

SQL Server 十大技巧集锦

2013 年 SQL Server 十大技巧集锦

新年干货大放送。TechTarget 数据库网站总结了过去一年最实用、最受欢迎的十大 SQL Server 技巧，分享给各位 DBA。预祝 DBA 们马年“马上有成功”！

✧ SQL Server 版本指南

摘要：本文讨论了 SQL Server 企业版、免费版和标准版三个版本的数据库平台，在选择的时候需要注意哪些问题。

✧ SQL Server 2012：你应该掌握的 6 个管理工具

摘要：为了满足企业数据库管理的新需求，微软公司发布的最新版 SQL Server 中包括了不少图形化的管理工具，可以帮助 DBA 与开发人员更高效地创建、管理和维护 SQL Server 解决方案，使得他们能够快速解决复杂的性能与配置问题。

✧ 如何制定完美的 SQL Server 备份策略

摘要：如果没有相应的规划，数据库恢复过程就无法成功完成，DBA 也可能因此丢掉工作。在本文中，我将介绍一些关于开发、创建和测试备份策略的概念。

✧ SQL Server 2014 概述

摘要：新版本的 SQL Server 2014 将在 2014 年初发布。本文简单介绍了 SQL Server 2014 中的一些新特性。如内存 OLTP 以及云计算等。

✧ SQL Server 2012 安全性解析

摘要：SQL Server 2012 增加了许多安全功能和增强功能。这些增强功能不只可以帮助组织改善数据访问控制，而且可以实现最高级别的数据保护和遵守。

◇ SQL Server 升级三部曲

摘要：SQL Server 版本升级规划是每个数据库管理员都渴望承接的一项任务。然而，升级 SQL Server 并不是一件容易的差事，规划不当会增加升级失败的风险。我们在设计 SQL Server 升级路线时，有必要把任务分成三步：升级前、升级中、升级后。

◇ 总结 SQL Server 虚拟化的几点注意事项

摘要：服务器虚拟化已经出现好多年了，技术已趋于成熟。几年前，DBA 还很抵触把 SQL Server 迁移到虚拟机，但是现在已经是再平常不过的了。然而，在决定虚拟化之前有一些重要的事情需要考虑。

◇ 深入了解 SQL Server 审计工具

摘要：微软公司在发布 SQL Server 2008 的时候引入了 SQL Server 审计 (Audit)，这是一款功能全面的数据库审计工具，弥补了 SQL Server 早期版本中的许多审计功能限制。

◇ SQL Server Data Tools 2012 让数据库开发更容易

摘要：SQL Server Data Tools (SSDT) 2012 提供了创建分析服务，报告服务和集成服务项目所必需的模板和组件。

◇ 如何创建 SQL Server 视图

摘要：本文介绍 SQL Server 视图并给出它们是如何工作的示例。这些示例基于在 SQL Server 2012 的一个本地实例上所创建的 AdventureWorks2012 数据库中的两个表。

SQL Server 版本指南

本文共分为三个部分，分别讨论 SQL Server 企业版、免费版和标准版三个版本的数据库平台，在选择的时候需要注意哪些问题。

企业版数据库

可以想象，企业版包含了所有标准版的功能，还包含更多的特性。和标准版一样，企业版支持最高 524 PB 的数据库。然而，企业版支持主系统所能提供的最大 RAM 和处理器内核数，从而进一步提升了可扩展性和性能。而且，企业版还包含一些扩展性特性，如表与索引分区、数据压缩和并行分区表。企业版也具有更高的可用性。您可以创建数据库快照、AlwaysOn 可用性分组、多子网集群和镜像备份等。

企业版超越标准版的另一个方面是安全性。与标准版相似，企业版支持基本审计、包含数据库和用户自定义角色。但是企业版还支持精细审计、透明数据库加密和可扩展密钥管理。此外，企业版还支持端到端事务复制、并行索引操作和自动使用索引视图。

商业智能 (BI) 方面也是企业版领先。它包含面向各种集成服务的高级适配器，如面向 Oracle 和 Teradata 的高性能目标组件，以及面向 SAP BW 的源与目标组件。企业版还带有主数据服务 (Master Data Services) 和面向 SharePoint 的 PowerPivot (PowerPivot for SharePoint)。主数据服务有一个中央数据中心，可以帮助保证整个组织的数据完整性和可管理性。面向 SharePoint 的 PowerPivot 则提供了支持 SharePoint 环境的高级数据分析功能。

此外，企业版支持分析服务 (Analysis Services) 的扁平化 BI 语义模型，其中包括为数众多的数据挖掘特性，如交叉验证、并行模型处理、顺序预报和插件算法支持。此外，SQL Server 还推出了商业智能版本，它也支持许多此类特性。

显然，在可用性、可扩展性、可靠性和 BI 方面，企业版都优于其他版本的 SQL Server。但是，您需要为这些特性支付额外费用。与标准版不同，企业版只提供了基于内核的授权方式，而且其价格高于标准版。企业版的每一个内核花费约 6,900 美元。如果有 16 个内核，那么费用接近 110,000 美元。这对于大多数组织是一笔不菲的投入，而在标准版中，这些价格不会计入渠道与容量授权费用之中。

无论选择哪一种 SQL Server 版本，一定要保证您了解所使用的技术。同样，要阅读 MSDN 的文章，详细了解每一个 SQL Server 版本的特性。在进行选择时，一定要同时考虑当时需求和长期需求。当然，您可以先选择一个较为基础的版本，然后根据需要进行升级，但是升级本身需要消耗一定的时间和资源，更不用说产品费用。它们不一定能和想象一样平稳升级。所以，要仔细考虑您的选择，然后再确定哪一个 SQL Server 最适合您使用：免费版、标准版或是企业版。总有一些功能适合所有人使用，您只需要知道您想要的是什么，以及从哪里获取这些功能。

标准版数据库

标准版是微软 SQL Server 产品线的入门级数据库系统。这个版本支持最高 64 GB 内存和 16 个处理器内核。此外，它还支持最大 524 PB 的数据库。所有这些数字表明，标准版比免费版强大一个级别。

标准版还提供了一些免费版不具备的高可用特性，如日志传输、备份压缩、故障恢复集群和数据库镜像，尽管后两个特性的支持不如企业版稳定。然而，与免费版相比，标准版还提供了完整实现的复制特性，而前者只是作为一个订阅功能。

标准版还包含许多免费版没有的工具，如 Distributed Replay、SQL Profiler、SQL Server Agent 和 Microsoft System Center Operations，这些都是 SQL Server 日常维护和运营所需要的重要工具。在标准版中，您还可以使用其他一些特性，如策略自动化、多实例管理、自动索引视图管理、标准性能报表和性能数据收集。

有一个与数据库系统选择密切相关的方面是商业智能(BI)。虽然免费版能够支持小规模 BI 运营，但是您至少需要标准版才能创建真正的数据仓库、分析和数据提取、转换和加载操作。标准版包含 SQL Server Data Tools，它可用于开发与集成服务、报表服务和分析服务相关的项目，它们都包含在标准版中。您可以从集成服务运行时获得所有特性，包括多维数据模型和各种报表选项与配置。显然，如果您考虑实施 BI，那么标准版肯定好于免费版。

当然，所有这些附加特性都需要支付费用。但是，具体费用取决于您所需要的授权模型。标准版提供了 2 种授权：基于内核授权和服务器+客户端访问授权(CAL)。如果您选择基于内核的授权方式，那么您需要为每一个内核支持 1800 美元左右费用。如果有 16 个内核，那么费用就超过 28000 美元。如果您选择服务器+CAL 授权模式，那么您需要为每一个服务器支持 900 美元左右费用，为每一个 CAL 支持 200 美元左右费用。所以，具体价格取决于连接 SQL Server 的客户端计算机的数量。注意，这些价格并不包括渠道和容量授权。每一种计费方式都可能算出一笔不菲总价，与免费版不可同日而语。要仔细考虑标准版所提供的全部特性，确定您只

购买自己所需要的功能，而不要花冤枉钱。微软开发者网络有一个便利资源，它详细分析了每一个版本提供的特性。您能够从标准版获得比免费版多的功能，但是功能肯定不如企业版多。

免费版数据库

SQL Server 2012 的版本选择并非易事。你必须平衡现在与将来的产品价格与业务需求。如果过于节约成本，你就会牺牲一些重要功能。如果功能选择出错，你就会浪费金钱。

我们今天主要介绍 SQL Server 2012 免费版，它唯一的成本就是需要下载和安装数据库，然后你就可以分文不花地创建一个完整的数据库管理系统——包括许多有用的附加组件。

免费版包含一个完整的数据库引擎，你可以将它轻松部署到一台服务器，或者作为一个应用的内嵌数据库进行再发布。免费版支持的 T-SQL 语言元素与其他版本的 SQL Server 完全相同。你不仅可以在数据库上执行数据操作语言查询语句，也可以执行数据定义语言(DML)语句，用于创建一些数据库对象，如视图、触发器、游标和存储过程。

免费版不仅仅包含基本的 T-SQL，还支持全文搜索、原生 XML、快照级隔离和 SQL 通用语言运行时。而且，免费版内置了许多重要工具，如 SQL Server Management Studio、Configuration Manager、Advance Query Optimizer、Service Broker 和 Import/Export Wizard。

然而，你所能使用的工具取决于你下载的免费版版本。SQL Server 有 4 个免费版：

- **免费版**：不带任何工具的核心数据库。如果所有数据库连接都由远程应用发起，则可以选择这个版本。
- **LocalDB**：轻量的免费版数据库引擎，最适合嵌入到一个需要本地数据库系统的应用程序中。你还可以将这个版本与 Visual Studio 或其他开发工具绑定。LocalDB 运行在用户模式下，而不能作为一个服务运行，安装这个版本有一些先决条件，而且它支持 0 配置安装。
- **带工具的免费版**：你可以选择免费版或 LocalDB 数据库引擎，同时有一组工具可以帮助你安装和管理免费版数据库。
- **带高级服务的免费版**：最完整的 SQL Server 免费版。它包含所有的管理工具，以及报表服务和全文搜索。

注意，如果你准备下 SQL Server 免费版，那么你还有第 5 个选择：SQL Server Management Studio Express。这个只包含工具的软件包具有管理已安装 SQL Server 数据库(包括 SQL Azure)的功能。

正如你所看到的，免费版具有很多的功能。但是，你也一定要理解免费版不能做什么。首先是它的大小限制。免费版只支持最大 10 GB 的数据库，只能使用 1 GB 内存和最多 4 个处理器内核。所以，如果你考虑使用免费版，那么你应该注意它只支持较小规模部署。此外，免费版不包含其他版本 SQL Server 所具备的许多高级高可用性和可靠性功能。

你会发现，免费版还不支持其他 SQL Server 版本包含的许多特性。然而，使用免费版本，你仍然能够获得很多物有所值的功能，特别是对开发者和独立软件供应商(ISV)。如果你还没有使用过 SQL Server，只是希望了解它的工作方式，那么免费版绝对值得。最大的好处是，你可以从免费版升级到某个更高级的 SQL Server 版本，所以先从免费版开始，然后逐步升级，可确保万无一失。

[\(来源: TechTarget 中国 作者: Robert Sheldon\)](#)

SQL Server 2012 : 你应该掌握的 6 个管理工具

软件工具在企业数据库管理过程中扮演了至关重要的角色，因为良好整合的工具能够有效扩展 DBA 的能力，而那些随机组合的工具带来的只能是更多的困扰，甚至误操作。高昂的培训成本以及低下的生产效率也会严重影响 DBA 的正常工作。

为了满足企业数据库管理的新需求，微软公司发布的最新版 SQL Server 中包括了不少图形化的管理工具，可以帮助 DBA 与开发人员更高效地创建、管理和维护 SQL Server 解决方案，使得他们能够快速解决复杂的性能与配置问题。在本文中，笔者就将为大家介绍一些 SQL Server 2012 管理工具以及它们的主要特性。

SQL Server Management Studio

我们通常把 SQL Server Management Studio 简称为 SSMS，它是一个高度集成的管理开发环境，能够应付大多数的管理任务，并在单独的 SSMS 控制台中支持注册多个 SQL Server，从而在同一 IT 部门中管理多个 SQL Server 实例。举个例子，你可以使用 SSMS 管理 SQL Server 服务，如数据库引擎、集成服务 (SSIS)、报表服务 (SSRS) 以及分析服务 (SSAS) 等，同时还可以管理在多个服务器上的 SQL Server 数据库。SSMS 自带一些向导，可以帮助 DBA 和开发人员熟悉各种管理任务的操作，如 DDL 和 DML 操作、安全服务器配置管理、备份和维护等。SSMS 还提供了丰富的编辑环境，DBA 能够编写 Transact-SQL、MDX、DMX 和 XML/A 等脚本。此外，你还可以根据具体的动作来生成脚本。

SSMS 还有 Template Explorer，它提供了一个丰富的模板集，DBA 可以根据它来创建自定义的模板。SSMS 还支持 sqlcmd 脚本、浏览 XML 结果，还可以在不请求 SQL Server 连接的前提下编写脚本或查询。2008 版之后的 SSMS 会包括 T-SQL 的调试器、IntelliSense 智能提示和集成的源码控制。

SSMS 还提供了 SQL Server Surface Area Configuration 和 Activity Monitor 的访问功能。你可以使用 SQL Server Surface Area Configuration 来启动了停止 SQL Server 数据库引擎功能，使用 Activity Monitor 查看当前进程的信息，找到正在使用哪些 SQL Server 资源。

SSMS 主要的视窗包括：Object Explorer、Object Explorer Details、Object Search、Solution Explorer 和 Database Engine Query。

SQL Server Configuration Manager

[SQL Server Configuration Manager](#) 是用来管理 SQL Server 相关服务的一款工具，包括配置和管理已安装客户端以及服务器通信协议。你可以使用 SQL Server Configuration Manager 来修改 dump 目录（当错误发生时，SQL Server 创建内存 dump 的位置）、SQL Server 初始参数、主数据库文件以及 ErrorLog 位置。SQL Server Configuration 是一个微软管理控制台（MMC）的嵌入式管理单元，在开始菜单中就可以启动。举个例子，你可以根据以下步骤来启动 SQL Server 2012 Configuration：点击“开始”，选择“所有程序”，找到“Microsoft SQL Server 2012”，然后在“配置工具”中选择“SQL Server Configuration Manager”。

此外，你还可以从任何其他的 MMC 显示中启动它。mmc.exe 使用 Windows System32 文件夹下的 SQLServerManager11.msc 文件来启动 SQL Server Configuration Manager。

SQL Server Profiler

微软 [SQL Server Profiler](#) 是一个图形化的用户界面，能够根据所选的事件来捕获 SQL Server 或分析服务的动作。SQL Server Profiler 将不活动事件存为跟踪数据，它可以另存到一个本地文件或者网络文件，还可以存在一个 SQL Server 表中。SQL Server Profiler 包括一系列预先定义的模板，可以满足大多数捕获场景的需求。你可以使用被 SQL Server Profiler 捕获的数据来进行测试和诊断，举例来说，你可以重放或者测试该跟踪文件来进行问题诊断，或者同 Windows 性能日志文件进行比对，来找到资源使用峰值时的数据库事件。你还可以为数据库表创建一个审计跟踪。

Database Engine Tuning Advisor Wizard

[Database Engine Tuning Advisor](#)（简称 SQL Server DTA）是一个实用的数据库管理工具，通过它 DBA 可以找到最佳的数据库索引，索引视图和分区，而不需要对数据库内部结构有太多深入了解。你可以使用 SSMS 的 Query Editor 创建 T-SQL 脚本作为工作负载，然后使用 SQL Server Profiler 的 Tuning Template 创建跟踪文件和表负载。再加载并对特定的跟踪文件进行分析，SQL Server DTA 能提供建议的索引创建和改进方法，以便提升查询性能。你还可以使用 SQL Server DTA 来评估单独的查询。你可以 SSMS 其中的一个查询窗口中启动 SQL Server DTA，从而找到即席查询造成的影响。

Sqlcmd 命令行工具

[sqlcmd](#) 是一个命令行工具，用来执行 Transact-SQL 语句、存储过程和脚本文件。Sqlcmd 工具会发布一个 ODBC 连接到数据库，来执行批量的 T-SQL。T-SQL 命令的结果会在命令提示窗口中显示。你可以使用 sqlcmd 工具指向一个脚本文件，其中包含多个 T-SQL 脚本或者语句。

SQL Server PowerShell

[SQL Server PowerShell](#) 提供了一个强大的脚本外壳，DBA 和开发人员可以将服务器管理以及部署进行自动化。PowerShell 语言比 T-SQL 支持更多复杂的逻辑，使得 DBA 能够创建更健壮的管理脚本。

(来源：[TechTarget 中国](#) 作者：[Basit Farooq](#))

如何制定完美的 SQL Server 备份策略

作为一个 DBA，最重要任务之一可能就是恢复一个应用数据库。当数据库由于某种原因遭到损坏时，就需要将数据恢复到最近一次的正常状态。为了做到这一点，DBA 需要制定一个备份策略，使之支持各种不同的恢复情况。因此，制定一个保证顺利恢复数据的备份流程应该是 DBA 的最高优先级任务。如果没有相应的规划，数据库恢复过程就无法成功完成，DBA 也可能因此丢掉工作。在本文中，我将介绍一些关于开发、创建和测试备份策略的概念。

● 理解备份需求

没有人知道数据库将发生什么情况。它可能由于代码错误或硬件错误而遭到破坏，也可能由于数据中心火灾、水灾或其他灾难事件而丢失数据。为了恢复数据或减少数据库损失，必须根据特定要求备份数据库。

在创建备份策略之前，需要确定备份需求。所谓备份需求，是指需要确定允许丢失多少数据，以及需要多长时间才能恢复所丢失或损坏的数据。允许丢失多少数据，这通常称为恢复点目标 (Recovery Point Objective, RPO)，而用于恢复丢失数据的时间长度通常称为恢复时间目标 (Recovery Time Objective, RTO)。

为了确定环境的 RPO 或 RTO，必须与数据拥有者或客户面谈。客户可以确定他们的数据重要性，以及是否能够在数据库损坏或数据丢失之后重新创建或重新输入数据。下面是一些确定需求的问题列表：

- 数据发生变化的频率有多快？
- 当数据损坏或丢失时，是否能够重新输入或重新创建数据？
- 如果能够重新输入或重新创建数据，时间控制在什么范围才合理？
- 当执行恢复操作时，能够容忍的系统停机时间有多长？
- 能够容忍的数据丢失有多少？

客户需求将指导 DBA 确定需要设计、创建和测试哪一种备份和恢复策略。

● 远程、本地和第二备份位置

备份需求很重要，但还需要考虑保存数据库备份的位置。需要保证备份文件的可靠保存，以备在需要时使用，同时要保证它们不会受到服务器故障或数据中心整体事故的影响。因此，DBA 需要考虑备份文件的本地存储位置，也可以考虑将备份文件存储在远程位置。

如果要存储在本地，则需要写入或存储备份文件，使得在发生数据库服务器故障或数据库损坏时能够随时使用。将备份文件直接保存在数据库服务器上，可以实现较快的备份写入速度，因为这个过程没有网络 I/O 操作。但是，如果将备份保存在数据库服务器上，那么当服务器出现故障时，就可能会丢失备份文件。另一种方法是将备份保存到网络磁盘上。这个就可以将备份文件与数据库服务器分离，但是这种方法并没有应对网络备份主机出现问题的第二位置。最好要将备份文件复制到其他备份位置，如磁带机或网络备份存储解决方案。采用这种方式之后，当主备份设备出现故障时，就仍然能够从第二个位置取回备份文件。DBA 要保证在备份文件创建之后，马上将它们复制到第二个位置。这样可以最大程度缩短只有一个备份副本的时间。

在本地保存备份文件，就可以在需要恢复数据时快速获得备份文件。但是，当本地设施出现重大事故时，如火灾、水灾或其他自然灾害，又会发生什么结果？如果本地备份发生了问题，那么除非预先将备份文件复制到远程位置，否则就无法恢复数据。由于是开发备份解决方案，所以必须制定一个计划，将备份文件复制到一个安全的远程存储位置。

● 备份副本保存多长时间？

当确定特定数据库的备份需求时，还需要确定保存这些备份文件的时间长度。保存备份文件的时间越长，就需要为这些备份文件准备更大的磁盘空间。在确定备份文件保存时间时，并没有固定的答案或时间规定。首先需要评估具体环境，然后再确定需要多少个副本，以及这些副本的保存时间。在确定需要保存多少副本时，应该考虑以下问题：

- 数据库更新频率是多少？
- 发现数据问题需要多长时间？
- 恢复备份文件要多快才合理？

当确定保存的备份数量时，需要平衡恢复速度与所使用的空间大小。一旦确定了备份保存时间及所需要的副本数量，一定要让数据所有者明确计划保存的副本数量及它们恢复回数据库的速度。一定避免这种情况：数据所有者要求将数据库恢复到 30 天以前，而你只保存了 14 天的备份。

● 备份类型

数据库备份类型有很多种。下表列出了不同的备份类型及其用法的简单描述：

备份类型	描述
COPY_ONLY	不影响正常备份顺序的 FULL 或 TRANSACTION 备份
FULL	备份数据库的所有数据
DIFFERENTIAL	备份从上一次 FULL、PARTIAL 或 FILE 备份之后发生变化的所有数据
TRANSACTION LOG	备份事务日志信息（仅仅与处于 FULL 恢复模式的数据库相关）
FILE	备份一个或多个数据库文件或文件组
PARTIAL	备份所有读写文件组，选择性备份一个或多个只读文件

这里每一种类型都有细微区别，它们有不同的用途。最常用的三种备份类型是：完全备份（FULL）、增量备份（DIFFERENTIAL）和事务日志备份（TRANSACTION LOG）。

要确定哪一种备份类型组合最适合你的环境。下面是一些指导方法：

- 如果不担心数据库备份时间写入恢复数据，那么可以执行 FULL 数据库备份。如果这种类型符合要求，那么还应该将数据库设置为 SIMPLE 恢复模式，以控制事务日志的增长速度。
- 如果数据库很大，需要较长时间才能完成备份，备份文件又非常大，或者数据库中并非有很多数据库会发生变化，那么可能更适合组合使用 FULL 或连续 DIFFERENTIAL 备份方式。
- 如果数据经常变化，而且希望能够及时恢复，那么要考虑组合使用 FULL 和 TRANSACTION LOG 备份，或者根据数据库大小使用 DIFFERENTIAL 备份。
- 如果知道只有特定的文件或文件组发生更新，则适合使用 FILE 和 PARTIAL 备份方式。
- 如果想要采用特殊备份方法，例如将数据库移到测试服务器进行测试，但是又不想破坏正常的生产备份过程，那么可以考虑使用 COPY_ONLY 备份方法。

当数据库环境中实现特定的数据库备份策略时，一定要完全理解 RTO 和 RPO 需求，以及每一种不同的备份类型与存储位置如何帮助您实现 RTO 和 RPO 时间要求。

- **建立备份解决方案**

一旦确定了备份需求，以及用于保存备份的本地与远程位置，就可以考虑如何执行备份了。大部分人都会使用三种方法创建数据库备份：购买客户自定义（COTS）解决方案，使用 SQL Server 内置的维护计划方法，或者自行创建备份解决方案。每一种方法都有其自身的优点。

COTS 解决方案一般会包含许多附加条件，而且有时候会有比维护计划或自行开发解决方案更多的特性。维护计划解决方案非常容易实现，并且有足够的功能为大多数组织实现可靠的备份解决方案。如果有一些特殊的备份需求，那么实现需求的唯一方法就是自行开发一个备份方案。在自行开发时，要考虑一个问题：DBA 需要不停地维护这个解决方案，而 COTS 和维护计划解决方案则由供应商负责维护。

- **确认可以执行恢复操作**

原因很简单，即使备份了数据库，也不意味着就能够从备份恢复数据库。在创建备份之后，还应该确认这些备份是有效的。

有几种方法可以确认备份有效，其中一种是适时地从保存的备份执行一次真正的数据库恢复操作。这种方法可能不适用于生产环境，但是至少应该定期进行测试，保证可以在非生产环境恢复所保存的备份。

有时候可以验证备份是否写入正确。这可以通过“BACKUP VERIFYONLY”命令实现。这个命令实际上会读取备份文件，然后验证备份的数据是否能够真正恢复。通过使用 SQL Server 的验证方法，能够检查备份是否写入正确。

恢复一个数据库备份仅仅意味着该备份是正确的，但是并不意味着所有备份是正确的。DBA 应该考虑定期执行裸机恢复。要尝试这种方法，需要找一台未使用的主机，然后依次恢复操作系统、SQL Server 软件和所有的数据库（包括系统数据库）。执行一次服务器完全恢复可以确认备份策略的有效性，并且也能够有效保证在真正发生灾难事故时能够执行完全恢复。

- **保证备份文件的安全**

在创建备份计划时，需要考虑备份文件的保存位置，以及这些位置的安全性。由于备份文件包含了企业数据，所以需要保证这些备份位置是足够安全的。保证备份位置的安全性，才能保证备份文件不会意外落入外人手中。

如果数据库包含敏感数据，如信用卡号或个人身份数据，那么需要注意这些数据库备份文件的访问权限。如果数据库备份文件没有加密，并且他们有 SysAdmin 权限，那么其他人很容易通过恢复数据库备份而获取数据。

如果有保密要求，或者数据库存储了敏感数据，那么应该保证只有授权查看、管理和使用数据库备份的人员才允许访问备份位置。此外，可以要求应用程序在数据库中加密保存数据，或者采用某种方式加密数据库备份文件。加密备份文件的一种方法是在包含未加密保密数据的数据库中打开透明数据库加密（Transparent Database Encryption, TDE）。如果使用 TDE 加密备份文件，就需要开发一种方法，管理和恢复用于加密数据库的证书。如果不能恢复证书，就无法恢复 TDE 加密数据库。

- **规模安全性**

在开发备份解决方案时，首先需要确定备份的 RPO 和 RTO。确定了这些时间线，就可以指导选择符合要求的备份类型，以及恢复备份的位置。

规模安全性是开发备份策略时需要考虑的问题。正如动物世界一样，只有大量同种动物聚集时，一些特定物种才会感到安全。如果备份数量多于需求，那么 DBA 也会有安全感。

只有可以恢复所创建的数据库备份时，备份解决方案才会最终生效。DBA 需要定期验证备份，而且应该考虑至少每一年左右执行一次全面服务器恢复测试。

(来源：TechTarget 中国 作者：张亮亮)

SQL Server 2014 概述

新版本的 [SQL Server 2014](#) 将在 2013 年末或 2014 年初发布。本系列文章简单介绍了 SQL Server 2014 中的一些新特性。第一篇主要关注于内存和云特性。第二篇主要关注于 SQL Server 2014 的可用性和可扩展性。

微软在 6 月发布了 SQL Server 2014 的第一个 CTP 预览，为开发者、管理员及其他数据库人员提供了许多新的增强产品特性。最终版本将包含数据库和商业智能工具等完整 SQL Server 套件。与许多新版本软件一样，SQL Server 2014 实现了很多改进，其中包括扩展了内存功能、优化了云整合和增强了可用性与扩展性。

内存技术改进

SQL Server 2014 中最吸引人关注的特性就是[内存在线事务处理\(OLTP\)](#)引擎，项目代号为“Hekaton”。内存 OLTP 整合到 SQL Server 的核心数据库管理组件中，它不需要特殊的硬件或软件，就能够无缝整合现有的事务过程。一旦将表声明为内存最优化，那么内存 OLTP 引擎就将在内存中管理表和保存数据。当它们需要其他表数据时，它们就可以使用查询访问数据。事实上，一个查询会同时引用内存优化表和常规表。

SQL Server 2014 还支持其他几种方式的内存优化表。例如，微软将增加一个诊断工具，帮助我们确定哪些数据库和表最适合在内存中运行。此外，只有 Hekaton 表引用的存储过程可以原生编译；也就是说，存储过程可以自动转换为 CPU 可以执行的原生代码，而不像常规存储过程那样需要进一步编译或解释。通常，原生代码运行速度更快，所需要的内存也比 SQL 等语言少。

SQL Server 2014 增强内存相关功能的另一个方面是允许将 SQL Server 内存缓冲池扩展到固态硬盘(SSD)或 SSD 阵列上。扩展缓冲池能够实现更快的分页速度，但是又降低了数据风险，因为只有整理过的页才会存储在 SSD 上。这一点对于支持繁重读负载的 OLTP 操作特别有好处。

在 SQL Server 2014 中，[列存储索引](#)功能也得到更新。列存储索引最初是在 SQL Server 2012 引入的，目的是支持高度聚合数据仓库查询。基于 xVelocity 存储技术，这些索引以列的格式存储数据，同时又利用 xVelocity 的内存管理功能和高级压缩算法。然而，SQL Server 2012 的列存储索引不能使用集群，也不能更新。SQL Server 2014 引入了另一种列存储索引，它既支持集群也支持更新。此外，它还支持更高效的数据压缩，允许将更多的数据保存到内存中，以减少昂贵的 I/O 操作。

云整合

微软一直将 SQL Server 2014 定位为混合云平台，这意味着 SQL Server 数据库现在更容易整合 [Windows Azure](#)。例如，从 SQL Server 2012 Cumulative Update 2 开始，您就能够将数据库备份到 Windows Azure BLOB 存储服务上。SQL Server 2014 引入了智能备份(Smart Backups)概念，其中 SQL Server 将自动决定要执行完全备份还是差异备份，以及何时执行备份。SQL Server 2014 还允许将本地数据库的数据和日志文件存储到 Azure 存储上。此外，SQL Server Management Studio 提供了一个部署向导，它可以帮助您轻松地将现有本地数据库迁移到 Azure 虚拟机上。

SQL Server 2014 还增加了一个功能，允许将 Azure 虚拟机作为一个 [AlwaysOn](#) 可用性组副本。可用性组(Availability Groups)特性最初在 SQL Server 2012 引入，提供了支持高可用性数据库的故障恢复服务。它由 1 个主副本和 1~4 个次副本(SQL Server 2014 增加到 8 个)构成。主副本可以运行一个或多个数据库；次副本则包含多个数据库副本。Windows Azure 基础架构服务现在支持在运行 SQL Server 的 Azure 虚拟机中使用可用性组。这意味着您用一个虚拟机作为次副本，然后支持自动故障恢复。

可用性

微软已经通过 SQL Server 2014 扩展了其数据库产品的可用性。正如上文介绍的，子副本支持的数量已经从 4 个增加到 8 个。此外，即使网络中断，或者底层 Windows Server 故障恢复集群服务出错，子副本也允许继续执行读负载，而这在 SQL Server 2012 中是不支持的。另外，SQL Server 2014 还交付了增强的诊断信息，简化了错误信息，并且更容易查找和关联到明细信息。

SQL Server 2014 还通过支持**集群共享卷(CSV)**而实现更大的可用性。故障恢复集群服务使用了 CSV，让集群中的节点通过一个共享磁盘执行读写操作。这样多个 SQL Server 集群节点可以同时连接同一个卷。如果 SQL Server 集群节点丢失了存储连接，那么它可以转而使用另一个节点连接。此外，如果您分割了数据表，那么 SQL Server 2014 允许在线重建分区索引。您也可以指定切换分区的锁优先级，这样可以降低 CPU 使用率，减少锁和内存过载。

除了实现更大的可用性，SQL Server 2014 还可以通过集成 Windows Server 2012 R2 实现更大的扩展性。在物理环境中，您可以扩展到 640 个逻辑处理器和 4TB 内存。在虚拟机中，虽然只能扩展到 64 个虚拟处理器和 1TB 内存，但是这已经是相当大于处理级别了。此外，您还可以使用网络虚拟化，它将抽象网络层，简化 SQL Server 从一个数据中心迁移到另一个数据中心的过程。然后，您可以使用存储虚拟化创建存储池和存储分层，对存储的数据划分优先级。

SQL Server 2014 : 适合所有人使用的特性

可以肯定的是,在实现内存改进、云整合、可用性和可扩展性之后,SQL Server 2014 取得了长足的进步。但是,这些改进并不是终点,在安全性方面还增加了一些特性。例如,SQL Server 2014 现在加入了创建角色和子角色的功能,从而实现更大的职责分离。只要正确设置,数据库管理员就不需要读取数据就能够管理数据,而审计人员也不需要管理服务器就能访问数据。此外,SQL Server 2014 现在支持 I/O 资源管理,更新了集群层面的 Sysprep,并且提高了总体查询性能。而且,除此之外还有许多其他的新功能和功能改进。

实际上,SQL Server 2014 似乎推出一些适合所有人使用的特性。但是要记住,我们仍然只有 CTP 和传闻的发布日期。在我们看到最终产品之前,我们都不确定微软是否能够交付他们所承诺的所有新特性。或者是否还有一些我们不知道的附加特性。此外,虽然微软似乎计划采用与 SQL Server 2012 相同的授权模型,关于授权细节仍然不明确。所以,SQL Server 2014 肯定还有更多的新闻,但是我们现在看到的是一个好的开始,特别是那些可以通过内存改进、增强云整合和可用性扩展性提升而受益的组织。

(来源: TechTarget 中国 作者: Robert Sheldon)

SQL Server 2012 安全性解析

[数据库备份](#)被盗了，病毒攻击了服务器，数据库有未经授权的更改——如果 SQL Server 安全保护设置宽松的话，这些事情都会发生。

数据库技术本质上就是许多信息系统的组件，这是由于它们用来存储大量敏感企业数据而决定的，比如客户信息和其它机密商业数据。正因如此，把像 [SQL Server](#) 这样的数据库称为任何组织最有价值的资产并不为过，这也正是数据库在内部和外部攻击面前必须确保安全的原因。

不幸的是，事实并非如此，许多组织仍然发现数据库安全是一个不太受人关注的话题。结果，他们没有全面的数据库安全策略来保护他们的数据库基础设施不受内部和外部攻击。许多数据库甚至不能满足常规法规需求，比如 Gramm-Leach-Bliley 法案(简称 GLBA)、欧洲数据保护联盟(简称 EUDPD)、健康保险流通与责任法案(简称 HIPAA)、支付卡行业数据安全标准(简称 PCI DSS)以及塞班斯法案(简称 SOX)。

公司缺乏 SQL Server 安全策略

XYZ 公司是一家大型制药公司，在全球范围内针对小型制药公司销售各种医药产品。因为 SQL Server 有良好的往绩记录，漏洞和弱点比较少，所以 XYZ 公司使用 SQL Server 作为主数据库技术，存储公司的保密数据，比如客户信息、卡信息和员工信息。这些数据是从原始 Excel 电子表格中移植过来的。近几年来，XYZ 公司已经在其所有业务领域获得了显著增长。然而，在过去一年里，该公司却面临巨大舆论压力——泄露客户数据，未经授权访问以及其它造成 SQL Server 基础设施不安全的原因。

下面是 XYZ 公司 SQL Server 基础设施安全漏洞列表：

- 未经授权的更改。

在过去的一年中，未经授权的数据库变更是其 SQL Server 基础设施故障的主要因素。这是因为 XYZ 公司目前没有变更管理流程。未经授权对数据库变更意味着更大的外部威胁，比如来自外部黑客或恶意代码的攻击。

对于 XYZ 公司 SQL Server 数据库代码部署，很需要一套文档化而且健全的变更管理流程，因为它可以确保授权变更应用到生产数据库并避免不必要的风险。

- 未授权用户。

在少数情况下，会有未经授权的用户连续访问 XYZ 公司的 SQL Server 数据库。这主要发生在 SQL Server 实例配置为混合授权模式的情况下，或者“BUILTIN\Administrators”是“系统管理员”角色也有可能发生。经调查，大部分 SQL 登录(包括用 sysadmin(sa)权限的登录)存在漏洞或者空密码，而且用户没有被强制要求定期修改密码。

微软 [SQL Server 安全](#) 最佳实践建议你在混合认证模式使用 Windows 认证，因为 SQL Server 存储的 SQL 登录密码是纯文本形式。如果混合模式是为了向后兼容的缘故，那么要确保给 sysadmin 和所有其它 SQL 登录账号使用复杂密码。这一点可以通过对 sysadmin 和所有其它 SQL 登录用户启用“强制密码过期”和“强制实施密码策略”选项实现。

SQL Server 安全最佳实践还推荐对这些账号重命名，因为 sysadmin 是超级用户，许多黑客都知道这个账号，修改用户名会使得黑客控制攻击更困难。最佳实践还建议不要使用 sysadmin 管理 SQL Server。同时，确保“BUILTIN\Administrators”不是 sysadmin 角色成员；而要授予 sysadmin 权限给本地管理员组中的任何账号。

- SQL 注入攻击。

SQL 注入是未经授权访问数据库的另一种方式。那是因为应用程序中的 SQL 代码允许直接更新表，而且大部分 SQL 代码参数没有做验证。推荐验证代码参数并用 SQL Server 存储过程执行所有数据库更新。

- 来宾用户。

使用与数据库内部用户没有关联的用户登录仍然有可能访问到敏感数据。这是因为这些数据库启用了来宾账户。所以我们要禁用来宾账号。这将确保公共服务器角色不能访问到 SQL Server 中的用户数据库，除非你明确地赋予了对数据库的访问权限。

- 计算机病毒攻击。

XYZ 公司的 SQL Server 在前几年已经被计算机病毒攻击了几次。大部分 SQL Server 安全补丁、修复包以及更新包还没有应用到这些受影响的 SQL Server 上。微软公司持续发布的安全更新和补丁应该升级上去，这样能保护避免前述问题。

- 使用惯用的 TCP/IP 端口。

人们已经注意到大部分服务器攻击情况，黑客使用的是惯用的 [TCP/IP 端口](#)，比如 1433 和 1432。如果条件允许的话，应该修改这些标准端口，可以确保 SQL Server 安全。

- 数据库备份被盗。

在过去几个月里，人们已经注意到由于网络安全性不足导致的数据库备份被盗情况。而且，数据已经被盗，从表复制到操作系统文件。XYZ 公司接到建议，加强数据库备份安全性，启用透明数据加密。另外，要确保 SQL Server 实例上的“xp_cmdshell”和“OPENROWSET”保持禁用，只有系统管理员能访问这些 SQL Server 功能特性。

- 缺乏监视。

目前，XYZ 公司 SQL Server 数据库的 SQL Server 用户活动并没有被监视起来。很明显，这就为黑客打开了大门，他们可以访问该公司服务器而没有任何后顾之忧。SQL Server 提供了各种功能，比如登录审计或者登录触发器，通用遵守准则，ddl 触发器，甚至还有专门跟踪 SQL Server 恶意活动的通知和审计。

数据库基础架构的安全对于任何组织来说都是及其重要的，这也正是微软公司近几年在 SQL Server 安全功能方面投入巨大的原因。微软公司为 SQL Server 2005 重新设计了数据库安全引擎，从那时起，SQL Server 安全在每次更新都会有所提升。

[SQL Server 2012](#) 继续延续了这个趋势，增加了许多安全功能和增强功能。这些增强功能不只可以帮助组织改善数据访问控制，而且可以实现最高级别的数据保护和遵守。另外，这些功能还使得 SQL Server 成为了从安全角度看不容辩驳的最强大的通用数据库平台，漏洞更少，维护系统需要的安全补丁也更少。

安全可管理性改善

SQL Server 2012 引入了两个比较小但是非常有价值的变更来改善安全可管理性。第一个特性是 Windows 组的默认模式，第二个特性是用户定义的服务器角色。

Windows 组默认模式

在 SQL Server 2012 引入之前 我们不可能为 Windows 组指定默认模式。这样一来 当用户通过 Windows 组成员获得访问时，SQL Server 会自动创建一个独立用户（与 admin 账户映射），并会在数据库中创建同名模式。由于这一安全可管理性问题，最后数据库中出现了数百个用户和模式，这就给管理工作带来了困难，对管理员也是一场噩梦。所以，SQL Server 社区通过微软公司连接站点发起请求要求修复这一安全问题。

幸运的是，SQL Server 2012 解决了这个安全问题，它允许我们为 Windows 组赋给默认模式，这就帮助组织简化了他们的数据库对象管理。

下面的 Transact SQL (T-SQL) 证明了 Windows 组默认对象赋值过程：

```
-- Creating Default Schema "ProdAdmins" for Windows Group "MyDomain\ProdDBAs"
CREATE SCHEMA [ProdAdmins] AUTHORIZATION [MyDomain\ProdDBAs]
GO
-- Set Default Schema for Windows Group "MyDomain\ProdDBAs"
ALTER USER [MyDomain\ProdDBAs] WITH DEFAULT_SCHEMA=[ProdAdmins]
GO
```

用户定义的服务器角色

用户定义的服务器角色与固定服务器角色类似，唯一差异在于它们是由 SQL Server 管理员创建和管理的。用户定义的角色允许管理员们创建和分配服务器范围权限给用户定义角色，然后给他们添加成员。下面的 T-SQL 返回这些服务器权限列表：

```
USE [master]
GO
SELECT * FROM sys.fn_built_in_permissions ( DEFAULT )
WHERE [class_desc] IN ( 'ENDPOINT'
, 'LOGIN'
, 'SERVER'
, 'AVAILABILITY GROUP'
, 'SERVER ROLE' )
ORDER BY [class_desc], [permission_name]
GO
```

用户定义角色的主要优点在于它们通过按照职责划分限制授权用户访问简化了 DBA 工作。

我们可以通过 T-SQL 或者通过 SQL Server 管理工具 (SSMS) 来创建和管理用户定义角色，请看下面演示。

使用 SSMS 创建用户定义角色

- 在对象浏览器中，展开 SQL Server 实例，然后展开“安全”文件夹。
- 右击“服务器角色”文件夹，然后点击菜单“新建服务器角色”，运行“新建服务器角色”向导。
- “常规”页，指定服务器角色的名称，所有者以及适当的安全对象。
- 点击“成员”打开成员页面，然后给你定义的服务器角色添加成员。
- 最后，点击“成员组”页面，在这里你可以把新建的服务器角色作为一个成员，添加到现存的服务器角色中。

使用 T-SQL 创建用户定义角色

我们可以使用“CREATE SERVER ROLE, ALTER SERVER ROLE 和 DROP SERVER ROLE ”Transact-SQL 语句来创建，修改和删除用户定义服务器角色。请看下面说明：

```
-- Creating user-defined roles
CREATE SERVER ROLE [JuniorDBA]
-- Granting server-wide permissions
GRANT CREATE ANY DATABASE TO [JuniorDBA]
-- Adding members to user-defined roles
ALTER SERVER ROLE [JuniorDBA]
ADD MEMBER [Domain\JuniorDBA_Group1]
ALTER SERVER ROLE [JuniorDBA]
ADD MEMBER [Domain\JuniorDBA_Group1]
-- Making user-defined role member of fixed server role
ALTER SERVER ROLE [processadmin]
ADD MEMBER [JuniorDBA]
-- Dropping user-defined roles
DROP SERVER ROLE [JuniorDBA]
```

SQL Server 审计增强功能

在 SQL Server 2008 和 SQL Server 2008 R2 中增加了服务器和数据库审计具体规范功能，这是 SQL Server 中最有用的功能特性，可以帮助组织满足各种法规遵从需求，但这些审计功能只有企业版才支持。

幸运的是 服务器级别的审计规范特性目前在 SQL Server 2012 的所有版本中都支持了。SQL Server 2012 的审计规范功能对于写审计日志的失败更富弹性，可以支持限制审计日志的数量，无需设置回滚日志。SQL Server 2012 审计规范特性还支持用户定义组，这意味着我们现在可以给审计日志中写审计事件，使用 “ sp_audit_write (Transact-SQL) ” 存储过程就能做到。最后，SQL Server 2012 支持过滤审计事件，包括新增审计组来监视其包含的数据库用户。

数据库认证功能增强

在 SQL Server 2012 之前，用户需要用 windows 或者 SQL 登录 SQL Server 数据库引擎进行数据库身份验证。很明显，当从一个 SQL Server 实例向另一个实例迁移时，这种依赖会引起认证问题，而数据库管理员不得不确保所有可以登录源 SQL Server 的用户也存在于目标 SQL Server。如果由于某种原因，在目标 SQL Server 上登录用户不存在，那么用户就不能通过数据库登录验证了。

必要工具

SQL Server 2012 通过引入内含数据库验证来解决这个问题，它支持用户认证到数据库，而无需登录 SQL Server。所有包含数据库认证的元数据被存储在数据库本身内部。内含数据库认证的另一个优势是它消除了孤立或未使用登录信息在 SQL Server 数据库引擎中存在的可能性。

SQL Server 安全工具可以帮助 DBA 在 SQL Server 上快速执行安全审计。本文将概括介绍一些可用于发现 SQL Server 实例安全漏洞的第三方工具和微软自带工具。

Idera SQL Compliance Manager

Idera SQL Compliance Manager 是一个 SQL Server 综合审计工具，它可以帮助您监控和审计 SQL Server 用户活动及数据修改，并适时发出警报。您可以详细监控执行修改的人、事、时间和方式。它可用于跟踪修改、监控和审计数据访问、模式更改和登录错误。有几种内置报表可以验证 SQL Server 审计日志。DBA 可以配置一些警报，在发生一些可疑活动时发出提醒。Idera SQL Compliance Manager 可以帮助您保证多个 SQL Server 符合规范和数据安全要求，如 Sarbanes-Oxley、GLBA、HIPAA、PCI DSS 和 Basel II。此外，需要指出的是，运行这个工具本身不会消耗太多资源。

ApexSQL Audit

ApexSQL Audit 是一个综合工具，能够跟踪 SQL 数据库的数据修改。它可以捕捉和存储一个中央数据库的数据修改，跟踪执行这些操作的人、事、时间和方式。这个工具支持遍布整个中央数据库的全面审计报告。此外，它还可以向用户通知事件发生的原因与时间，捕捉相关的信息，如相关的 SQL 或 Windows 主机与应用名称。ApexSQL Audit 还可以跟踪表结构变化和未授权或恶意修改。

Application Security 公司的 DbProtect 数据库安全工具

Application Security 公司的 DbProtect 数据库安全工具是一个全面的数据库安全与规范工具，它可用于监控数据库安全需求。这个工具中以发现数据库漏洞、确定用户权限、应用用户访问控制和用户权限、实现恰当的漏洞检测策略和向用户报告恶意活动。

Quest ChangeAuditor for SQL Server

Quest ChangeAuditor for SQL Server 能够实时跟踪、审计和报告环境的变化，并在需要时发出警报。这个工具能够跟踪所有用户与管理器的活动，包括他们报告了哪些操作、操作执行的时间及操作执行的工作站。这个工具会监控环境的重要变化，而且您可以配置实时警报，在环境发生重要修改时发出警报。由于这个工具有一个中央知识库，所以使用它可以更轻松地在同一位置监控多个服务器。此外，它还支持基于角色的访问，因此允许审计员运行搜索和生成报表，而不需要修改应用的任何配置。

Microsoft SQL Server 最佳实践分析工具

Microsoft SQL Server 最佳实践分析工具是一个免费诊断工具，管理员可以使用它发现 SQL Server 实例的漏洞。使用这个工具，数据库管理员就可以快速收集与 SQL Server 和 Windows 配置相关的信息。这个工具有一些内置规则，它们可以确定 SQL Server 配置，以及环境配置是否符合微软提出的最佳实践。这个工具会报告 SQL Server 实例中所有不符合最佳实践的配置。此外，这个工具还可以帮助 DBA 发现所运行 SQL Server 实例中潜在的问题。在这个工具中，我最喜欢的功能是它能够对所发现的潜在问题自动推荐解决方法，这样可以帮助我们快速修复问题。

(来源: TechTarget 中国 作者: Ashish Kumar Mehta)

SQL Server 升级三部曲

SQL Server 版本升级规划是每个数据库管理员都渴望承接的一项任务。然而，升级 SQL Server 并不是一件容易的差事，规划不当会增加升级失败的风险。

我们在设计 [SQL Server 升级路线](#) 时，有必要把任务分成三步：升级前、升级中、升级后。

升级三部曲之一：升级前的任务

首先，我们要分析现有 SQL Server 环境，确保我们可以把当前版本升级到目标版本。我们要检查目标版本的硬件和软件需求。例如，如果你想把 SQL Server 2000 升级到 [SQL Server 2012](#)，你要确保硬件支持运行 SQL Server 2012。

另外，我们要确定好升级策略。要为 SQL Server 升级路径考虑选择直接升级还是迁移升级。我更喜欢迁移升级的方式，因为这种方式风险更小一些，虽然也更困难些。如果我们选择直接升级，一旦我们升级之后就不可能回滚了。直接升级的关键优势在于你不必担心 SQL Server 安全性和其它配置，因为这些信息在升级前和升级后会同步保留。当然，直接升级的方式需要在执行升级之前对所有系统和用户数据库做一套妥善的备份。

采用迁移升级的方式，新版本 SQL Server 会独立于老版本同步安装在独立的实例上，可以在同一台服务器或者是不同的服务器。这一步相当于是新安装一套 SQL Server，然后把现有 SQL Server 实例做迁移。例如，如果你采用这种方式把现有 SQL Server 2008 升级到 SQL Server 2012，你必须单独安装 SQL Server 2012，同时 SQL Server 2008 实例作为一个独立实例也是存在的，可能在不同的服务器上。这种方法的优势在于，你可以对数据库功能执行相关测试。此外，如果你想回滚也比直接升级的方式简单，因为原始的服务器仍然是完整的。这种方法的缺点是，在新实例启用以后，你必须执行数据库迁移，手工修改 SQL Server 实例的配置和安全性。如果你是升级到新硬件环境，或者在做 SQL Server 实例整合，那么采取迁移升级的方案是比较理想的。

不管你选择了哪种 SQL Server 升级路径，你都必须确保你的数据库升级以后仍然正常工作。幸运的是，SQL Server 提供了几款工具，你可以用他们来避免常见的升级错误。例如：你可以利用 SQL Server 升级顾问工具帮助识别可能影响升级的关键问题，比如不推荐使用或不在支持的功能选项、破坏的变更和行为变更。你

可以从 SQL Server 安装启动界面选择安装升级顾问工具。系统会提示你接受授权协议并在安装之前提供注册信息。安装这款工具，你必须提供产品密钥。要了解更多信息，请参考[微软公司官网资源](#)。

你还可以使用另一款叫做 SQL Server 升级助手的工具，它可以帮助你测试和验证为前一个版本设计的数据库应用功能。这款工具利用负载测试现有数据库源代码中的数据库应用的功能。要了解更多信息，请参考微软官网“[SQL Server 2012 升级助手工具](#)”相关文档。

升级三部曲之二：升级过程中的任务

SQL Server 升级路径与安装过程类似。你可以同时做安装和升级服务。例如，你可以升级数据库引擎的现有实例，同时安装集成服务、分析服务和报表服务。

如果选择独立升级的方案，我们可以使用数据库向导从一个 SQL Server 版本到支持的更高版本迁移单个数据库和数据库对象。我们可以使用 SQL Server 管理器（这是主要的 SQL Server 管理和维护工具）连接到 SQL Server 数据库服务器。要运行 SQL Server 2012 管理器，可以点击开始菜单，然后选择所有程序，然后点击“**SQL Server 管理器**”。出现提示时，连接到你将要管理的 SQL Server 实例。你可以在 SQL Server 管理器中运行“**复制数据库向导**”。该向导会引导你完成复制过程，支持你选择原数据库，并在 SQL Server 2012 数据库服务器上选择目标数据库。或者，你可以备份所有数据库和脚本安全设置，然后使用这个功能配置新实例。

升级三部曲之三：升级后的任务

我们需要监控应用程序性能，审查 Windows 和 SQL Server 的错误日志来检查升级后是否有问题发生。此外，你应该重新构建所有用户数据库索引，对所有内部和外部表运行“**sp_updatestats**”来运行“**UPDATE STATISTICS**”。

除了这些任务，还要对所有用户数据库执行“**DBCC UPDATEUSAGE**”，因为运行它可以纠正目录视图的页和行计数。

(来源：TechTarget 中国 作者：Basit Farooq)

总结 SQL Server 虚拟化的几点注意事项

[服务器虚拟化](#)已经出现好多年了,技术已趋于成熟。几年前,DBA 还很抵触把 [SQL Server](#) 迁移到虚拟机,但是现在已经是再平常不过的了。然而,在决定虚拟化之前有一些重要的事情需要考虑。

[SQL Server 虚拟化](#)最流行的解决方案是 [VMware](#) 和微软公司的 [Hyper-V](#)。首先你应该决定采用哪一种方案。VMware 仍然是虚拟化领域的王者,有更成熟的产品和更多的功能。而 Hyper-V 追赶的速度也很快,它比 VMware 成本要低一些。这两种产品都提供了免费试用版,如果你正好刚开始使用 SQL Server 虚拟化,我推荐你两种产品都试试。

当确定了哪些功能适合自己的企业,并研究清楚它们如何运作之后,你就可以比较成本和性能,然后做决定。SQL Server 开发团队与 Windows 和 Hyper-V 团队都出自微软,这是他们最大的优势。如果你只是寻求针对 SQL Server 的虚拟化方案,那么 Hyper-V 在大部分情况下都是不错的选择。但是如果你的公司还要虚拟化其它服务器,比如 Web 服务器或者域控制器,而且其他 IT 人员更熟悉 VMware 的话,可能坚持用统一方案是你最好的选择。

接下来要考虑的是什么服务器最先做迁移。首选对象是非生产用的 SQL Server。许多公司对于这些环境有不同的称谓,但是通常会叫做“开发环境”,“QA 环境”,“阶段环境”,“测试环境”,“基准测试环境”以及“压力测试环境”。更常见的是,超高性能并不是最高的优先级,这就使得对它们做虚拟化更容易了些。此外,在这些环境中的服务器可能比较老旧,经常是淘汰下来的生产环境服务器,有很多使用寿命问题和配置不一致的问题。另外,它们缺乏足够空间来还原生产数据库。另一个很糟糕但又常见的现象是,DBA 们通常需要过于关注生产环境,而忽略非生产环境。这就可能导致在开发周期内由于硬件老化故障或其它问题丢失数据库,耗费更多开发时间。如果你是在给这些环境做虚拟化,你会在新硬件上运行,而且你可以使他们配置一致,使 DBA 们工作更容易。

那么生产服务器怎么样呢?答案很简单:视情况而定。根据我的个人经验和与其它 SQL Server DBA 们的讨论,目前的共识是转向虚拟化对于较小的或者中等负载的服务器不成问题。这一技术已经足够先进,虚拟机可以与 SQL Server 共享内存和磁盘 IO。你可以考虑给重负荷的服务器使用 VMware 或者 Hyper-V,但是在这种情况下,我会考虑先做一些基准测试或者压力测试,确保运行于虚拟层的负载不会使性能下降到不可接受的程度。你还可以使用虚拟化作为运行数据库应用的初始设置,但是这种方式是否可以成功并不十分确定。这

些年来，我经历过几种情况，我们做了最坏的打算，构建了超级强大的服务器并运行应用程序，事情并没有像我们想象的那样发生。

还有一个问题需要提醒，那就是 BI 服务器。这些服务器通常托管非常巨大的数据库，这些数据库需要非常快速的处理能力。当 OLAP 多维数据集构建起来的时候，SQL Server 可能需要扫描 TB 级的数据量。在这种情况下，需要那种“猛踩油门”类型的性能，所以虚拟化的总开销可能难以被接受。例如，在我的公司里，我们没有看到在虚拟化数据仓库上的良好性能表现，需要转换到原来的物理服务器上去。

高可用性在当今时代非常重要，这个领域虚拟化可以提供很大帮助。例如，VMware 有一种称为“vMotion”的技术，支持虚拟机在物理服务器之间移动。这就使得你可以保持一套虚拟机处于运行状态，同时你只需要把虚拟机转换到另一台物理服务器就可以执行硬件升级。或者，如果数据库应用比较繁忙的话，当前物理服务器不再够用，你可以在秒级时间内把虚拟机迁移到另一台物理服务器上去。最终，如果底层物理硬件出故障了，vMotion 将自动把对虚拟机的控制转移到另一台物理服务器上去。所以，如果你用 vMotion 在虚拟机上设置了 SQL Server，这会极大地提升可用性；如果底层物理硬件出故障了，vMotion 将迅速把虚拟机转移到另一台物理节点上，SQL Server 仍然处于运行状态。

随着更多的新特性增加到了虚拟化软件中，在 SQL Server 环境下考虑虚拟化实现的可能性也大大增加，而转向虚拟化的总开销也变得更低了。

(来源：TechTarget 中国 作者：Roman Rehak)

深入了解 SQL Server 审计工具

微软公司在发布 SQL Server 2008 的时候引入了 SQL Server 审计 (Audit)，这是一款功能全面的数据库审计工具，弥补了 SQL Server 早期版本中的许多审计功能限制。那时候 DBA 们必须依赖于 SQL Trace 和其它工具。但是 SQL Server 审计现在已经变成了 SQL Server 本身自带的功能，并集成到 SSMS 中，提供了简单的界面可以实现细粒度的审计，这样你就可以针对目标特定对象、行为和规则进行审查。

SQL Server 审计会对发生在数据库引擎中的事件跟踪并记录日志。它可以记录服务器级别或者单个数据库级别的事件，尽管后者可能只在 SQL Server 企业版和开发版才有。SQL Server 审计功能提供了对于设置、授权、存储和视图事件数据必要的审计工具，不但可以完全管理整个 SSMS，还可以管理 T-SQL 和 SMO。与 SQL Trace 不同，SQL Server 审计只关注为了保证确保全面审计的安全、性能和管理功能的审计，而 SQL Trace 更侧重性能监控。

SQL Server 审计组件

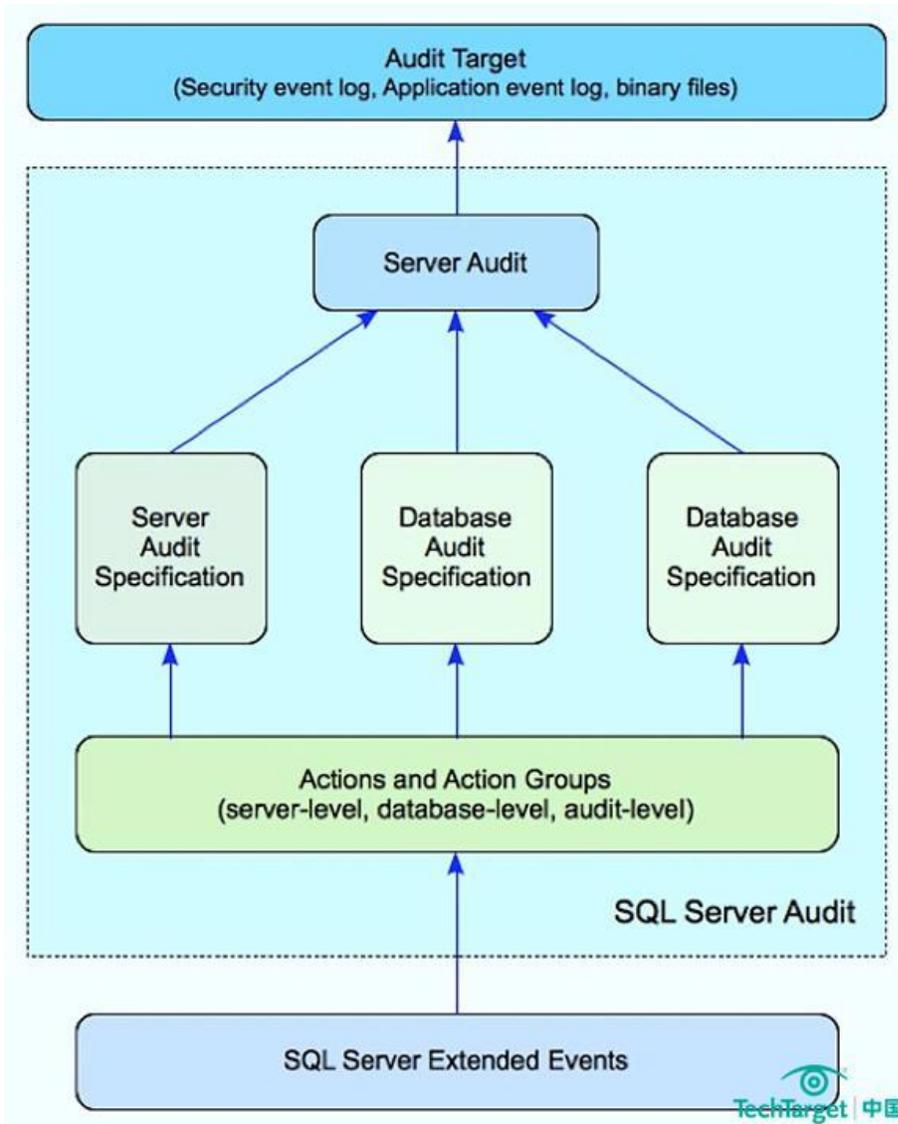


图 1 : SQL Server 审计功能包括审计对象和规范对象。

SQL Server 审计完成的大部分工作都发生在幕后。然而，为了执行审计，你应该熟悉这些必须配置并启用的对象，这些都是审计事件必须的。在图 1 的下半部分展示了参与审计过程的 SQL Server 审计工具。要注意的是，这些组件中有一些是专门针对 SQL Server 审计的，而另外一部分是 SQL Server 或者 Windows 操作系统环境的组件。

SQL Server 审计的核心是服务器审计，服务器级的对象可以作为整体配置设置审计，比如服务器应该如何响应审计失败的问题。这些设置中最重要的是定义目标，即存储事件数据的目标位置。这个问题上你有三种选择：Windows 安全事件日志，Windows 应用事件日志或者存储在指定位置的二进制文件。

SQL Server 文档中将服务器审计称为 SQL Server 审计或者简单称之为审计，这也是在 SSMS 的对象浏览器中提及服务器审计的方式。而这个提法更令人困惑的是，SQL Server 的文档也把已经配置的 SQL Server 审计对象集称为审计。例如，在上面的图中，包括在蓝色虚线边框中的那些组件都被认为是审计。在这里我们用这个提法代表了整个包。

配置完服务器审计，你可以添加服务器审核规范。规范要定义哪些服务器级别事件要包含到本审计中来。你可以对每个服务器审计只定义一条服务器审计规范，该规范必须指向服务器审计。此外，当你使用 SQL Server 审计工具配置规范时，你可以选择一个或者多个动作组，每个组都与一组事件集合有关联。例如，如果你想监视哪个负责人试图登录 SQL Server 实例而且登录失败了，那么你可以把“FAILED_LOGIN_GROUP”行为组添加到你的规范中。

除了服务器审计规范，你还可以添加一个或多个数据库审计规范。毫不奇怪，这是针对数据库的规范。你可以为每个服务器审计的每个数据库审计只创建一个数据库审计规范。例如，图 1 包含两项数据库审计规范，也就意味着两个数据库将被审计，每个数据库都关联了一个规范。

你可以选择一个或多个与数据库审计规范有关联的行为组，每个组都指向一组相关事件的集合。此外，数据库审计规范还支持你在具体数据库对象（比如表或者视图）或特定主题（比如 SQL Server 登录或数据库角色）上选择单个行为（比如更新或者删除）。例如，你可以监视系统管理员负责人什么时间运行了“DELETE”语句从“AdventureWorks2012”数据库的“Persons”表中删除过数据。

对于服务器和数据库审计规范，SQL Server 还支持“AUDIT_CHANGE_GROUP”，它支持你审计与审计流程本身有关的事件。然而，要注意的是，SQL Server 审计默认已经监视了审计有关的事件。

你可以注意到在上图 1 中还有 SQL Server 审计工具的另外一个组件，那就是 SQL Server 扩展事件（Extended Events）。这些是行为和行为组引发的实际事件。扩展事件针对 SQL Server 捕获的事件提供了内建的事件处理架构。这些事件出现在 SQL Server 项目的执行点。扩展事件使得为 SQL Server 审计和 SQL 跟踪这类工具捕获事件并应用成为了可能。

(来源：TechTarget 中国 作者：Robert Sheldon)

SQL Server Data Tools 2012 让数据库开发更容易

[SQL Server](#) 2012 中的一项新功能已经引起了很多反响——SQL Server Data Tools 2012 是对 Visual Studio 2010 中的一系列[数据库](#)和商务智能开发工具的整合。类似于之前 SQL Server 版本中的 Business Intelligence Development Studio (BIDS 商务智能开发平台)，SQL Server Data Tools (SSDT) 2012 提供了创建分析服务，报告服务和集成服务项目所必需的模板和组件。但 SSDT 2012 还添加了一些其他组件，它可以开发 SQL Server 数据库项目而且可以执行一般只在 SQL Server Management Studio (SQL Server 管理平台 SSMS) 中才有的任务，例如编辑数据，更新数据库对象，执行查询以及执行轻型管理。

SQL Server Data Tools 2012 中令人困惑的地方

尽管有着一个如此全面的工具集，SQL Server Data Tools 2012 还是经历了一个坎坷的开始，主要是由于对于 SSDT 究竟意味着什么，你向不同人提问，会得到不同的答案。已经安装了 SSDT 的数据库开发人员在 SQL Server 2012 的安装过程中肯定见过 BIDS 的替代品。另一方面，对于开发人员来说，当它们从网上下载（这是免费的）SSDT 时，会看到它是作为 Visual Studio 的插件来取代数据库开发工具 Data Dude 的。而导致困惑的地方部分来自于这样一个情况，即当 SSDT 首次安装时，它会包含 BI 开发工具或是数据库开发工具，但是从来不会两个都包含。

我们来看下 SSDT，它是 SQL Server 安装的一部分。当你首次打开[应用程序](#)时，你会看到一个与图 1 所示类似的界面。如果你已经用 SSDT 创建了开发项目，它们会列在 Recent Projects（最近项目）区域。你也

可能在以前就已经打开了一个或是多个窗口，所以它们也同样会被显示。即便如此，你的界面应该与图示中颇为相似。正如你所看到的，其他的都是“Getting Started (开始)”中的标准内容。

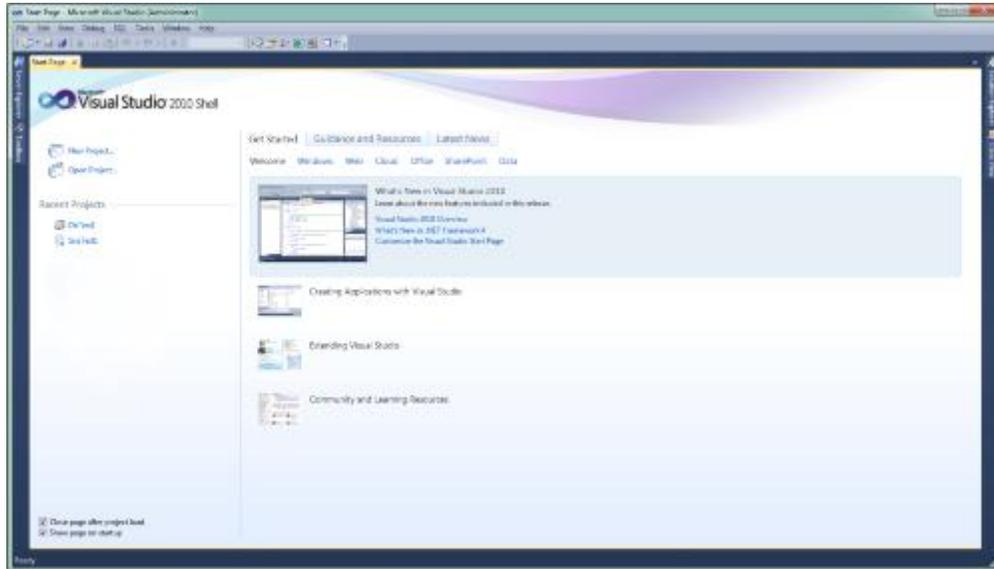


图 1. 首次启动 SQL Server Data Tools 2012 时的开始页面

当你创建一个新项目时，有趣的事儿就发生了。要创建新项目，就要点击 New Project (新项目) 链接。这会启动新项目对话框，如图 2 所示。注意到在 Installed Templates (已安装模板) 栏中，列出的第一类是 Business Intelligence (商务智能)。当这个被选中，如图示中一样，中间栏就会显示一系列 BI 开发可用的主要模板，例如 Integration Services Project (集成服务项目) 和 Analysis Services Tabular Project (分析服务列表项目)。

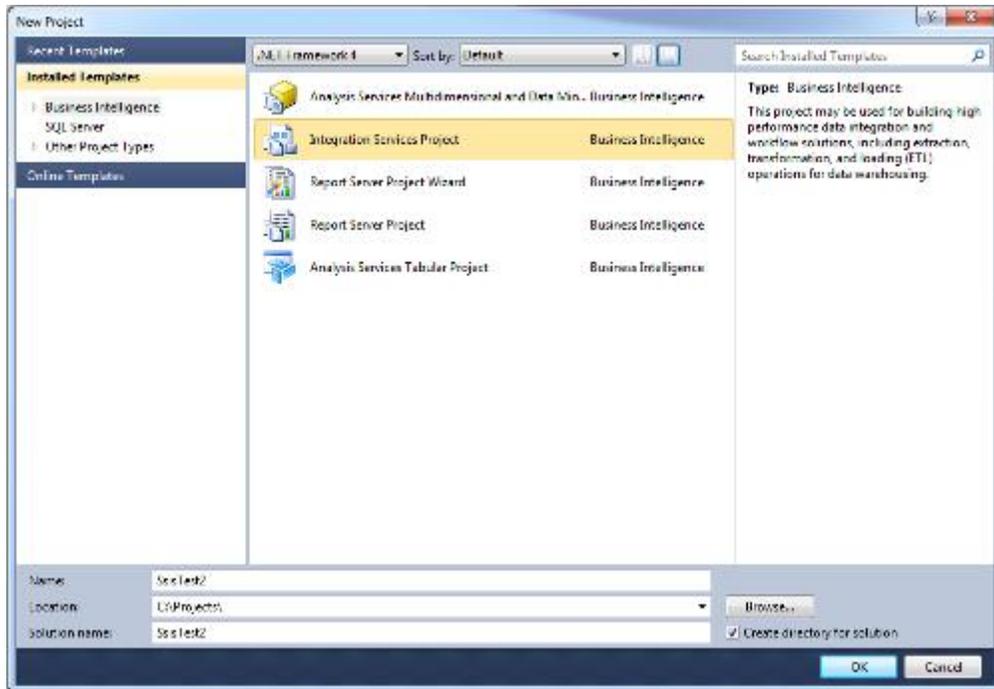


图 2. SQL Server Data Tools 2012 支持大量 BI 开发模板

如果你想扩展 Business Intelligence 结点,你会发现一系列 BI 项目类型:Analysis Services(分析服务), Integrations Services(集成服务)和 Reporting Services(报告服务)。对于每种类型,你会发现针对此类型可用的一系列完整模板,如图 3 所示。例如,图中选中了 Analysis Services 类型,并且正如所期待的那样,中间栏会显示特定于 Analysis Services 开发的一系列模板。

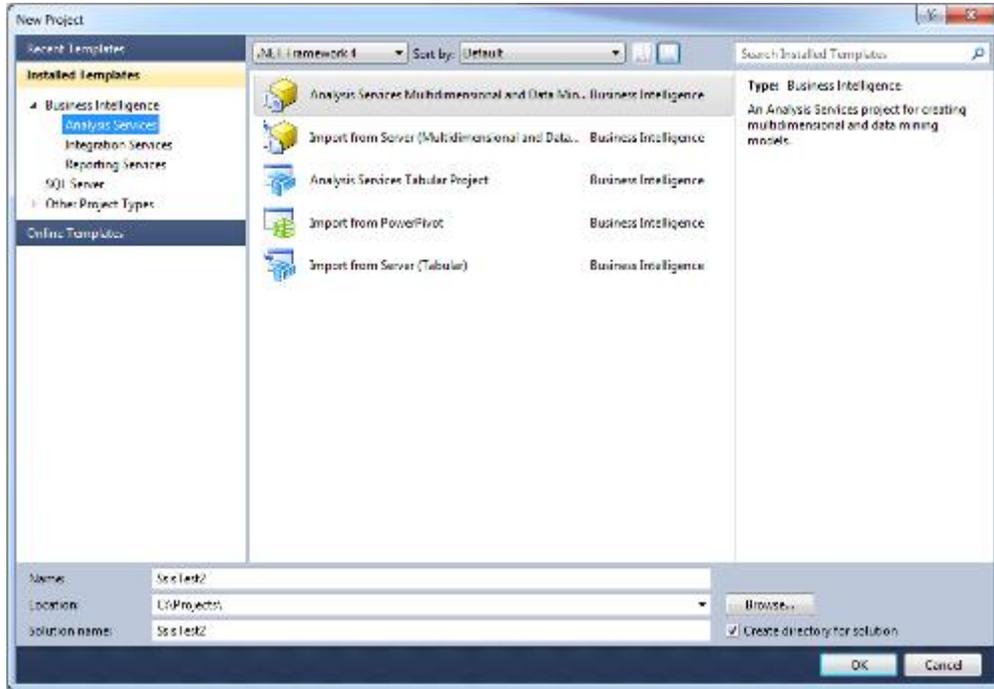


图 3. SSDT 包含针对 SSAS 开发的大量模板。

目前为止一切还好。但是要注意在 Installed Templates 栏列有另一个结点 – SQL Server – 你是不会在 BIDS 见到它的。如果你选择此结点,如图 4 中所示,你会在中间栏看到一个单独的条目:Microsoft SQL Server Data Tools (Web Install)。没错:数据库开发组件是不包含在内的。相反,你必须点击此选项,然后按照在线提示来安装这些工具。

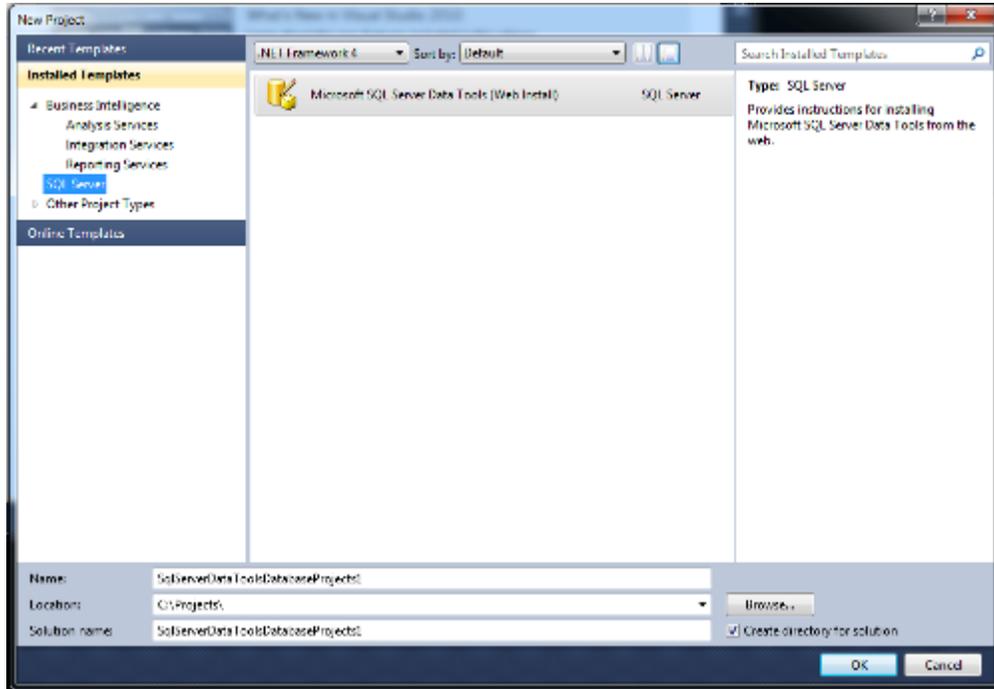


图 4. SQL Server 类别包含一个下载数据库工具的指示。

一旦你安装了数据库工具，你就必须重启 SSDT。当你这样做的时候，就会被提示选择你的主要开发环境，例如数据库开发，BI 开发或是一种编程语言。然而，一旦你到了这一步，就已经做好了开发一个数据库项目的准备。如图 5 所示，在工具安装好后，SQL Server 结点可用选项会得以更新，因此就不会再有 Web 安装的条目。你现在应该有了要在 SSDT 中进行 BI 和数据库开发所需要的一切。

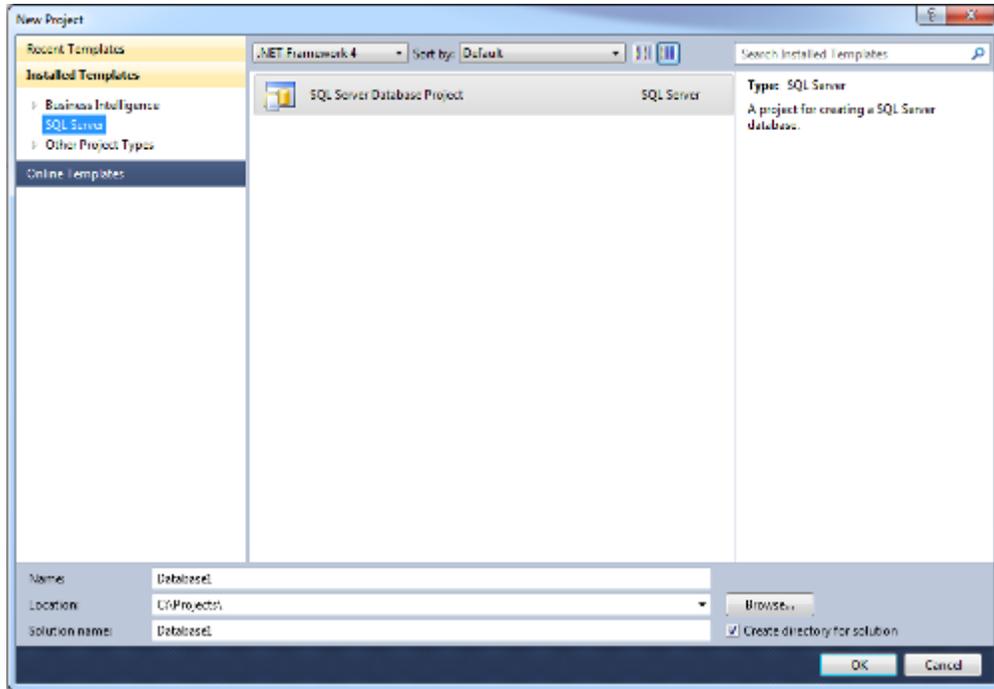


图 5. SQL Server Database Project 模板安装后的 SQL Server Data Tools 2012

当然,你在这里已看到的是一个以 SQL Server 为中心的视角。应用程序开发人员会有非常不一样的体验。

他们在大多数情况下会直接去 MSDN 网址检索数据库开发工具。如果他们已经在运行 Visual Studio, 这些工具就已经安装到了现有环境。如果不是这样的话, Visual Studio shell 和数据库工具就会得以安装。

如果你通过这种方式安装 SSDT, 然后想要安装 BI 开发工具的话, 你就必须返回到 SQL Server 2012 安装程序并安装 SSDT 功能。安装程序足够智能, 可以检测已经安装的 SSDT 数据库工具并只会添加 BI 工具。

SQL Server Data Tools 2012 [BI 工具](#)

SSDT 中的商务智能开发与你在 BIDS 中用过的是类似的。可见的差异表现在 SQL Server 2012 BI 栈添加了新功能, 例如 Analysis Services 表格项目, 或是更新已经调整过的界面组件以使开发更为容易, 例如 Integration Services 组件运行时的显示方式。图 6 展示了 SQL Server Integration Services (SQL Server 集

成服务 SSIS) 项目的控制流。请注意, 你从 SSIS Toolbox 窗口选择控制流组件, 在 Control Flow (控制流) 选项卡的设计面板使用这些组件, 并在 Solution Explorer (解决方案资源管理器) 窗口查看解决方案和项目的详细信息。

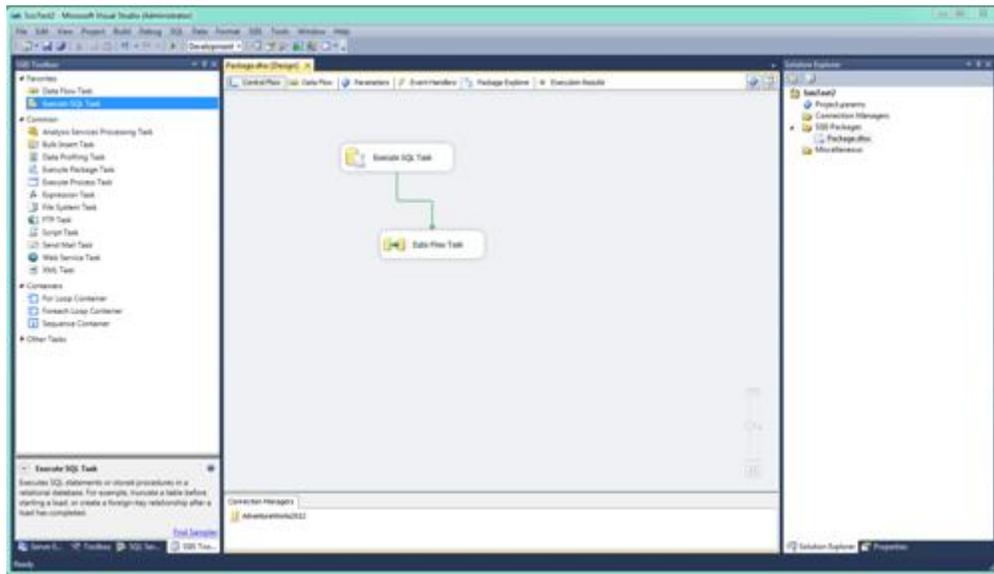


图 6. SQL Server Data Tools 2012 中的 BI 工具可以让你开发一个 SSIS 解决方案。

如果你熟悉 BIDS, 那么在适应 SSDT 的 BI 开发时应该不会有太大问题。从概念上讲, 几乎是没有什么变化的。如果你尚不熟悉 BIDS, 那么学习 SSDT 的过程将会与学习 BIDS 是一样的 – 在这种情况下你会想要通过 SQL Server Books Online (SQL Server 联机丛书) 和其他资源挖掘入门的详细信息。

SSDT 数据库工具

对于那些主要依赖于 SSMS 数据库开发的 SQL Server 开发者来说, SSDT 提供了一个非常不同的视角。SSDT 支持面向工程的离线开发, 这意味着你可以在不触及生产数据库的情况下创建, 删除和更新数据库工程直到你做好准备。尽管你仍可以直接对数据库做出变更, 但基于工程的离线开发会提供一个有效而安全的机制

用于数据库开发。并且你可以对从 2005 版本以来的所有 SQL Server 版本开发工程，还包括 Windows Azure SQL 数据库。

为了更好的了解 SSDT 工具中所蕴含的巨大威力，我们来看一些 SSDT 数据库开发功能。但是首先，要说明这些功能，我们需要在一个可用的 SQL Server 实例上创建一个测试数据库和表。对此，这些例子都是基于以下数据库和表定义的：

```
use master;

go

create database DbTest;

go

use DbTest;

go

create table TestTable

(

ColA int primary key,

ColB varchar (60) not null,

ColC datetime not null default getdate()

);

go
```

当你创建一个 SQL Server 工程的时候，SSDT 会显示一个与图 7 所示相似的窗口。在这种情况下，SQL Server Object Explorer (SQL Server 对象资源管理器) 和 Solution Explorer (解决方案资源管理器) 都是打开的，每个都会显示此新工程 (在此例中是 DbTest2)。在 SSMS 中 The SQL Server Object Explorer 窗口与 Object Explorer 窗口是相似的，会根据显示和层级顺序 (与一些服务器对象相同) 来提供对数据库和它们对象的访问。然而，此窗口还会显示你的数据库工程以及与此工程相关的其他工程。

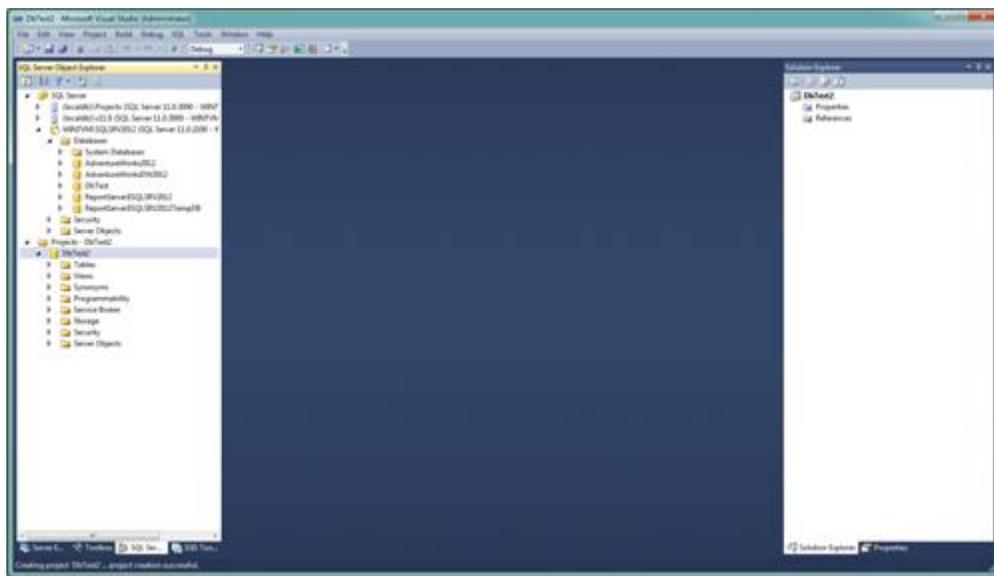


图 7. Solution Explorer 和 SQL Server Object Explorer 中列有数据库工程。

SSDT 中的 Solution Explorer 窗口是与 BIDS 中的 Solution Explorer 窗口和 SSDT 中的 BI 工具组件相似的。它提供了一个解决方案的层级视图，且其工程和工程文件每一个都是一个 SQL 脚本文件。并且由于 SQL Server Object Explorer 和 Solution Explorer 窗口显示一些相同的信息，所以你可以从任一窗口执行一些操作，例如访问表设计器工具。

假设现在我们想要修改 DbTest 模式，但不想涉及到在线数据库。则我们可以将此模式导入到我们的工程。

为此，右击 Solution Explorer 中的 DbTest2 结点，点击导入，点击数据库，然后根据提示导入数据库即可。

一旦你已经导入了此模式,SQL Server Object Explorer和Solution Explorer会在表的列表中显示TestTable,如图8所示。

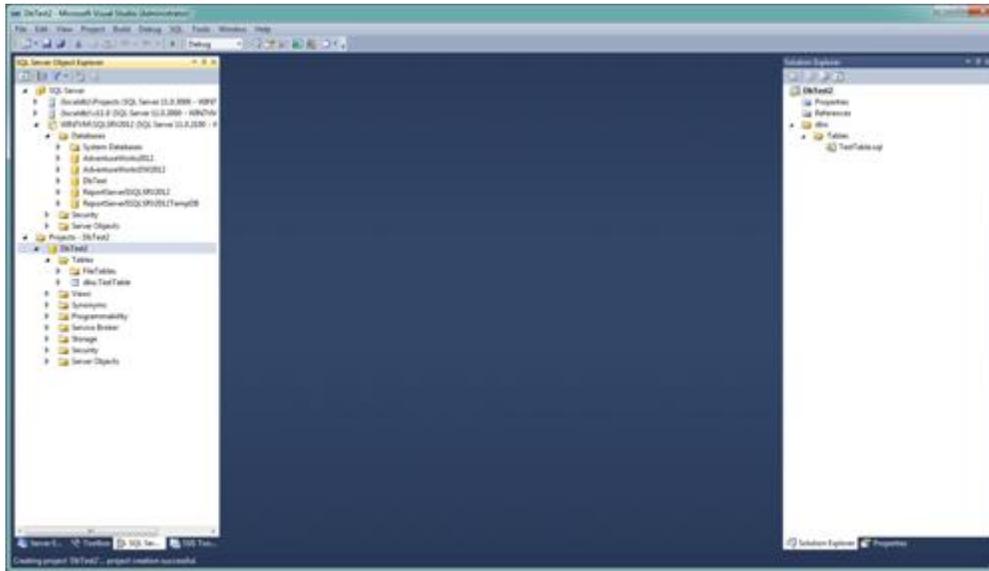


图 8. 你可以在 SQL Server Data Tools 2012 将一个现有模式导入到一个数据库工程

SSDT 还包含一个表设计器,可以让更新模式元素更加简单。右击 SQL Server Object Explorer 或 Solution Explorer 中的表,然后点击 View Designer (视图设计器)。这样会在 SSDT 中打开一个表可以让你简便地修改字段,键,约束以及触发器,如图9所示。在此选项卡上,你可以在网格内,属性窗口或底部的 T-SQL 面板做变更,而且这些变更会自动映射到其他部分。

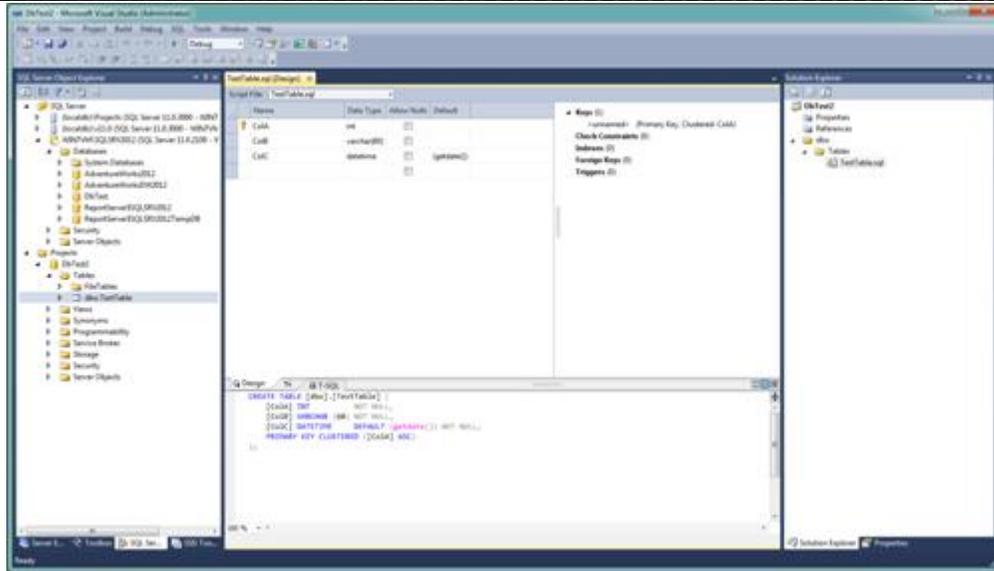


图 9. SQL Server Data Tools 2012 为更新数据库表提供了一个设计视图。

例如，我们给网络添加一个名为 CoID 的字段，给此字段分配 bit 数据类型并允许空值。正如你在图 10 中所见到的，当你做这些变更时，它们会自动映射到位于底部面板中的 T-SQL 代码里。

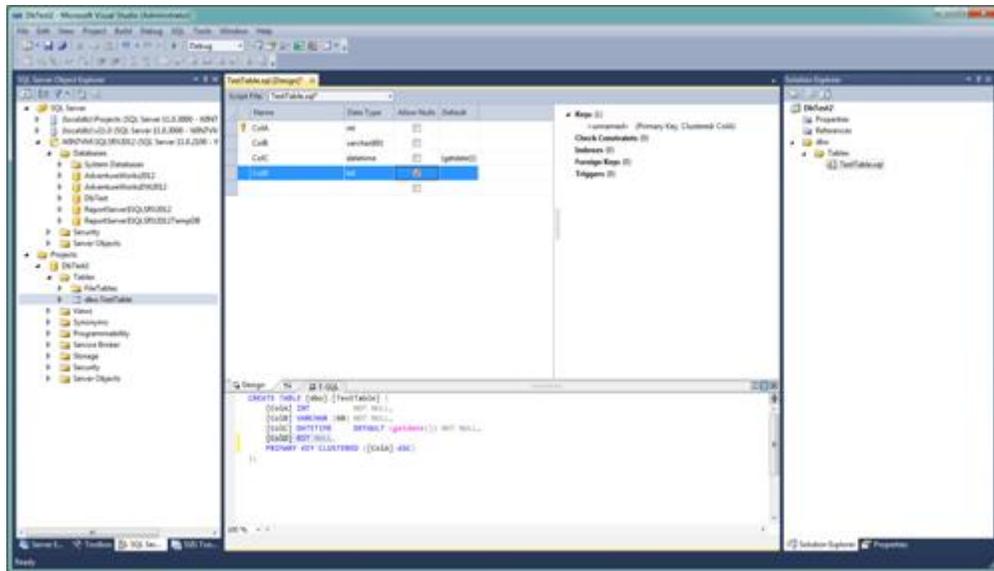


图 10. 表更新会立即出现在 T-SQL 代码里。

然后你可以比较工程中的数据库模式和目标数据库的模式。在 SQL Server Object Explorer 或 Solution Explorer 中 右键 DbTest2 点击 Schema Compare (模式比较) 然后选择目标数据库 (在本例中是 DbTest)，

当你点击比较按钮的时候，SSDT 会比较这两个模式并显示结果，如图 11 中所示。注意到我们在 DbTest2 中的 TestTable 所添加的字段会以黄色高亮显示，并且此字段在 DbTest 中会显示为丢失。

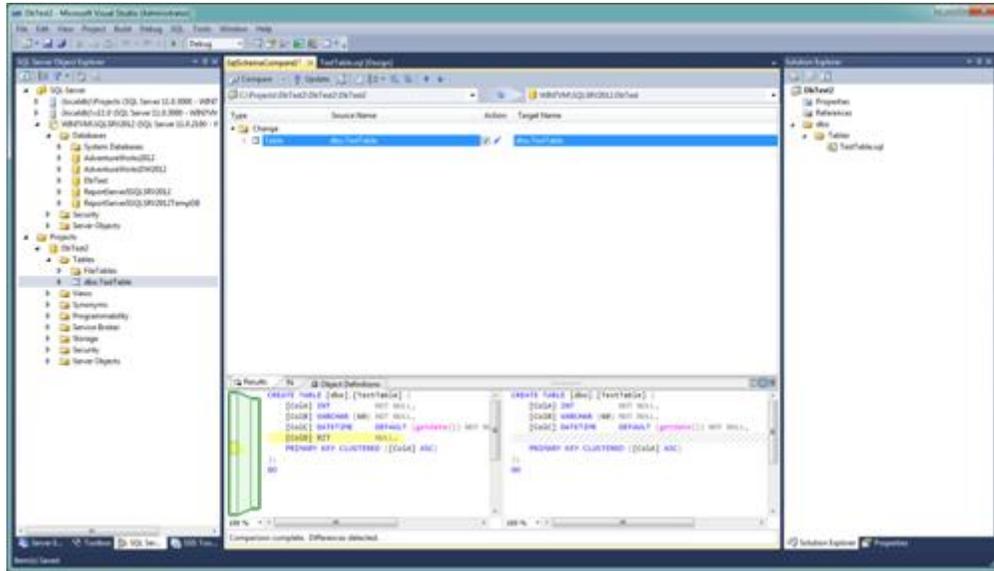


图 11. SQL Server Data Tools 2012 可以让你比较数据库模式

基于此对比，接着你可以立即更新目标数据库 (DbTest) 或生成一个脚本以供后面执行更新之用。另外一个选择是在你的目标 SQL Server 实例上创建 DBTest2。为此，右键 Solution Explorer 中的 DbTest2，点击发布，然后按提示操作。你就会有立即发布新数据库或生成一个脚本以供后面运行之用。一旦你已经发布了新数据库，它就会映射到 SQL Server Object Explorer，如图 12 所示。

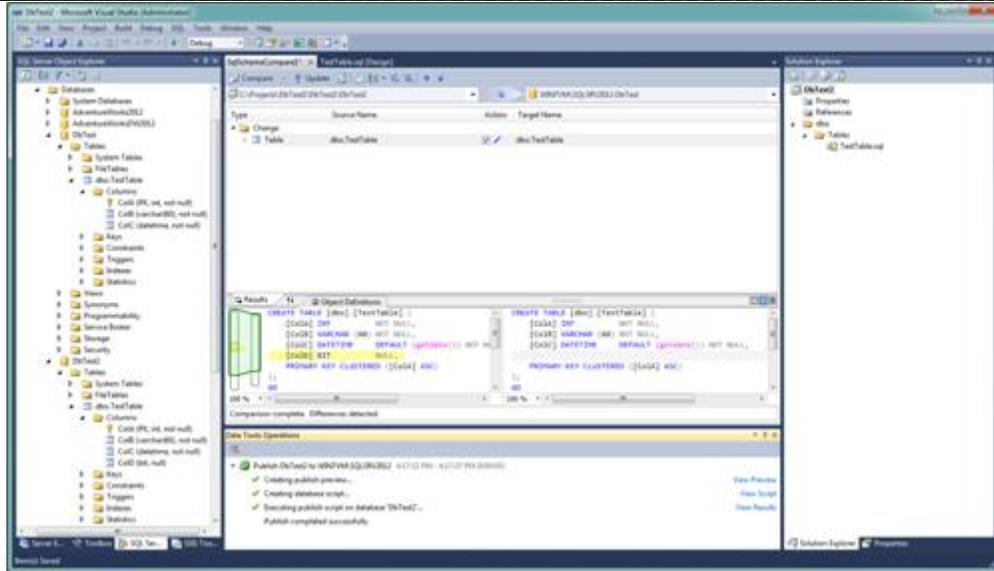


图 12. 你可以发布新数据库到一个 SQL Server 实例。

当然，这些例子只代表了 SQL Server Data Tools 2012 中数据库开发功能的一小部分。这些工具还支持智能提示，代码调试，搜索，以及大量其他用于开发和维护数据库的组件。添加这些到 BI 开发功能并且你会有一组工具可以满足你所有 SQL Server 数据库开发工作的需要。实际上，SQL Server 数据库和 BI 开发从来都没有像如今它们包含有 SSDT 这样简单。尝试使用这些工具对你来说有百利而无一害，特别是如果你已经准备好采用新方法来进行数据库开发。

(来源: TechTarget 中国 作者: Robert Sheldon)

如何创建 SQL Server 视图

在 [SQL Server](#) 数据库中你可以创建的对象之一就是视图，它是可以从一个或多个表（或其他视图）中 [检索数据](#) 的虚拟表，此过程类似于用一条 SELECT 语句返回一个数据集。SQL Server 视图会抽象出底层表模式却如同一个常规表一样以记录和字段的方式展示数据。因此，用户以及应用程序可以像查询一个表一样查询一个视图，但是此视图定义了什么样的数据是它们可以看到的和数据是如何被引用的。

你可以把 SQL Server [视图](#) 看做是可以简化和控制数据访问的过滤器。它们本质上是提供查询返回数据快照的命名查询。SQL Server 视图不仅可以让你关注并自定义如何展示数据，而且它们还可以通过抽象底层数据结构来提供一个额外安全层。你可以在不授权访问视图所基于的表的情况下授权访问一个视图。

SQL Server 视图还可以抽象对底层表的模式变更。例如，如果一个应用程序通过一个视图来检索数据并且此视图的结构（返回字段）保持不变，那么可以在不影响视图访问的情况下改变此视图的查询来适应底层表的变化。视图还可以用来提供向后兼容性来模拟一个显著变化或不再存在的表。

在本文中，我会为你介绍 SQL Server 视图并给出它们是如何工作的示例。这些示例基于在 SQL Server 2012 的一个本地实例上所创建的 AdventureWorks2012 数据库中的两个表。以下代码显示了我创建这些表的 T-SQL：

```
USE AdventureWorks2012;
GO

IF OBJECT_ID('CurrentEmployee', 'u') IS NOT NULL
DROP TABLE CurrentEmployee;
GO

SELECT BusinessEntityID, FirstName, LastName
INTO dbo.CurrentEmployee
FROM Person.Person
WHERE PersonType = 'em';
GO

IF OBJECT_ID('EmployeeInfo', 'u') IS NOT NULL
DROP TABLE EmployeeInfo;
GO

SELECT BusinessEntityID, JobTitle, HireDate
INTO dbo.EmployeeInfo
FROM [HumanResources].[Employee];
GO
```

正如你所看到的,我用了两个 SELECT...INTO 语句来定义并填充 CurrentEmployee 和 EmployeeInfo 表。如果你想要尝试此练习,你只需要执行此 T-SQL 就可以了。如果你正在执行一个 AdventureWorks 数据库不同的版本,你可能就必须调整代码和示例来让它们能在你的系统上工作。

CREATE VIEW 语法

在 SQL Server 中创建一个视图,要使用 CREATE VIEW 语句,如下面语法中所示:

```
CREATE VIEW [schema_name.]view_name
  [(column_name [, ...])]
  [WITH view_attribute [, ...]]
AS
  select_statement
  [WITH CHECK OPTION]

<view_attribute> ::=
  {ENCRYPTION | SCHEMABINDING | VIEW_METADATA}
```

你会以 CREATE VIEW 语句作为开始，提供名称以及可选性的提供你想要定义此视图的模式。如果你不指定模式名称，则会使用默认模式。

接下来，你可以指定一个或多个字段名，它们包含在圆括号中并以逗号进行分隔。用户会在查询此视图的时候看到这些字段。如果你不包含这些字段名，那么视图就会使用 SELECT 语句返回的名称。如果你想要使用不同的名称 – 或者你正在创建计算字段 – 你便可以将字段名包含在这里，即 CREATE VIEW 语句之后，或是在你查询的 SELECT 列表中指定字段别名。在这些字段名之后，你可以包含一个 WITH 语句，它可以接受一个或多个如下三项参数：

1. 加密: 加密 CREATE VIEW 语句文本并防止 SQL Server 从发布的视图进行复制。
2. 模式绑定: 把视图绑定到 SELECT 语句中所引用的表上，这样这些表就不能以任何会影响到视图的方式被修改。
3. 视图元数据: 返回视图的元数据信息以代替基表的相关信息。它得以应用的场景是当视图通过 DB-Library, ODBC 或是 OLE DB API 被查询以及当查询请求浏览模式的元数据的时候。

在可选的 WITH 语句之后，你可以指定 AS 关键字，紧跟着的是访问底层数据的查询。此查询是一条从一个或多个表或视图中检索数据的标准 SELECT 语句。你还可以通过使用 UNION 操作符连接 SELECT 语句来检索分区数据。但是，你不能包含 ORDER BY 语句，除非 SELECT 列表包含一个 TOP 语句。此外，你不能使用 INTO 关键字或 OPTION 语句，并且你不能引用一个临时表或表变量。

CREATE VIEW 定义的最后一个可选元素是 WITH CHECK OPTION，它可以强制执行所有视图的数据修改语句以与视图的 SELECT 语句相符合。换句话说就是，如果会影响视图以某种方式返回数据，即视图不再返回数据，那么 WITH CHECK OPTION 可以避免数据被更新。要更好的理解此项，同样还有其他与视图定义相关的项目，请参阅 SQL Server 联机丛书中的文章 [《创建视图 \(Transact-SQL\)》](#)。

在 SQL Server 中创建视图

一旦你对 CREATE VIEW 语法有了一个基本了解，那么你就做好了建立你自己视图的准备。以下示例包含了一个 CREATE VIEW 语句来定义 vEmployees 视图：

```
IF OBJECT_ID('vEmployees', 'v') IS NOT NULL
DROP TABLE vEmployees;
GO

CREATE VIEW dbo.vEmployees
(EmployeeID, FirstName, LastName, JobTitle, DateHired)
WITH SCHEMABINDING
AS
SELECT ce.BusinessEntityID, ce.FirstName, ce.LastName,
       ei.JobTitle, ei.HireDate
FROM   dbo.CurrentEmployee ce
       INNER JOIN dbo.EmployeeInfo ei
       ON ce.BusinessEntityID = ei.BusinessEntityID;
GO
```

首先,我们使用一个 IF 语句来检查视图是否存在然后在它存在的情况下进行删除。接着我们执行 CREATE VIEW 语句。我们以 CREATE VIEW 语句作为开始然后指定返回数据要使用的字段名。再一次,你就不需要在这里指定字段名了;你可以替代使用那些由 SELECT 语句返回的内容。

在这些字段名之后,指定了 WITH SCHEMABINDING 语句以确保在会影响视图的情况下表不会被更新。

接下来是 AS 关键字,然后才是实际查询。

SELECT 语句连接 CurrentEmployee 和 EmployeeInfo 表,从 CurrentEmployee 表返回 BusinessEntityID,FirstName 和 LastName 字段,从 EmployeeInfo 表返回 JobTitle 和 HireDate 字段。注意视图定义会把字段名 BusinessEntityID 改为 EmployeeID,将 HireDate 改为 DateHired。

一旦你已经创建好了视图,你就可以像是在一个表中一样在一个查询中调用它。例如,以下 SELECT 语句从 vEmployees 视图检索几个数据字段:

```
SELECT EmployeeID,  
       (FirstName + ' ' + LastName) AS FullName,  
       JobTitle  
FROM vEmployees;
```

注意,我们是根据字段名在视图中的定义方式来对它们进行引用的。在本例中,我们使用的是 EmployeeID 而非 BusinessEntityID。我们还连接了 FirstName 和 LastName 两个字段并将结果命名为 FullName。我们甚至可以像对表一样来限制查询,如下例中所示:

```
SELECT EmployeeID,  
       (FirstName + ' ' + LastName) AS FullName,  
       JobTitle, DateHired  
FROM vEmployees  
WHERE YEAR(DateHired) > 2002;
```

这一次，我们添加了一个 WHERE 语句来指定年份必须大于 2002。然后，我们在视图中引用字段名（DateHired），而不是表的字段名

当然，你可以创建比我们之前两个例子更为复杂的查询，但我们在这里所展示的应该可以让你对视图是如何工作的有一个很好的理解。

只要你能满足一些条件就可以在 [SQL Server](#) 视图上创建索引。例如，你必须保证视图中所有的表上都正确配置了 SET 选项。另外，此视图的查询必须是确定的；而且当输入值重复的情况下，此查询必须返回相同的结果。打个比方，你不能在查询中使用 getdate（）函数，因为这样的话返回值总是不同。此外，你必须在视图定义中始终包含 WITH SCHEMABINDING 语句。

如果你想要在一个视图上创建索引，这些仅仅是一些需要考虑的注意事项。对于更详细的要求，请参考 TechNet 的文章 [《创建索引视图》](#)。

你在视图上创建的第一个索引必须是唯一聚簇索引。在添加了此索引之后，你可以创建非聚簇索引。不过要记住，如果数据更新频繁，那么你可能不想索引化一个视图，因为索引本身会不断更新，这就会影响其他操作的性能。但是如果源数据是相对稳定的，那么索引化视图可以显著提高查询性能。例如，假设在第一部分的例子中当我们执行了 SELECT 语句的时候我们便生成了一个执行计划。图 1 显示了此执行计划的外观。

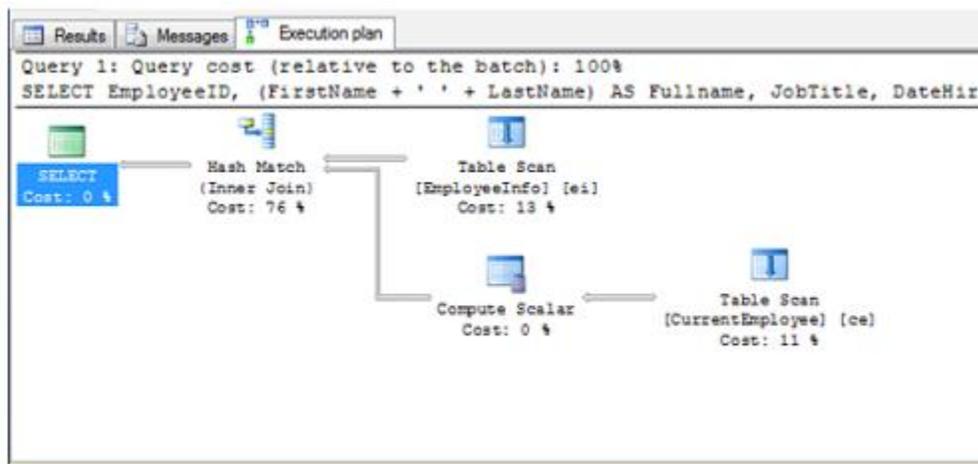


图 1：非索引视图在查询此视图时访问基础表

这是一个相当基本的查询。没有源表是索引化的，因此数据库引擎会执行表扫描来检索数据。但即使表被索引化了，数据库引擎必须查询两个表然后将数据连接在一起。现在，我们在表上创建一个索引。以下语句在视图的 EmployeeID 字段上定义了一个唯一聚簇索引：

```
CREATE UNIQUE CLUSTERED INDEX uq_vEmployees  
ON dbo.vEmployees (EmployeeID);  
GO
```

正如你所看到的，在一个视图上创建索引与在一个表上创建索引类似。如果在索引创建好后我们想要返回相同查询，我们就要有一个不同的执行计划，如图 2 中所示。


```
UPDATE vEmployees
SET JobTitle = 'VP Engineering'
WHERE FirstName = 'Terri'
AND LastName = 'Duffy';
```

正如你所看到的，我们在 UPDATE 语句中可以像引用一个表一样引用视图。接着我们使用 SET 语句来提供更新后的值，并用 WHERE 语句来指定要更新哪条记录。注意我们可以使用 CurrentEmployee 表中的 FirstName 和 LastName 值来更新 EmployeeInfo 表中的 JobTitle 字段。由于这两个表是通过视图连接的，那么在我们更新数据的时候就不用连接这两个表了。

我们还可以使用视图来插入数据。例如，以下的 INSERT INTO 语句使用 vEmployees 视图来添加了一个新雇员：

```
INSERT INTO vEmployees
(EmployeeID, FirstName, LastName)
VALUES(999, 'Terri', 'Duffy');
```

你可以从视图定义回想一下，此视图的 EmployeeID 字段是直接映射到 CurrentEmployee 表中的 BusinessEntityID 字段的，此表包含 FirstName 和 LastName 两个字段。我们可以执行此更新是因为所有三个字段是在同一个表中的。如果我们还要尝试添加一个职位和雇佣日期的话，那么我们的语句就会失败。

与 SQL Server 共事

视图为向用户和他们的应用程序公开 SQL Server 数据提供了一个有价值的工具。它们简化了底层数据结构并提供了一个额外安全层。它们可以索引化来改善性能并且甚至可以用来修改数据。另外，视图从 SQL

Server 的一个版本到下一个版本可以很容易地定义接口。实际上，自 SQL Server 2000 以来，基本的 CREATEVIEW 语法并没有改变。显然，你可以通过在你的 SQL Server 数据库中应用视图来获得很多东西。它们易于创建和使用。如果你能写一条 SELECT 语句，那么你就可以定义一个视图。

(来源：TechTarget 中国 作者：Robert Sheldon)