估数据库，还评估跟踪工作负载和 SQL 脚本。通过评估这些额外的信息，升级顾问不仅可以评估数据库的当前信息，还可以评估跟踪文件和批处理文件中包含的过去的数据库的使用和行为信息。你只须选择跟踪文件或批处理文件所在的目录路径。

图 3-4

在完成要评估组件的配置后，将提示开始分析。如果在配置过程中存在任何问题，通过“帮助”按钮可获得“与升级顾问相关的联机丛书”(UABOL)的内容，其中包含了大量相关信息和指导。完成组件级分析后，将出现绿色、黄色或红色的对话框以表明测试结果。

在测试结束后，通过升级顾问报表查看器查看发现的问题。如图 3-5 所示，报表本身在类似于 Web 浏览器的界面中显示。可按服务器、实例、组件或问题类型对报表进行筛选，以分析这些信息。本章后面将介绍如何理解该报表的结果。

图 3-5

脚本化升级顾问

如果你有服务器场或更喜欢使用脚本，那么还可以使用命令行功能。利用 UpgradeAdvisorWizardCmd 实用程序，通过 XML 配置文件来配置该工具，并得到 XML 文件形式的结果。下列参数可传递给 UpgradeAdvisorWizardCmd 实用程序：

- help 命令行参数。
- 配置文件路径和文件名。
- 用于 SQL Server 的 SQL Server 登录名和密码(假定使用 SQL Server 身份验证而不是 Windows 身份验证)。
- 表明是否以逗号分隔值(CSV)的形式输出报表的可选标志。

向导部分中介绍的所有功能和参数都可通过配置文件提供。通过 XML 文档或 Excel(如果使用 CSV 选项)，命令行方式执行的结果仍可在报表查看器中查看。例如，下列来自升级顾问的 XML 文档反映了在名为 SQL12Demo 的服务器和名为 SQL2012 的实例上分析所有数据库、Analysis Services 和 SSIS 包的选择：

```
<Configuration>
  <Server>SQL12Demo</Server>
  <Instance>SQL2012</Instance>
 <Components>
   <SQLServer>
     <Databases>
       <Database>*</Database>
     </Databases>
   </SQLServer>
 </Components>
</Configuration>
```

可在 XML 编辑器(如 Visual Studio)中修改此文件并用新文件名保存该文件。然后，可使用该新文件作为升级顾问命令行工具的输入。例如，下面显示的命令提示符使用 Windows 身份验证运行升级顾问命令行。配置文件已包含名为 SQL2012 的远程服务器名和名为 SQL2012 的实例名，PATH 环境变量包含升级顾问的路径：

```
C:\>UpgradeAdvisorWizardCmd -ConfigFile "SQL2012Config.xml"
```

还可以通过命令提示符安装或删除升级顾问应用程序。这样无论是否有 UI，都可控制安装过程，还可配置安装路径和进程日志选项。

要了解有关升级顾问的配置文件的更多信息，可以查看升级顾问帮助的“Upgrade-AdvisorWizardCmd 实用工具” 部分。

解决升级问题

升级顾问的报表包含大量信息。关键是理解这些信息如何显示、需要解决哪些问题以及何时解决。如图 3-5 所示，第 1 列表明结果或推荐方案的重要性，第 2 列说明需要何时解决，而第 3 列说明了有关问题的信息。推荐的分析方法是首先按重要性和解决时间对信息进行分类。具体地说，它们将表明问题应在升级前还是升级后解决。表 3-2 列出了作者针对何时解决这些问题提供的建议。

表 3-2　何时解决升级问题

重 要 性	何 时 解 决	作 者 建 议
红色	升级前	升级前解决
红色	任何时候	升级前解决
红色	升级后	升级后解决
黄色	任何时候	升级后解决
黄色	升级后	升级后解决
黄色	取决于升级顾问	升级后解决

对于“重要性”为红色且“何时解决”为“升级前”或“任何时候”的问题，应在升级前解决。通常，由于 SQL Server 2012 功能发生改变，例如不再支持某些功能，这些问题需要进行补救。其他问题一般可在升级后解决，因为这些问题可能在升级过程中有替代方案或是根本没有受到影响。如果将错误展开，就将显示其他信息，如图 3-6 所示。

图 3-6

“显示受影响的对象”链接显示了升级顾问标记的受影响的对象，而“请告诉我有关此问题的详细信息以及如何解决”链接将显示 UABOL 的相应部分。UABOL 描述了相关条件，并提供关于如何解决这一问题的指导。UABOL 非常有用，因为它对该工具无法解决的问题的解决方案提供了指导(如复制、SQL Server 代理和全文搜索)。

“此问题已解决”复选框用于跟踪已解决的问题。这个元数据复选框用于支持补救过程，允许按问题的状态(已解决或未解决)进行筛选来查看报表。

如果你更喜欢命令行脚本，将知道查看器只是应用于 XML 结果文件(位于 My Documents\ SQL Server 2012 Update Advisor Reports\目录中)的 XSLT 转换。可在每个基于服务器名称的目录中找到各个组件的结果和配置文件，甚至还可将基于查看器的报表导出为其他输出格式，如 CSV 或文本。

2. SQL Server 2012 升级助手(Upgrade Assistant for SQL Server 2012，UAFS)

UAFS 最初用于 SQL Server 2005 应用程序兼容性实验室，是作为 Microsoft Ascend(SQL 2005 客户培训)和 Touchdown(SQL 2005 合作伙伴培训)项目的一部分运行。这些实验室的目的是帮助客户分析 SQL Server 2000(或 7.0)应用程序，以使他们了解升级到 SQL Server 2005 的影响并提供有关为成功迁移数据库和应用程序必须做哪些修改的指导。这些实验室通过对 SQL Server 2005 进行升级和对客户的实际工作负载进行影响分析，来帮助改进 SQL Server 2005 的质量。这些实验室本身由 Microsoft 及其合作伙伴(如 Scalability Experts)的员工负责。全球大约有 50 个实验室，使用这一工具测试了数百个 SQL Server 2000 应用程序。他们还为 SQL Server 2008 特别开发了一个 UAFS 新版本，并针对 SQL Server 2012 进行了更新；可以从 www.scalabilityexperts.com/tools/downloads.html 免费下载。

从概念上来看，该工具与升级顾问的区别是 UAFS 包括了真正的升级和测试方法体系。通过对相同负载在 SQL Server 2008(或 2005)和 SQL Server 2012 中运行的结果进行比较，可以确定升级阻止程序及需要的应用程序代码修改。对于 SQL Server 2012 来说， UAFS 工具支持从 SQL Server 2008 和 2005 进行升级。

下面将概述这个过程，然后详细介绍 UAFS 中的每一步操作，如图 3-7 所示。通过使用 UAFS，将备份所有数据库和用户并捕捉部分生产负载。然后还原备份的数据库和用户，并处理捕捉的工作负载。这样做的目的是生成新的输出文件，也就是基准文件。接着，将测试服务器升级到 SQL Server 2012，重新运行该工作负载以捕捉 SQL Server 2012 参考输出并作比较。

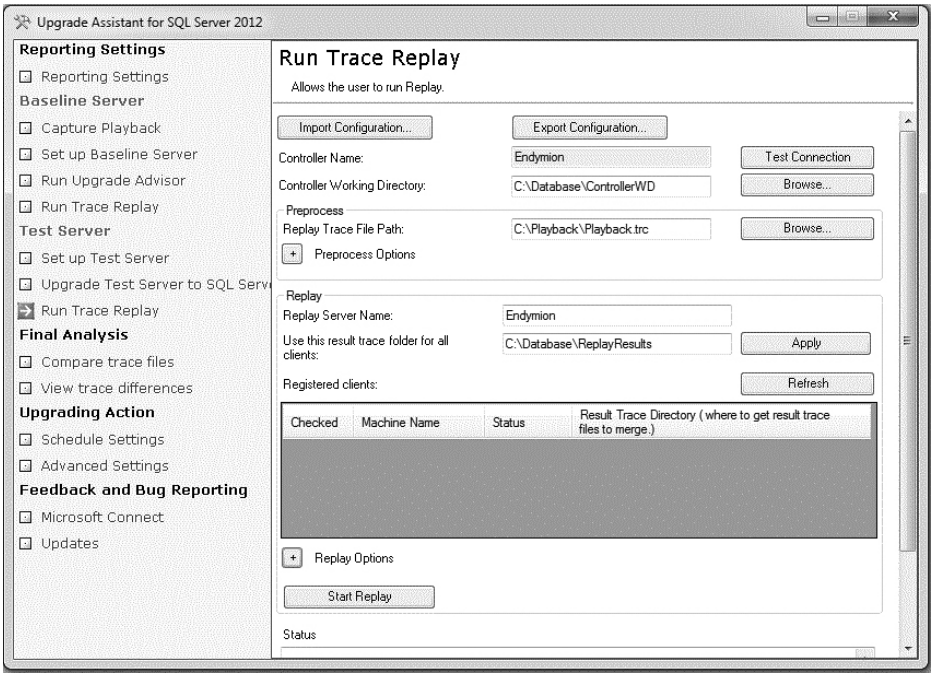

图 3-7

捕捉环境

应通过备份服务器中的所有 SQL Server 2008(或 2005)系统和用户数据库的方法来建立基准。接着，需要开始捕捉跟踪文件以避免进程中出现缺口。在捕捉跟踪文件时，需要良好地表

示描述工作环境特征的负载。因此，可能需要创建人工负载，以便更好地表示应用程序工作负载随时间推移出现的情况。捕捉跟踪文件时，最好避免多服务器操作，如链接服务器调用或批量复制操作依赖性。要知道，跟踪时将导致性能开销。表示可重复、可重用的工作负载的跟踪文件和数据库备份称为重放(playback)。

建立基准服务器

在捕捉了重放后，可建立用于其余测试的基准系统。这个服务器应安装了 SQL Server 2008 R2 SP1、2008 SP2 或 SQL Server 2005 SP2，满足升级到 SQL Server 2012 的最低要求。实际上，应采用与源系统相同的排序规则和补丁程序。然后，该工具将检查服务器的这一匹配情况。如果有必要，将提示安装补丁程序或重建 master 数据库。接下来是以正确的顺序还原数据库，使 DB ID 与生产环境匹配(这也包括了添加 DB 创建过程来实现该目标)。最后，UAFS 将重新创建登录名，并确保 ID 与生产环境匹配，因为这些都是运行跟踪文件所必需的。该过程的下一步是按本章前面所述运行升级顾问。在修复好环境后，就可以进行重放跟踪。

运行跟踪

当运行跟踪时，首先更新所有数据库中的统计信息，然后重放工具使用 API，并依次运行跟踪文件中的所有查询。该工具是单线程重放，但可能会出现阻塞。如果跟踪运行得非常缓慢或停止，就应检查 SQL Server 阻塞情况；如果不能自动清理，就需要终止阻塞进程。该步骤会生成跟踪输出文件，用于在最终分析时进行比较。

升级到 SQL Server 2012

现在准备升级到 SQL Server 2012，可以有两种选择。可以使用 UAFS 将 SQL Server 2008(或 2005)的状态还原到基准状态，然后本地升级到 SQL Server 2012。也可以将 SQL Server 2008(或 2005)迁移到现有 SQL Server 2012 实例。如本章前面所述，执行本地升级还是并列升级的决定是根据环境特定的一系列因素做出的。这里不是在测量性能指标，因此这些服务器不必相同。这是在两种 SQL Server 版本间测量工作负载行为。在 SQL Server 2012 平台上还原基准后，将再次进行“运行跟踪”步骤，但这次是在 SQL Server 2012 上。该步骤生成另一跟踪输出文件，用于在最终分析时进行比较。

最终分析

在完成所有这些过程之后，就进入比较输出文件的最终步骤，方法是筛选并比较两个跟踪文件中的所有批处理结果，查看它们的不同点。报表查看器一次显示一个错误条件，显示了最后一个正确的步骤、错误的步骤和接下来的正确批处理文件序列。一旦标识出错误条件，就可以在错误检查过程中筛选出该错误，使 DBA 专注于查找新的错误条件。在 SQL Server 2012 升级完成后，将数据库兼容性模式改为 110 并运行应用程序以验证能在 SQL Server 2012 兼容性模式下工作。这确保了当在数据库兼容性模式为 110 的情况下运行时，不存在应用程序行为差异。

3.4 向后兼容性

本节介绍 SQL Server 2012 出现的主要变化，它们分为三类：不支持、不推荐使用的变化和影响 SQL Server 2008 或 2005 行为方式的变化。尽管升级顾问工具会在条件与环境相关时突出

显示它们，但读者还是应阅读本节来了解这些改变。

3.4.1　SQL Server 2012 中不支持和未延续的功能

有时为了技术向前发展，必须做些权衡。从 SQL Server 2008 到 SQL Server 2012，下列功能不再可用：

- 系统存储过程 sp_ActiveDirectory_Obj、sp_ActiveDirectory_SCP 和 sp_ActiveDirectory_Start。
- sp_configure 选项 user instance timeout 和 user instances enabled。
- 对 VIA 协议的支持。
- SQL 邮件(改用数据库邮件)。
- 使用 sp_addserver 创建新的远程服务器(改用链接服务器)。
- 数据库兼容性模式 80。
- RESTORE {DATABASE | LOG} … WITH DBO_ONLY(改用 WITH RESTRICTED USER 子句)。

这个列表并不完整。关于未延续和弃用功能的完整列表，可访问以下网址：http://msdn.microsoft.com/en-us/library/cc280407(v=SQL.110).aspx。

3.4.2　SQL Server 2012 弃用的数据库功能

这些功能在 SQL Server 2012 或后续版本中不再可用。下面列出的是一些将要弃用的功能，建议你逐步用推荐功能替换这些功能：

- 使用 CREATE ENDPOINT 和 ALTER ENDPOINT 创建的 SOAP/HTTP 端点(它们已被 Windows Communication Framework (WCF)或 ASP.NET 取代)。
- 兼容性模式 90 在 SQL Server 2012 后不再可用。
- 使用 RC4 或 RC4_128 的加密计划在下一个版本中被移除。考虑改用另一种加密算法，例如 AES。
- 在 SQL Server 将来的版本中，将不再支持 T-SQL 不以分号结束的情况。

3.4.3　SQL Server 2012 中其他影响行为的变化

下面这些功能的行为变化可能会影响向 SQL Server 2012 迁移：

- 如果通过复制已有作业的脚本来创建新作业，新的作业可能会对已有作业产生负面影响。这是因为参数@schedule_uid 不应被复制。应该在新作业的脚本中手动删除该参数。
- 现在，确定性的标量值 CLR 用户定义函数和确定性的 CLR 用户定义类型的方法是可被缓存的。其目的是当这些函数或方法被多次以相同参数调用时提高性能。但是，如果非确定性函数或方法被错误地标记为确定性的，那么可能产生意外的结果。
- 当带有分区索引的数据库升级时，这些索引的直方图数据中可能存在变化。这是因为 SQL Server 2012 使用默认的采样算法而不是完整扫描来生成统计数据。
- 在 SQL Server 2012 中，在 XML 模式下使用 sqlcmd.exe 的方式发生了变化。

要了解有关行为变化的更多内容，可以参阅 SQL Server 2012 联机丛书或访问以下网址：http://msdn.microsoft.com/en-us/library/cc707785(v=SQL.110).aspx。

3.5 SQL Server 组件的考虑事项

在本节中将介绍各个组件以及在升级过程中对它们应做的考虑。这里未涉及的组件将在相应的章节中介绍。

3.5.1 升级全文目录

在升级过程中，带有全文目录的所有数据库都应标记为禁用全文功能，原因是重建全文目录可能需要较长时间。在升级全文搜索环境前，应熟悉一些改进之处。数据库的附加和分离过程也会导致全文目录被标记为禁用全文功能。建议读者阅读 SQL Server 2012 联机丛书来了解其他行为变化。

3.5.2 升级 Reporting Services

可以从 Reporting Services 2008 和 Reporting Services 2005 升级到 Reporting Services 2012。Reporting Services 2012 将支持 SQL Server 上的报表数据库。在升级前，运行 SQL Server 2012 升级顾问并遵循其建议、对可能的迁移选项的指导以及步骤。然后在执行升级前，备份数据库、应用程序、配置文件以及加密密钥。

升级 Reporting Services 的方法有两种：

- **本地升级**：通过执行 SQL Server 2012 setup.exe 来实现；选择旧版 Reporting Services，将其升级为 Reporting Services 2012。这种方法的优缺点与前面介绍的本地升级相同。风险在于这是完全成功或完全失败型的方法，难以回滚，除非重新安装。
- **并列升级**：指的是 Reporting Services 2012 实例与 Reporting Services 旧版本安装在同一台物理服务器上或是安装在单独的服务器上。在安装好 Reporting Services 2012 实例后，报表内容使用下列选项之一单独或批量迁移：
  - 使用 SQL Server 2012 Business Intelligence Development Studio 重新部署报表
  - 使用 rs.exe 提取和部署报表
  - 使用报表管理器

发布的报表和快照报表都将得到升级。在升级 Reporting Services 2012 后，重新部署自定义扩展和程序集，在完善后的 Reporting Services 2012 上测试应用程序，然后从之前的版本中删除未用的应用程序和工具。

第 19 章将详细讨论如何升级已有的群集。第 16 章将详细讨论如何升级数据库镜像。

3.5.3　升级到 64 位

不支持将 32 位的 SQL Server 2005 或 2008 升级到 SQL Server 2012 x64 位平台。虽然在 Windows x64 位子系统上支持运行采用 Service Pack 2 的 32 位 SQL Server 2005 或 SQL Server 2008 平台，但并不支持将这种配置升级到 SQL Server 2012 x64 位环境。只有并列迁移支持将数据库从 32 位迁移到 x64 位平台。

3.6　升级后检查

下面介绍使很多人感到惊讶的产品行为。在成功地升级了环境后，由于查询性能差而遭受最终用户的责备并不是愉快的事情。对升级后的问题提前采取措施，可以降低精心规划和努力工作被安装后的问题破坏的风险。

升级后查询性能较差

升级到 SQL Server 2012 之后导致查询性能较差的一个可能原因是，原来的统计信息过时，不能为查询优化器使用。对于大多数情况来说，只要启用了自动更新统计信息和自动创建统计信息的选项，这就不会成为问题。这样一来，在查询编译时，如果需要，就将默认自动更新统计信息。通过这些功能获取的统计信息只来自样本数据。因此，与从整个数据集获取的统计信息相比，它们的精确性要差些。在带有大型表的数据库中或在以前用全扫描创建的统计信息表中，这种质量差异可能导致 SQL Server 2012 查询优化器生成非最优的查询计划。

在 SQL Server 2012 中创建索引时，统计信息会使用查询优化器的默认抽样算法。

为了解决这个问题，应在升级到 SQL Server 2012 后立即更新统计信息。使用带有参数 resample 的 sp_updatestats 将根据继承的抽样率重构所有已有的统计信息。通常，对于基于索引的统计信息，可以得到全部样本结果，对其他列则得到抽样统计信息。从该过程可得到的额外好处是，如果数据小于 8MB(最小抽样大小)，将用全扫描生成统计信息。

3.7　小结

本章介绍了升级到 SQL Server 2012 的充分理由，并回顾了一些策略和工具，如 SQL Server 2012 升级助手和 SQL Server 2012 升级顾问，它们都可以在升级时运用。接着，本章讨论了升级过程，包括成功升级的升级前、升级中和升级后的步骤，指出了一些已不再推荐使用的功能(如数据库兼容性模式 80 和 SQL 邮件)以及行为变化可能影响升级的功能(如升级后的分区索引直方图和 XML 模式下的 sqlcmd.exe)。现在，我们已为成功升级打下了坚实基础，接下来将开始进入 SQL Server 2012 的世界。