

第 1 章

1

SQL Server 2008概述和安装

欢迎阅读本书。如果你对SQL Server 2008尚不了解，但对使用Microsoft SQL Server 2008创建解决方案很有兴趣，本书就是为你编写的。你可能熟悉其他一些数据库（如MySQL、Oracle或者Microsoft Access），但相对于其他的数据库产品，SQL Server无论是在界面上还是在工作方式上都有所不同。本书旨在使读者快速达到游刃有余地使用SQL Server 2008进行开发的水平。本书专门针对SQL Server 2008的新手，以及即将开始使用SQL Server 2008的读者，同时，它还适用于那些曾经使用SQL Server 2005并想快速掌握学习SQL Server 2008的方法的读者。本书对于读者理解市面上其他数据库产品的基础知识也大有裨益，尤其是在使用T-SQL方面。由于许多数据库都使用基于ANSI标准的SQL，因此在学完本书之后，从SQL Server转向Oracle、Sybase等数据库也更为容易。

本章包含以下主题。

- 为什么选择SQL Server 2008？
 - 怎样知道硬件是否符合要求？
 - 是否可以确定拥有合适的操作系统？
 - 使用SQL Server 2008能够做什么？
- 接下来关注所选定版本的安装，并介绍以下内容。
- 在Windows XP平台上安装SQL Server 2008。
 - 默认没有安装的选项。
 - SQL Server实际上安装在何处？
 - 一台计算机上的多个安装。
 - SQL Server在机器上如何运行？
 - 如何实现安全性？
 - SQL Server的登录ID，特别是sa（系统管理员）登录。

1.1 为什么选择 SQL Server 2008

下面的论述是我个人的观点，无疑与其他人的观点会有所不同，但讨论的基础是没有问题的。SQL Server面对的数据库竞争产品，不仅有同样来自微软的其他数据库（如Microsoft Access和

2 第1章 SQL Server 2008 概述和安装

Microsoft Visual FoxPro)，而且还有来自竞争对手的数据库（如Oracle、Sybase、DB2和Informix等）。

大量的PC上都运行着Microsoft Access。多年来，该数据库一直是一些版本的Office软件包中的一个组件，这一事实使得Microsoft Access的身影随处可见。事实上许多人的确在使用这种数据库。遗憾的是，这种数据库在可扩缩性、速度以及灵活性上都有局限性。当然对于许多小型的内部系统而言，它们不需要太强的数据库功能，因而这些方面的不足并不算是问题。

那么，来看看与SQL Server真正旗鼓相当的对手：Oracle和Sybase。在数据库领域，Oracle常常被认为是市场领导者，有着非常庞大的用户基础。虽然在安装和管理上，Oracle比SQL Server复杂些，但不可否认，它是一个非常棒的产品，非常适宜为大公司提供大型解决方案。Oracle有许多部件，这使得该产品功能强大，在可扩缩性和性能方面不可小觑。Oracle还非常灵活，允许使用者按需要添加工具，在这一方面Oracle比SQL Server具有更强的适应性。例如，无论用户是否需要使用新的.NET特性，SQL Server 2008都要求用户必须在服务器上安装.NET Framework。然而，从开发者的角度来看，Oracle的使用在许多方面都不太方便，例如，专门的SQL Query工具、XML和Web技术工具，以及建立完整数据库解决方案的方式。Oracle的其他一些缺点还包括它较高的价格以及在安装和高效运行上的复杂性。尽管SQL Server也能同样高效地工作，但Web搜索引擎还是广泛地使用Oracle。具有新功能的SQL Server 2008将给Oracle带来压力，迫使其扩展现有的功能以迎接挑战。SQL Server采取的向来是一次性购买的解决方案，如果购买了正确版本的SQL Server，那么用于分析数据或从一个数据源（如Excel）向SQL Server中复制数据的所有工具都将包含于其中。而对于Oracle，必须额外购买所需要的每一个附加特性。

接着来看看Sybase。Sybase与SQL Server十分相似，它们之间的主要不同是：Sybase没有图形用户界面前端。主要用于小规模安装的Sybase iAnywhere是有前端的，但高端的Sybase没有。有些人可能认为图形用户界面前端首先是为那些不懂如何写代码的人准备的（当然，这是他们的一家之言），因而没有必要设计图形用户界面前端，但是既然指向、点击和拖动就可以搞定，为什么还要按下60多个键呢？

虽然Sybase有用于Windows的版本，但它主要还是用在UNIX上。尽管可以通过工具从Windows的机器连接到UNIX上的Sybase，但依然要完全用代码来建立数据库解决方案。Sybase非常快也十分稳健，一年通常只需重启一两次。关于Sybase的另一点是，它没有SQL Server那样丰富的命令和特性。SQL Server有更加强大的编程语言，功能也比Sybase强大。

每个数据库都有其特有的SQL语法，尽管它们的基本SQL语法是一致的，即公认的ANSI-92标准。这意味着用于检索数据等的语法在数据库之间是相同的，但每个数据库用于维护的语法是其自身特有的。若试图在一个数据库中使用源自另一个数据库的独特SQL语法，可能是无效的，也可能所起的作用大相径庭。

因此，SQL Server在数据库市场中似乎是最佳的选择，而且在很多情形下也确实如此。它可以小到只用于少量用户，也可以大到足以支持最大的企业。它没有Oracle或者Sybase那样高的价钱，但却能向上扩展并处理太字节（TB）的数据而无需太多的考虑。你将会发现，SQL Server易于安装，并且由于SQL Server的大多数功能在一个完整的包中，当需要剩余的功能时，执行简单的安装就够了。

既然知道了为何选择SQL Server, 接下来需要了解市面上有哪些可以购买的SQL Server版本, 每种版本针对的市场是什么, 什么版本对你最合适, 以及你的机器上能够运行什么版本。

1.2 SQL Server 的发展

SQL Server经历多年后发展到了今天的产品。表1-1概述了这一发展历程。

表1-1 SQL Server发展历程

年 份	版 本	说 明
1988	SQL Server	与Sybase共同开发的、运行于OS/2上的联合应用程序
1993	SQL Server 4.2 一种桌面数据库	一种功能较少的桌面数据库, 能够满足小部门数据存储和处理的需求。数据库与Windows集成, 界面易于使用并广受欢迎
1994		微软与Sybase终止合作关系
1995	SQL Server 6.05 一种小型商业数据库	对核心数据库引擎做了重大的改写。这是首次“意义非凡”的发布, 性能得以提升, 重要的特性得到增强。在性能和特性上, 尽管以后的版本还有很长的路要走, 但这一版本的SQL Server具备了处理小型电子商务和互联网应用程序的能力, 而在花费上却少于其他的同类产品
1996	SQL Server 6.5	SQL Server逐渐突显实力, 以至于Oracle推出了运行于NT平台上的7.1版本作为直接的竞争
1998	SQL Server 7.0 一种Web数据库	再一次对核心数据库引擎进行了重大改写。这是相当强大的、具有丰富特性的数据库产品的明确发布, 该数据库介于基本的桌面数据库(如Microsoft Access)与高端企业级数据库(如Oracle和DB2)之间(价格上亦如此), 为中小型企业提供了切实可行(并且还廉价)的可选方案。该版本易于使用, 并提供了对于其他竞争数据库来说需要额外附加的昂贵的重要商业工具(例如, 分析服务、数据转换服务), 因此获得了良好的声誉
2000	SQL Server 2000 一种企业级数据库	SQL Server在可伸缩性和可靠性上有了很大的改进, 成为企业级数据库市场中重要的一员(支持企业的联机操作, 其所支持的企业有NASDAQ、戴尔和巴诺等)。虽然SQL Server在价格上有很大的上涨(尽管算起来还只是Oracle售价的一半左右), 减缓了其最初被接纳的进度, 但它卓越的管理工具、开发工具和分析工具赢得了新的客户。2001年, 在Windows数据库市场(2001年价值25.5亿美元), Oracle(34%的市场份额)不敌SQL Server(40%的市场份额), 最终将其市场第一的位置让出。2002年, 差距继续拉大, SQL Server取得45%的市场份额, 而Oracle的市场份额下滑至27%(来源于2003年5月21日的Gartner Report)
2005	SQL Server 2005	对SQL Server的许多地方进行了改写, 例如, 通过名为集成服务(Integration Service)的工具来加载数据, 不过, SQL Server 2005最伟大的飞跃是引入了.NET Framework。引入.NET Framework将允许构建.NET SQL Server专有对象, 从而使SQL Server具有灵活的功能, 正如包含Java的Oracle所拥有的那样
2008	SQL Server 2008	SQL Server 2008以处理目前能够采用的许多种不同的数据形式为目的, 通过提供新的数据类型和使用语言集成查询(LINQ), 在SQL Server 2005的架构的基础之上打造出了SQL Server 2008。SQL Server 2008同样涉及处理像XML这样的数据、紧凑设备(compact device)以及位于多个不同地方的数据库安装。另外, 它提供了在一个框架中设置规则的能力, 以确保数据库和对象符合定义的标准, 并且, 当这些对象不符合该标准时, 还能够就此进行报告

1.3 硬件要求

在对SQL Server稍有了解之后,接下来要涉及的重要问题很可能是:“我所拥有的计算机是否足以运行所选择的SQL Server版本?这是否有助于我的决策?”

若根据现今的最低硬件规格标准来判断,哪怕是最低成本的方案,多数情况下对于大部分的SQL Server版本,其答案都是肯定的。然而,手边可能会有较早的硬件(事物发展得如此之迅速,即便是几个月前购买的硬件,也可能很快就被认为是低于当前最低硬件规格标准的),因此,需要了解一下最低硬件要求是什么,并以此检查所拥有的计算机,以确定是否具备满足需求的硬件资源。

1.3.1 CPU

对于运行SQL Server的CPU,建议的最低要求是32位版本对应1 GHz的处理器,64位版本对应1.6 GHz的处理器,或兼容的处理器,或具有类似处理能力的处理器,但推荐使用2 GHz的处理器。然而,像这里列出的大多数最低要求一样,微软事实上推荐的是更快的处理器。处理器越快,SQL Server运行得就越好,由此而产生的瓶颈也越少。现在的很多机器使用的都是2 GHz及以上的处理器。这将缩减开发所花费的时间。

然而,跟提升SQL Server的运行速度相关的硬件并非只有处理器,SQL Server的速度很大程度上也受当前计算机中内存空间的影响。

1.3.2 内存

确认系统的处理器速度足以满足需求后,接着要检查系统中是否有足够的内存。SQL Server需要的RAM至少为512 MB。不应该打开和运行太多的应用程序,因为那样很容易让SQL Server得不到足够的内存,从而使其运行变慢。微软推荐1 GB或者更大的内存,当真正开始使用SQL Server时,实际上内存大小至少应该是推荐大小的两倍。

如果要运行企业版,特别是若想要使用更高级的特性时,则至少(最低限度)需要有1 GB的内存。

内存越大越好。对于任何开发人员所使用的计算机,强烈建议至少要有1 GB的内存,若有2 GB则比较理想,这样才能够获得良好而全面的性能。如果内存足够大,那么进程就可以保持在内存中,而不是在要运行另一个进程时,将进程交换到硬盘上或别的区域中,因而当要从进程停止的地方继续运行时,则不必等待SQL Server被重新载入回内存。上面的情形称为交换(swapping),内存越大,可能发生的交换就越少。

将CPU速度和内存作为一个整体来考虑,对于计算机运行速度至关重要的正是这两方面因素。当运行速度足够快时,开发的速度也会尽可能地快。

在SQL Server的安装过程中,内存不足不会导致安装停止,但会发出警告,以告知你需要更多的内存。

1.3.3 硬盘空间

SQL Server需要比较大的硬盘空间。这不足为奇，如今主要的应用程序都是需要大量的硬盘空间的。不考虑要添加的数据文件，SQL Server自身将占用1 GB以上的硬盘空间。当然，本章后面要用到的安装选项将决定总共所需的硬盘空间。通过选择不安装某个可选部件，可以减少对硬盘空间的需求，例如，选择不安装联机丛书。不过，如今大多数的笔记本电脑都至少配有40 GB的硬盘，配备80 GB的硬盘也是很平常的事。硬盘空间是廉价的，因此，最好是购买容量远远超出当前所需容量的硬盘，而不要采用恰好满足眼下空间大小要求的硬盘，使得将来可能不得不另行购买硬盘以满足增长的要求，这样将随之带来移动资料、整理原先硬盘上的空间等问题。

此外，还需要在硬盘上留有备用的空间，以满足SQL Server和数据库的扩展。另外，还需要为开发过程中要用到的临时文件准备硬盘空间。总之，考虑大的硬盘空间吧——多多益善！

1.3.4 操作系统要求

SQL Server 2008可以运行在Windows Vista Home Basic及更高版本上，也可以在Windows XP上运行。从服务器端来看，它可以运行在Windows Server 2003 SP2及Windows Server 2008上。它也可以运行在Windows XP Professional的64位操作系统上以及Windows Server 2003和Windows Server 2008的64位版本上。因此，可以运行SQL Server的操作系统是很多的。

1.4 示例

为了充分展示SQL Server 2008，我们将一起为一家金融公司开发一个系统，该系统具有存款、股票购买功能。该应用程序适合大型机构，或者经过稍加修改也能够供个人记录银行业务使用。

本书基于这种观念来开发实例，演示如何将所采纳的想法在设计中用正确的架构表达出来。由于不想把注意力从SQL Server转移开，因而没有对这个例子做更多的处理，应该说只是刚好能运行而已。通过学习本书，你将具备足够的知识和能力来根据实际金融应用程序的需求扩展这个例子，使其细节和复杂度能够完全满足实际所需。

当然，在讲述这些之前，首先要安装SQL Server。

1.5 安装

本章将介绍SQL Server开发版的安装过程，事实上在每一种版本的安装过程中所看到的几乎都一样。微软在<http://www.microsoft.com/china/sql/2008/trial-software.aspx>提供了180天试用版，如果手头没有SQL Server 2008，可以使用这种试用版来学习本书的例子。

本书要讲述的选项和特性组合可以在安装过程中完成。安装过程中包含SQL Server提供的许多不同的工具。先来看看这些工具，以便对它们有基本的了解，从而决定将安装什么。

6 第1章 SQL Server 2008 概述和安装

安装包含很多不同的范围：

- 安全问题；
- 不同的安装类型——是首次安装并且安装的是SQL Server的第一个实例，还是用于开发、测试或生产的后续实例的安装；
- 自定义安装；
- 只安装少许有用的产品。

本章介绍的安装过程将涉及这些范围中的大多数，因此，到本章末，读者将具备足够的自信和知识来根据自己的需要完成任何后续的安装。

本书选择使用开发版是因为该版本最符合开发者所需，它没有企业版对于操作系统的种种要求。在光驱中插入所选择版本的Microsoft SQL Server 2008光盘，启动安装程序。接下来讲述标准安装。

1.5.1 开始安装

首先，确保以管理员身份登录，从而能够在机器上创建文件和文件夹，这显然是成功安装所必需的。

如果是使用CD-ROM进行安装，并且安装进程没有自动启动，就打开Windows资源管理器并双击autorun.exe（位于CD-ROM根目录）。如果不使用CD-ROM进行安装，则双击你下载的可执行的安装程序。

如果当前没有安装Microsoft .NET Framework 3.5版，则会出现该版本的安装对话框。.NET是微软创建的一种框架，允许用不同编程语言（如VB .NET、C#以及其他）编写的程序有一个公共编译环境。SQL Server 2008在其自身内部的一些工作要使用.NET，当然，开发人员也可以用任何微软的.NET语言编写.NET代码，放入SQL Server中。在SQL Server 2008中，除了可以用T-SQL以外，还能够使用.NET和LINQ来查询数据库。

注解包含.NET代码是超出本书讲述范围的高级主题。更多相关信息请参看由Robin Dewson和Julian Skinner编写的*Pro SQL Server 2005 Assemblies*（Apress, 2005）。

安装完成后，会出现“SQL Server安装中心”。如图1-1所示，该对话框涉及计划一个安装，设定安装方式（包括全新安装，从以前版本的SQL Server升级），以及用于维护SQL Server安装的许多其他选项。

单击安装中心左边的“安装”条目，然后，从“安装”选项列表中选择第一个项目，即“全新SQL Server独立安装或向现有安装添加功能”，这样就开始了SQL Server 2008的安装。

在输入产品密钥并接受SQL Server许可条款之前，将进行快速的系统检查。在SQL Server的安装过程中，要使用大量的支持文件，此外，支持文件也用来确保无瑕的和有效的安装。在图1-2中，可以看到快速系统检查过程中有一个警告，但仍可以继续安装。假如检查过程中没出现任何错误，则单击“下一步”。



图1-1 使用“SQL Server安装中心”开始安装

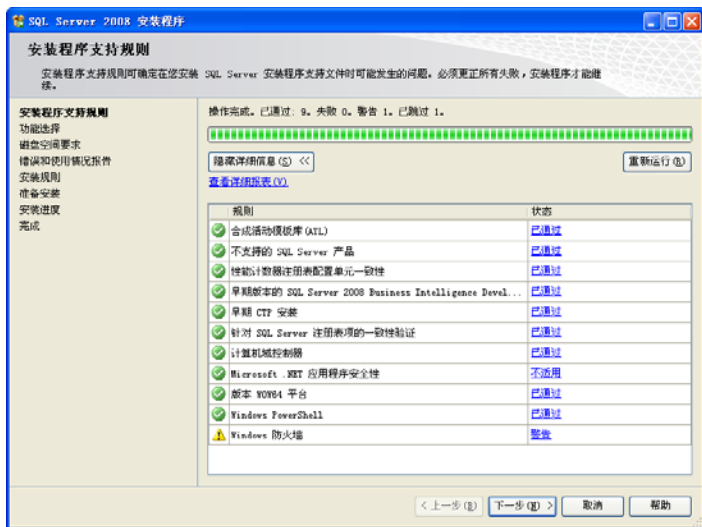


图1-2 系统配置检查

1.5.2 选择要安装的功能

接着，来到“功能选择”界面，需要作出一些决定。如图1-3所示，此处的安装会安装所有的功能，因为这将是一个开发实例，开发者将脱离所有正在进行的项目开发来测试SQL Server的各个方面的。不过，也可以根据需要，有选择性地安装各种组件。就本书而言，需要安装“数据库引擎服务”、“Reporting Services”、“客户端工具”以及用来创建报表的“Business Intelligence Development Studio”，因此，请确保至少选中了这些功能。

8 第1章 SQL Server 2008 概述和安装

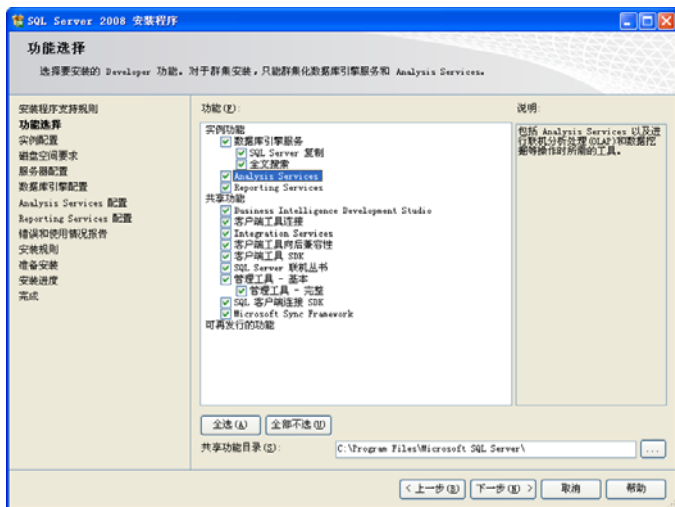


图1-3 选择安装所有的组件

下面简要说明一下图1-3中的大部分组件。

- ❑ 数据库引擎服务。这是SQL Server 2008的主要核心，安装SQL Server运行所需的主要引擎、数据文件等。
 - SQL Server复制。当你在数据库上执行数据修改时，如果你不仅想要把修改发送到该数据库上，而且想要把修改发送到一个相似的数据库上（这个相似数据库是为了复制修改而创建的），那么你可以使用这一选项把修改复制到那个数据库上。
 - 全文搜索。这一选项允许对数据库中的文本进行搜索。
- ❑ Analysis Services。使用该工具可以获取数据集，并对数据切块、切片，分析其中所包含的信息。
- ❑ Reporting Services。这一服务允许从SQL Server生成报表，而不必借助第三方工具，如Crystal Report。该组件将在第14章详细讲述。
- ❑ 客户端工具。这些工具中，一些为客户端机器提供到SQL Server的图形化界面，另一些则在客户端协同SQL Server一起工作。这一选项适于布置在开发人员的机器上。
- ❑ Microsoft Sync Framework。当与脱机应用程序（例如，移动设备上的应用程序）一起工作时，必须在适当的地方存在某种同步机制。这一选项允许发生这些交互。
- ❑ SQL Server联机丛书。这是一个帮助系统。如果在SQL Server的任何方面需要更多的信息、说明或额外的详细资料，请求助于联机丛书。
- ❑ Business Intelligence Development Studio。如果你想要使用基于分析的服务来分析数据，那么可以使用这个图形用户界面与数据库进行交互。本书不介绍这个选项。
- ❑ Integration Services。最后这个选项使你能够创建完成行动的过程，例如，从其他数据源导入数据并使用这些数据。在第7章中，当我们考虑创建一个备份维护计划时，将看到Integration Services的实际运作。

在以上这些组件中，Analysis Services、Business Intelligence Development Studio超出了本书的范围，我们只简单介绍这里提到的Integration Services。

注解此刻，SQL Server不再提供安装示例数据库的选项。微软也改变了示例数据库和示例的交付方式，因而你可以在SQL Server网站上（<http://www.microsoft.com/sql> 或 <http://www.codeplex.com/SqlServerSamples>）找到更新的版本。

1.5.3 为实例命名

众所周知，SQL Server是安装在计算机上的，那么在一台计算机上多次安装SQL Server是完全有可能的。如果服务器功能强大，有足够的资源（如内存、处理器等）运行两三个不同的应用程序，这种情形就可能出现。这些不同的应用程序都想拥有自己的SQL Server。每一个安装称为一个**实例**（instance）。现在应该为安装的实例命名。每一个实例必须有一个属于它的唯一的名字，就连“无名”的**默认实例**（default instance），其“无名”也算是一个唯一的名字。

作为建立外部环境的第一步，为实例命名是很重要的。例如，可能有一个实例用于开发，一个实例用于系统测试，最后还有一个实例用于用户测试。让除生产数据库之外的任何东西与生产数据库共享生产服务器硬件，这是一种很糟糕的习惯。如果无视这种提醒，当出现不当的开发行为并导致服务器崩溃时，将会使连续的生产停顿下来。虽然，眼下的安装过程一开始的时候，你就应该作出决定，但是，在为实例命名时再次提醒注意，这也是有帮助的。

当没有为安装指定明确的名字时，将选定为默认实例。一旦在学习环境之外安装SQL Server，则应避免这种情况，因为这样会导致没有命名的安装，因而关于它的使用没有任何提示。因为目前尚在学习阶段，而最易于理解的选项是使用默认实例，所以如图1-4所示，选择“默认实例”，

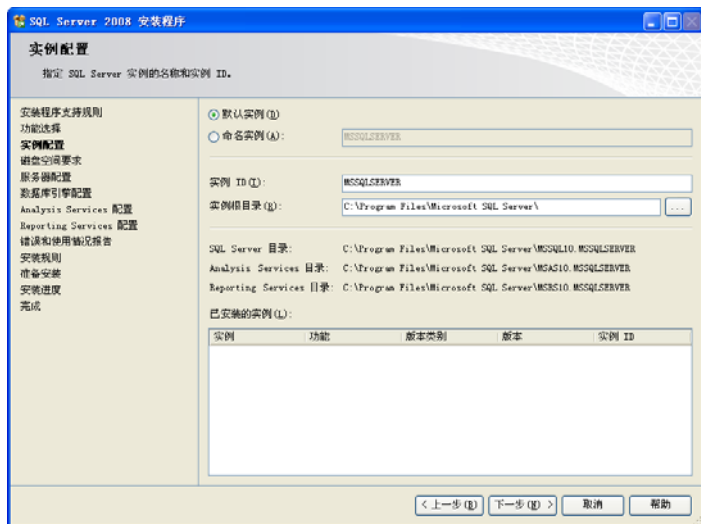


图1-4 为安装的实例命名

10 第1章 SQL Server 2008 概述和安装

并单击“下一步”。一旦在服务器上安装了实例，就会在这里列出已安装的实例。另外，对于前一步中所选的3个服务，我们还可以在这里看到每一个服务的目录的详细路径。

1.5.4 选择服务账户

正如用户在使用系统前必须先登录到Windows一样，SQL Server以及在“功能选择”界面（见图1-3）中定义的其他服务在启动前也必须先登录到Windows。SQL Server、Reporting Services等服务不需要任何人登录到安装SQL Server的计算机上就可以运行，只要计算机成功启动即可。当SQL Server安装在位于远程服务器机房中的服务器上时，这种情况极为平常。

在本章的结尾，我们会更详细地讨论这些选项。图1-5中所示的选择将用低级别特权安装SQL Server。

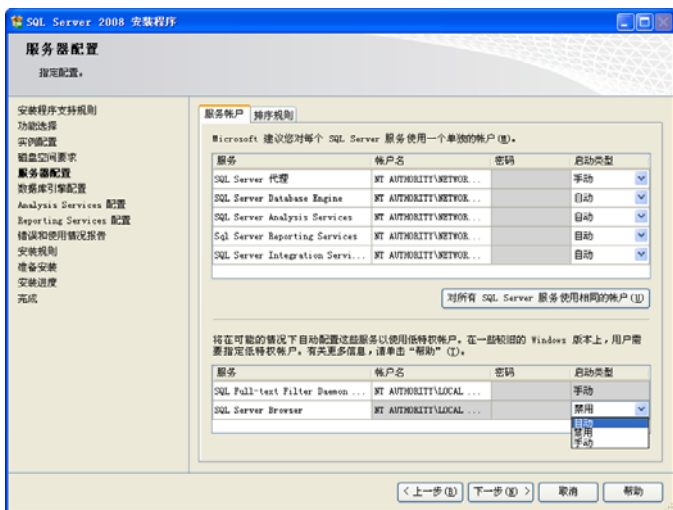


图1-5 选择服务账户

以后，通过“控制面板”中的“管理工具”里的“服务”图标，你也总能对此进行更改。然而，使用“配置工具”中的“SQL Server配置管理器”或许会更好些。通过使用“SQL Server配置管理器”，将会把账户添加到正确的组中，并给予恰当的权限。单击“下一步”。

如果你注意到SQL Server Browser（即SQL Server Management Studio的另一个名字），会发现它默认是被禁用的。多数的SQL Server安装是在服务器上，并且常常是远程服务器上，因此，没必要让SQL Server Browser运行。一般来说，你会从客户端机器上连接到SQL Server。尽管如此，这里还是假设该SQL Server安装并非在服务器上，而是在一台本地计算机上，因此，将该选项更改为自动启动。

1.5.5 选择身份验证模式

现在，将定义如何在SQL Server的安装中强制实施安全性。如图1-6所示，这里有两个选

择：Windows身份验证模式和混合模式。在本章的后面，你将了解更多关于模式的知识。十分简单明了，Windows身份验证模式表明将使用Windows的安全机制维护SQL Server的登录，混合模式则或者使用Windows的安全机制，或者使用SQL Server定义的登录ID和密码。此外，如果使用混合模式，还需要为名为sa的特殊登录ID设置密码。关于这些，你很快就会了解到更多，但目前必须为其输入一个有效的密码。请使用有意义的、难以猜测的密码，同时你自己要记牢。

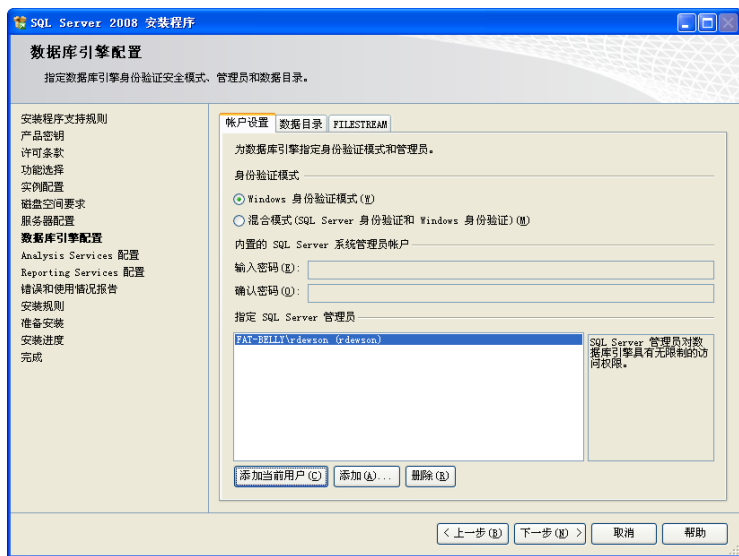


图1-6 选择身份验证模式（包括SQL Server管理员账户）

另外，还必须指定SQL Server管理员账户。这是一个特殊的账户，在极其紧急的情况下（例如，当SQL Server拒绝连接时），能够使用这个账户进行登录。你可以用这个特殊的账户登录，对当前的情形进行调试，并让SQL Server恢复运行。通常，管理员账户是某个服务器账户ID，但现在，我们使用了登录到计算机上的这个当前账户。

对于Analysis Services也会有类似的界面，并且，也使用相同的设置。

1.5.6 确定数据目录的位置

如图1-7所示，在“数据目录”选项卡中，可以明确SQL Server默认把数据存储在何处，备份目录的默认位置在哪里，以及临时数据库的默认目录是什么。你要为每一个实例确定相应的文件夹应存在于何处。正如前面说过的，在一个物理服务器上，能够有若干个安装，并且，让一个物理服务器容纳一个以上的SQL Server的生产安装是十分平常的。例如，可能有一个安装用于账户，另一个安装用于产品控制，等等。每一个实例的数据应该位于不同的数据目录中。这包括创建的任何临时数据库和产生的任何日志文件，因此，尽管共享一个物理服务器，但安装是隔离的。鉴于本书中只涉及一个实例，这里让这些设置保持原状。



图1-7 确定SQL Server数据目录的位置

在“FILESTREAM”选项卡中配置的是另一种类型的数据目录，不过，目前暂且先不管它，在本书后面将讨论更多相关的内容。虽然如此，为了让你稍有了解，这里先说明一下，当处理大量非结构化数据时，要使用FILESTREAM。过去，这类数据完全与SQL Server分开存储，而现在，FILESTREAM使SQL Server能够管理这些数据，该选项卡告知SQL Server这种FILESTREAM数据存储在哪里。

如果前面选中了Analysis Services，你会在下一步中看到同样的选项卡。

1.5.7 创建报表服务数据库

由于我们在前面选择了安装Reporting Services，因此，需要创建报表服务器所使用的数据库。对Reporting Services而言，有3个不同的安装选项：“本机”、“SharePoint”和“安装但不配置”。如果选择最后一个选项，将在服务器上安装SQL Server Reporting Services，但不会对其进行配置。如果只是为了报表选项而构建特定的服务器，则该选项十分理想。安装完成后，必须创建报表数据库。

如图1-8所示，本机模式配置是最简单的选项，也是我们要使用的选项。选择该选项，将在SQL Server中安装Reporting Services，并创建必需的数据库。仅当你在本地实例而非远程实例上进行安装，并且Reporting Services也存在于那个本地实例上时，该选项才是有效的。对服务账户，本地实例上（即localhost）的报表服务器的URL、报表管理器URL以及报表服务数据库的名称使用默认值。

如果部署了SharePoint安装，并且想要Reporting Services使用该体系结构，则选择这一选项，这一选项允许你使用SharePoint的功能。这些内容超出本书的讲述范畴。

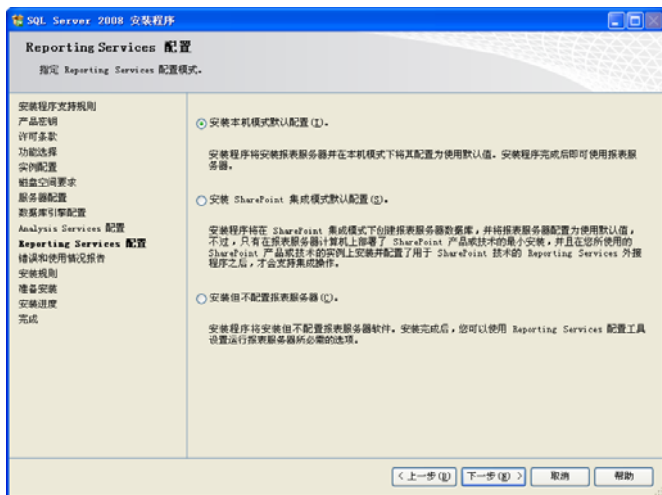


图1-8 为Reporting Services安装本机模式配置

1.5.8 配置错误和使用情况报告

在SQL Server中，可以自动报告任何错误并把错误报告发送到微软，其中包含SQL Server异常关闭时的致命错误。推荐启用图1-9中的错误设置。因为不会发送组织机构的任何信息，所以数据依然是安全的。这与在Excel崩溃时发送报告是类似的。最好是使该功能处于激活状态。发送错误报告给微软，微软将有望更快地开发出补丁修复程序，并在将来发布更好的版本。另外，对于SQL Server来说，用户可以获得如何使用SQL Server的信息。打开这一功能也是非常有帮助的，这样微软能够接收到有助于改进其产品的信息。不管怎样，在与这一功能更加密切相关的生产环境中，打开这一功能将会十分有益。

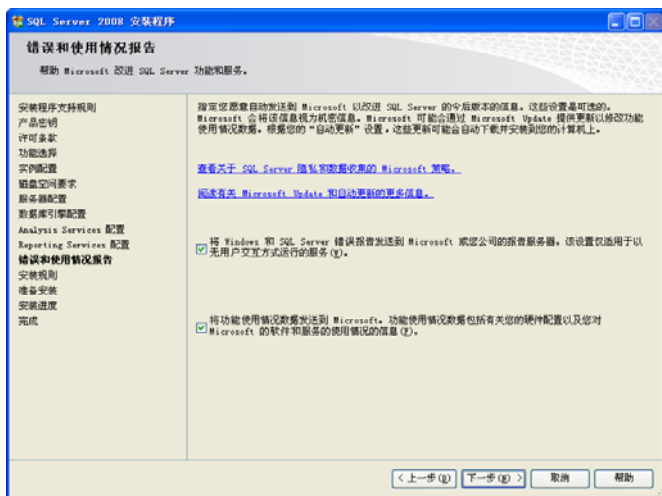


图1-9 “错误和使用情况报告”设置

点击“下一步”，将会出现关于安装规则详细信息的界面。在这里没什么可处理的，只是点击“下一步”就可以了，这时将显示最终的界面（如图1-10所示）。现在完成了设置收集，已经准备好进行安装了。

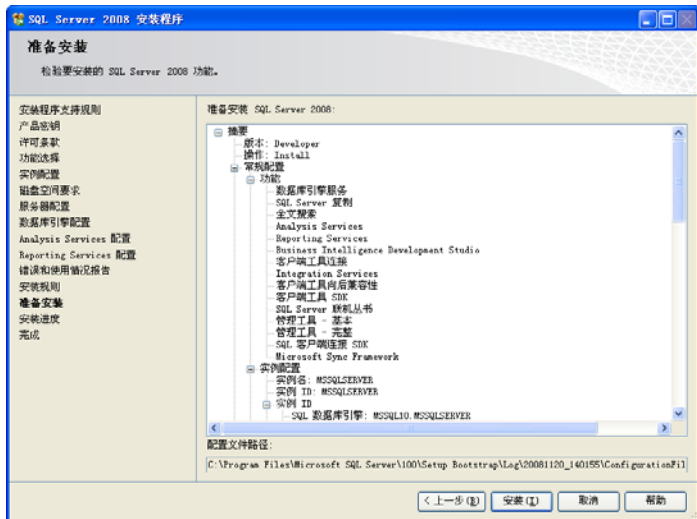


图1-10 完成设置详情

1.6 安全性

为了透彻地讨论安装中遇到的“服务账户”对话框，需要深入研究Windows安全性。

本节中，首先考查Windows服务（而非程序）的概念，然后讨论在安装SQL Server时可以选择的不同的身份验证类型。

1.6.1 服务账户

SQL Server是作为一种Windows服务来运行的。那么，什么是服务？服务的一个很好的例子是防病毒软件，它从用户重启计算机开始到计算机关闭为止持续运行。而程序要么是载入到内存并运行着，要么是没有启动。运行服务的优势在于：如果一项工作可以作为服务运行，Windows就能够对那个进程做更多的控制。可以将服务设置为在任何用户登录以前自动启动，而所有其他程序都需要用户先登录到Windows，进而启动服务。

此外，服务完全没有用户界面。在运行时没有需要显示的表单，也没有要处理的用户输入。与进程之间唯一的交互，或者通过单独的用户界面（完全隔离的工作单元）链接到服务中来运行（例如SSMS），或者是从那个服务自身的Windows管理中进行。源自服务的任何输出都必须输出到事件日志（Event Log）中，事件日志是用来存储服务所发出的通知的Windows区域。

服务没有界面意味着对整个进程的控制无需用户的干预。倘使服务的设计良好，Windows就能独自处理所有的可能性，并且能在任何用户登录到计算机之前启动服务。

在大多数生产环境中，SQL Server是运行在远程服务器上的，该服务器可能锁藏于安全的、受控制的地方，唯一能进入那里的人或许就是硬件工程师。在那里，甚至可能没有安装远程访问程序，因为这可能带来对计算机的未经授权的访问。SQL Server将在那里顺畅地运行着，幸运的话，完全不发生任何错误。但是，如果某天发生了错误呢？假设SQL Server是作为程序运行的，则不得不采取某种对策。即使SQL Server崩溃了，至少还有某种机制能够重新启动它。这意味着需要运行另外一个监视进程，而监视进程自身也可能导致大量问题。然而，当SQL Server作为服务运行时，是在Windows的控制之下的。一旦发生问题，无论是与SQL Server有关，还是与Windows有关，或是由任何外因所致，Windows完全能够通过服务进程来巧妙地处理发生的问题。

如果确实登录到了计算机上，那么，还可以用这个Windows用户ID登录SQL Server并启动它的服务。在你学习本书时，由于SQL Server可以运行在家庭或本地系统上，当你进入SQL Server时，多半采用的是这种方式。这就是所谓的**本地系统账户**（local system account）。

另一方面，也可以创建一个完全用于SQL Server的Windows登录账户。这样做有几个理由。举例来说，在对Windows账户进行一番设置后，账户的密码在使用了若干天后会过期，或者在若干次输入错误密码的登录尝试后会锁定账户。如此一来，就从许多其他事物中保护了计算机和网络。然而，SQL Server应该使用单独的账户，该账户的密码同样会过期，并且具有在若干次登录尝试失败后锁定账户的能力。这种非用户特定的“一般”账户去除了SQL Server和组织机构中人员间的联系。在图1-5中，如果考虑域用户账户选项，该账户多半是在网络环境或生产环境中。有一种选择是为每一个服务指定一个不同的账户。当迁移至公司环境中时，由于必须应对随之而来的安全问题，这会是非常至关重要的选择。

SQL Server为不同的工作提供了几种不同的处理过程。SQL Server用来运行SQL Server自己，而SQL Server代理运行一些处理过程，如批处理作业。实际上，SQL Server只应访问其自身。因此，只应该采用严格限制特权的域登录账户。

SQL Server代理运行批处理和复杂任务（包括与其他服务器协同工作的复杂任务），需要功能更强大的域账户。你的网络管理员可能已经创建了这些账户，并且知道对于这些任务来说，最好使用哪个账户或者最好创建什么账户。

现在，进一步讨论在安装过程中对于身份验证模式所做的选择。

1.6.2 身份验证模式

将要应用于服务器上的身份验证模式，或许是整个安装过程中最至关重要的信息以及要作出的最重大的决定。正如在前面的设置过程中看到的，关于身份验证模式有两个选择：**Windows身份验证模式**（Windows authentication mode）和**混合模式**（mixed mode）。

1. Windows身份验证模式

要登录到Windows 2003/XP/Vista机器上，必须提供用户名。关于这一点，没有选择的余地（不像Windows 9x/ME那样，是否要提供用户名是可选的）。因此，要登录到Windows，必须先验证用户名和密码。当完成这一步时，Windows事实上是在域控制器中根据用户名凭据验证用户；或者，如果Windows/SQL Server是在本地的独立机器上运行，则进行本地验证。凭据检查用户所属的访

问组（用户权限）。用户可以上至管理员，具有改变计算机中任何事物的能力，也可以下至基本用户，只拥有极其有限的权限。于是这就给出了一种信任连接，换句话说，在登录Windows之后启动的应用程序可以相信，经Windows验证，账户已经通过了必要的安全检查。

当应用Windows身份验证模式时，一旦登录到Windows，SQL Server就将使用信任连接。如前面所述，这意味着SQL Server相信用户名和密码已被验证过了。可是，如果用户名不存在，那么仅基于用户ID，你将无法登录到那台机器上。如果登录无效，那么将检查用户所属的Windows组，并检查其安全性，以确定该组是否可以访问SQL Server。如果该用户具有管理员权限，则至少连接到SQL Server上是完全不成问题的。

假如其他用户可以访问你的机器，那么使用他们的用户ID和密码也能够登录到机器上。虽然这些用户可以通过找到C驱动器上的可执行文件到达SQL Server，但SQL Server首先要检查其在SQL Server中是否拥有有效的登录名。

目前的情况有点进退维谷。在安装过程中，需要了解安全性，可是要充分说明它意味着要在SSMS中进行工作，这是第2章要讲述的内容。为了使其简单，现在先看一个涉及安全性的例子。

练习：Windows身份验证模式

(1) 确保以管理员身份登录到机器上。如果是本地计算机，有可能当前的登录名就是管理员ID；如果是网络中的计算机，并且不能确定访问权限，请求助于计算机管理人员帮助解决ID和密码的问题。在Windows Vista中，如果想避免每一步都出现一个对话框向你确认是否要继续，你需要修改用户账户控制的设置。

(2) 单击“开始”→“控制面板”，选择“用户账户”。

(3) 当出现用户和密码对话框时，在XP上单击“创建一个新账户”，在Vista上单击“管理其他账户”，随后再单击“创建一个新账户”。

(4) 如图1-11所示，当出现“为新账户起名”对话框时，输入用户名JThakur。



图1-11 创建新的用户账户

(5) 确保在XP指定的账户类型是“受限”或在Vista指定的账户类型是“标准用户”。这意味着该账户没有管理员权限。然后单击“创建账户”。

(6) 因为要添加第二个用户名，故停留在“用户账户”对话框中。重复前述的步骤，并使用如下资料：

用户名：VMcGlynn

账户类型：计算机管理员

(7) 从Windows中注销，然后用刚才创建的第一个ID（JThakur）登录。

(8) 登录后，通过选择“开始”→“所有程序”→“Microsoft SQL Server 2008”→“SQL Server Management Studio”，启动SSMS。你需要在对话框中输入安装的服务器名称。在“服务器名称”下拉列表中单击“浏览更多”，然后选择“数据库引擎”，再选择安装的服务器名称。在第2章中，将更详细地讲述这些内容。该对话框与图1-12显示的类似。



图1-12 尝试连接到SQL Server

(9) 查看出现的错误信息，错误信息应该与在图1-13中所看到的类似。JThakur作为一个登录名没有在SQL Server中明确定义，并且也不属于允许访问的组。当前唯一的组是在Windows的Administrators组中的一个用户。回想前面，JThakur是受限用户。

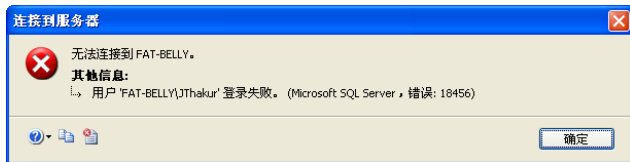


图1-13 登录到服务器失败

(10) 接下来测试刚才创建的另一个用户。关闭SQL Server，从Windows中注销，使用创建的第二个ID（VMcGlynn）登录。登录Windows以后，启动SSMS并连接到服务器上。这一次可以成功登录。

前面创建的两个用户名，一个具有受限的访问权限（JThakur），另一个具有管理权限（VMcGlynn）。无论如何，这两个用户名都不存在于SQL Server中（毕竟没有在SQL Server中输入它们，它们似乎也没有因为某种魔力而出现）。那么，何以一个能成功登录而另一个会失败呢？

Windows安全模式确保了两个ID都是有效的。如果ID或密码不正确，根本就不能登录到Windows。因此，当试图连接到SQL Server时，唯一进行的检查是：用户是以操作系统的组成员身份访问SQL Server的，还是通过特定的登录用户账户访问的。正如图1-14所示，JThakur和VMcGlynn都不存在于SQL Server中。

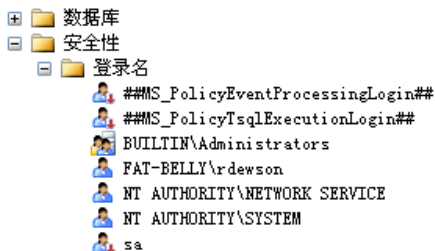


图1-14 SQL Server的对象资源管理器

但是，在SQL Server的对象资源管理器中，可以看到一个名为BUILTIN\Administrators的Windows组。这意味着任何属于Administrators组的用户名都能够登录到SQL Server上。从而避免了将用户设置为他们自己计算机的管理员（如果可能的话）。

在生产环境中，如果允许用户成为管理员，则将该组从系统中移除或许是可取的。由于VMcGlynn是Administrators组的成员，因而该用户名也是BUILTIN\Administrators组的成员。

2. 混合模式

如果将SQL Server安装为混合模式，意味着既可以使用Windows身份验证（如前面所述），也可以使用SQL Server身份验证。

混合模式与Windows身份验证模式有何不同？首先，需要提供用户ID和密码进行连接，而不是由SQL Server取用登入账户的Windows ID或登入账户用户所属的组；其次，并不认定提供的ID都是有效的。当工作中用到ISP时，混合模式在许多情况下都是适宜的。为阐明这一观点，假设用户要在本地客户端机器上使用远程数据进行工作，那么远程的机器需要知道登录凭据，因而使用SQL Server身份验证将是最简单的方法。不过，请不要在这里弄混了。如果想要在ISP上处理数据，ISP可能提供了某种工具，或者，可以使用SSMS连接到数据上。接下来，你就能够随心所欲地进行工作了。如果网站的代码是以ASP.NET编写的，将使用Windows账户进行登录，因此，尽管可能锁定了SQL Server混合模式账户，也不会妨碍网站正常工作。

当在第4章谈论安全性时，将学习到如何向SQL Server中添加用户名（而不是添加Windows用户）。

关于安全性，还有一点需要讨论，这就是sa登录。

1.6.3 sa 登录

1

sa登录是默认的登录，对SQL Server有完全的管理权限。如果安装过程中选择了混合模式身份验证，将会强制在这里为该账户设置密码。这是因为sa用户ID是能力十分强大的登录名。此外，sa存在于所有的SQL Server安装中，因此，任何黑客都知道存在这一用户ID，并将试图通过这个用户ID连接到服务器上。从SQL Server 2005起，开始强制为sa登录名创建密码，而在2005之前的版本，许多SQL Server安装中的sa密码都为空，结果就让黑客能够直接进行访问。当用sa登录到SQL Server时，将能够完全控制SQL Server的任何一个特性。无论安装采用何种身份验证模式，SQL Server中都将加入这一ID。如果有一个Windows账户定义为sa，例如Steve Austin定义为sa，当服务器设置为采用Windows身份验证模式，且对该用户没有任何干预时，该用户就可以登录SQL Server，所以应该尽量避免名为sa的登录ID。

在混合模式安装中，sa将是一个有效的用户名，并经证实如此。可以想象，如果其他用户获得了这一用户名和密码，他便具有完全访问的能力，可以查看、修改或删除任何数据项。最坏的情况下，该用户可能毁坏所有的数据库，并毁掉SQL Server本身。该用户甚至可以设置任务，通过邮件向远程计算机发送数据。

如果在“身份验证模式”界面选择了混合模式身份验证，则必须为sa账户设置强密码。在这一点上，SQL Server 2008有了很大的改进，它强制要求为账户设置密码，尽管设置的密码可能是极易被猜出的。例如，不要使用类似password或adminpwd的密码。不仅要始终保证密码安全，而且要在安全的地方记下密码。否则，如果忘记了sa的密码，而sa是当前存在的唯一的管理ID，那么只有重装SQL Server了。妥当的密码应该是数字和字母的组合，但不要包含酷似数字的字母或者酷似字母的数字。例如，pa55word是很容易由password猜出的，4pr355则酷似Apress。

不用sa用户名登录的原因还有一个。有时，必须要了解是谁在SQL Server数据库上运行特定的查询。例如，在生产数据库中，某人可能正在进行数据更新，进而填满了磁盘空间或者填满了事务日志。这时，需要联系那个人以核实其是否能终止进程。如果此人是以sa登录的，则将无从知道他是谁。但是，如果此人是以一个可确认的名字登录，那么将会在SQL Server中有其ID，从而就可以进行追踪。通过限制sa登录，人们不得不使用他们自己的账户，这样就能确保更高级别的系统监控和完整性。

有些时候需要使用混合模式身份验证，这是一种完全可以接受的需求。例如，由于很多应用程序可能运行在一台Web服务器上，因此因特网提供商使用混合模式。如果ISP是转售商（换句话说，世界各地的许多人都使用同一台计算机），谁也不会希望同样使用该计算机的其他用户能看到自己的数据。可是已经决定了不以sa身份登录，那么该怎么做呢？可以这样，创建一个具有用户需要的访问权力的登录ID，也就是说，该ID具有刚好能够看到数据并用所需要的数据进行工作的能力。ISP可能要求提供用户ID和密码，以便在其SQL Server实例上创建账户。第4章中将遇到更多这方面的内容。

注解无论身份验证模式是哪一种，始终使用强密码是十分重要的。

1.7 小结

至此，读者应该了解了各种版本SQL Server之间的细微差别，同时还应该知道怎样检查计算机是否适合安装SQL Server。

在前面所讲述步骤的指引下，读者应该已经在计算机上成功安装了SQL Server 2008，甚至可能已经完成了两次安装，从而既安装了一个开发服务器又安装了一个测试服务器。这是个好主意，如果到目前为止只有一个安装的话，可以考虑安装另一个。无论是在大型公司工作，还是在人数极少的小公司工作，保持生产代码和开发代码的分离将极大降低复杂性（如果在开发时，需要为生产做准备的话）。

本章介绍了SQL Server中的安全性，使读者能轻松地知晓自己想要以何种方式实现安全性，以及如何处理不同的用户名。虽然目前可能还没有什么数据，但需要确保一旦有了数据，只有合适的人才能够查看到它们。

现在，已经做好了系统学习SQL Server 2008的准备。管理SQL Server最好的方法之一是使用SSMS，接下来将讨论这方面的话题。