高等学校计算机应用规划教材

SQL Server
数据库应用与开发教程
(第五版) (2016 版)

卫　琳　主　编
马建红　副主编

清华大学出版社
北　京

内 容 简 介

本书全面讲述 Microsoft SQL Server 关系数据库管理系统的基本原理和技术。全书共分为 15 章，深入介绍 Microsoft SQL Server 2016 系统的基本特点、安装和配置技术、Transact-SQL 语言、安全性管理、数据库和表的管理，以及索引、数据更新、备份和恢复、数据完整性、数据复制、性能监视和自动化技术等内容。

本书内容丰富、结构合理、思路清晰、语言简洁流畅、示例翔实，主要面向数据库初学者，既适合作为高等院校的数据库教材，也适合作为 Microsoft SQL Server 应用开发人员的参考书。

本书提供实例操作的教学视频，读者扫描封底或前言中的二维码即可观看。本书配套的电子课件、习题答案、实例源文件和实例数据库可以通过 http://www.tupwk.com.cn/downpage 网站下载，也可以通过扫描封底或前言中的二维码推送到指定邮箱。

图书在版编目(CIP)数据

SQL Server 数据库应用与开发教程：2016 版 / 卫琳主编. —5 版. —北京：清华大学出版社，2021.4
高等学校计算机应用规划教材
ISBN 978-7-302-57762-1

Ⅰ. ①S… Ⅱ. ①卫… Ⅲ. ①关系数据库系统—高等学校—教材 Ⅳ. ①TP311.132.3

中国版本图书馆CIP数据核字(2021)第048973号

责任编辑：胡辰浩
封面设计：高娟妮
版式设计：妙思品位
责任校对：成凤进
责任印制：沈 露

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 装 者：三河市中晟雅豪印务有限公司
经 销：全国新华书店
开 本：185mm×260mm **印 张：**21.25 **字 数：**614 千字
版 次：2007 年 9 月第 1 版 2021 年 4 月第 5 版 **印 次：**2021 年 4 月第 1 次印刷
定 价：79.00 元

产品编号：083909-01

前　言

信息技术的飞速发展大大推动了社会的进步，逐渐改变了人类的生活、工作、学习等方式。数据库技术和网络技术是信息技术中最重要的两大支柱。自从 20 世纪 70 年代以来，数据库技术的发展使得信息技术的应用从传统的计算方式转变到了现代化的数据管理方式。在当前热门的信息系统开发领域，如管理信息系统(Management Information System，MIS)、企业资源计划(Enterprise Resource Planning，ERP)、供应链管理系统(Supply Chain Management System，SCMS)、客户关系管理系统(Customer Relationship Management System，CRMS)等，都可以看到数据库技术应用的影子。

作为一个关系数据库管理系统产品，Microsoft SQL Server 起步较晚。但是，由于 Microsoft SQL Server 产品不断地采纳新技术来满足用户不断增长和变化的需要，该产品的功能越来越强大、用户使用起来越来越方便、系统的可靠性也越来越高，从而使该产品的应用越来越广泛。在我国，Microsoft SQL Server 的应用已经深入到银行、邮电、电力、铁路、气象、民航、公安、军事、航天、财税、制造、教育等许多行业和领域。Microsoft SQL Server 为用户提供了完整的数据库解决方案，可以帮助各种用户建立自己的商务系统，增强用户对外界变化的敏捷反应能力，以提高用户的竞争能力。

本书从 Microsoft SQL Server 2016 的基本概念出发，由浅入深地详细讲述 Microsoft SQL Server 2016 体系结构、该系统的安装过程、服务器的配置技术、Transact-SQL 语言、安全性技术、数据库管理、各种数据库对象管理，以及索引技术、数据更新技术、数据完整性技术、数据复制技术、数据库性能监视和调整技术、自动化 SQL Server 技术等内容。在讲述 Microsoft SQL Server 的各种技术时，运用了丰富的实例，注重培养读者解决实际问题的能力，让读者能够快速掌握 Microsoft SQL Server 的基本操作技术。

本书内容丰富、结构合理、思路清晰、语言简洁流畅、示例翔实。每一章的引言部分概述了本章的主要内容。在每一章的正文中，结合所讲述的关键技术和难点，穿插了大量极富实用价值的示例。每一章末尾都安排了有针对性的经典习题，有助于读者巩固所学的基本概念，培养读者的实际动手能力。

本书主要面向数据库初学者，适合作为各种数据库培训班的培训教材、高等院校的数据库教材及各种数据库应用程序开发人员的参考书。

除封面署名的作者外，参与本书编写的人员还有高军亮、王海亮、徐静、王震源、王稀瑶、王秉宏等。由于作者水平有限，本书难免有不足之处，欢迎广大读者批评指正。我们的邮箱是 992116@qq.com，电话是 010-62796045。

本书提供实例操作的教学视频，读者扫描下方的二维码即可观看。本书配套的电子课件、习题答案、实例源文件和实例数据库可以通过 http://www.tupwk.com.cn/downpage 网站下载，也可以通过扫描下方的二维码推送到指定邮箱。

扫一扫　　配套资源

看视频　　扫描下载

作　者

2020 年 11 月

目　录

第 1 章

初识SQL Server 2016

本章主要内容：

- SQL Server 2016 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 2016 在原有的基础上，集成了云技术和内存技术，以适应未来的发展需求。新增的大量功能使本地和云端的工作负载得到了性能提升，提高了可用性和可管理性。本章概要介绍了 SQL Server 2016 体系结构，为如何运行 SQL Server 提供了帮助。

1.1 SQL Server 2016 应用领域

本节将从整体上介绍 SQL Server 生态系统。SQL Server 2016 主要关注以下 3 个领域。

- **任务关键的性能**：新增了内存联机事务处理(Online Transaction Processing，OLTP)功能，使得在不修改应用程序的情况下能够提升性能，再加上可更新的列存储索引、使用固态硬盘(Solid-State Drive，SSD)的缓冲池扩展以及 AlwaysOn 功能的增强(包括支持多达 8 个副本)，让 SQL Server 2016 成为超强大的 SQL Server 版本。
- **更快地获得有用信息**：借助于新的基于 Office 的商业智能(Business Intelligence，BI)工具(如 Power Query 和 Power Map)，以及 Power View 和 Power Pivot 的改进，使用户在任何时候都可以方便地访问数据。另外，企业选项(如 Parallel Data Warehouse with PolyBase)让组织能够利用 Microsoft BI 工具的强大功能，方便地探索其大数据，获得关于自己数据的前所未见的深入见解。
- **混合云的平台**：不管环境是纯本地的、虚拟化的还是完全在云中的，SQL Server 2016 都提供了对应的选项。新增功能(如 Microsoft SQL Server Backup to Windows Azure Tool)允许将备份存储到 Windows Azure Blob，并且可以加密和压缩本地或云中存储的 SQL Server 备份。在 AlwaysOn 可用性组配置中，现在也可以选择一个 Windows Azure 虚拟机作为副本。

1.2 SQL Server 2016 的重要新增功能

本节将简要介绍 SQL Server 2016 的一些新增功能，其中的许多功能使用起来都很便捷，可以快速上手。

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死机。如果数据库出现死机，生产 DBA 可以恢复数据库。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质量保证，再到生产。其新增功能如下。

1) 全程加密技术

全程加密(Always Encrypted)技术支持在 SQL Server 中保持数据处于加密状态，只有调用 SQL Server 的应用才能访问加密数据。该功能支持客户端应用所有者控制保密数据，指定哪些人有权限访问。SQL Server 2016 通过验证加密密钥实现了对客户端应用的控制。该加密密钥永远不会传递给 SQL Server，使用该功能，可以避免数据库或者操作系统管理员接触客户端应用程序的敏感数据(包括静态数据和动态数据)。该功能支持敏感数据存储在云端管理数据库中，并且永远保持加密状态，即便是云供应商也看不到加密数据。

2) 动态数据屏蔽

如果用户对保护数据感兴趣，希望被授权人可以看到加密数据，而另一些人只能看到加密数据混淆后的乱码，那么需要关注动态数据屏蔽(Dynamic Data Masking)功能。利用动态数据屏蔽功能，可以将 SQL Server 数据库的表中待加密的数据列混淆，那些未授权用户就看不到这部分加密数据。利用动态数据屏蔽功能，还可以定义数据的混淆方式。例如，如果在表中存储信用卡号，但是希望只看到卡号的后四位，使用动态数据屏蔽功能定义屏蔽规则就可以限制未授权用户只能看到信用卡号的后四位，而有权限的用户可以看到完整的信用卡信息。

3) JSON 支持

JSON 的全称是 JavaScript Object Notation (JS 对象简谱)，是一种轻量级的数据交换格式。在 SQL Server 2016 中，可以在应用和 SQL Server 数据库引擎之间使用 JSON 格式进行交互。微软公司在 SQL Server 中增加了对 JSON 的支持，可以解析 JSON 格式数据然后以关系格式存储。此外，利用对 JSON 的支持，还可以把关系数据转换成 JSON 格式数据。微软公司还增加了一些函数，用于对存储在 SQL Server 中的 JSON 数据执行查询操作。有了这些支持 JSON 操作的函数，应用程序使用 JSON 数据与 SQL Server 交互就更容易了。

4) 多个 tempdb 数据文件

如果运行的是多核计算机，那么运行多个 tempdb 数据文件就是最佳实践做法。在 SQL Server 2016 版本之前，安装 SQL Server 之后总是需要手工添加 tempdb 数据文件。在 SQL Server 2016 中，可以在安装 SQL Server 时直接配置需要的 tempdb 文件数量。这样就不必在安装完成之后再手工添加 tempdb 文件了。

5) PolyBase

PolyBase 支持查询分布式数据集。有了 PolyBase，可以使用 Transact-SQL 语句查询 Hadoop 或者 SQL Azure Blob。可以使用 PolyBase 写临时查询，实现 SQL Server 关系数据与 Hadoop 或

者 SQL Azure Blog 中的半结构化数据之间的关联查询。此外，还可以利用 SQL Server 的动态列存储索引针对半结构化数据进行优化查询。如果组织跨多个分布式位置传递数据，PolyBase 就是利用 SQL Server 技术访问这些位置的半结构化数据的便捷解决方案。

6) Query Store

如果经常使用执行计划，新版的 Query Store 功能会提供更多帮助。在 2016 之前的版本中，可以使用动态管理视图(DMV)来查看现有的执行计划。但是，DMV 只支持查看计划缓存中当前活跃的计划。如果出了计划缓存，就看不到计划的历史情况。有了 Query Store 功能，SQL 现在可以保存历史执行计划。不仅如此，该功能还可以保存那些历史计划的查询统计，可以利用该功能随时跟踪执行计划的性能。

7) 行级安全

SQL Server 数据库引擎具备了行级安全(Row Level Security)特性以后，就可以根据 SQL Server 登录权限来限制对行数据的访问。限制行是通过内联表值函数过滤谓词实现的。安全策略将确保过滤谓词后每次都能实现“SELECT”或者“DELETE”操作。在数据库层面实现行级安全意味着应用程序开发人员不再需要维护代码限制某些登录或者允许某些登录访问所有数据。有了这一功能，用户在查询包含行级安全设置的表时，甚至不知道他们查询的数据是已经过滤后的部分数据。

8) SQL Server 支持 R 语言

微软公司收购 Revolution Analytics 公司之后，可以在 SQL Server 上针对大数据使用 R 语言实现高级分析功能。SQL Server 支持 R 语言处理以后，数据科学家们可以直接利用现有的 R 代码并在 SQL Server 数据库引擎上运行。这样就不必为了使用 R 语言处理数据而将 SQL Server 数据导出来进行处理。该功能可以直接对数据进行 R 语言处理。

9) Stretch Database

Stretch Database 功能提供了将内部部署数据库扩展到 Azure SQL 数据库的途径。有了 Stretch Database 功能，访问频率最高的数据会存储在内部数据库，而访问频率较低的数据会离线存储在 Azure SQL 数据库中。当设置数据库为“stretch”时，那些比较过时的数据就会在后台迁移到 Azure SQL 数据库。如果需要在运行查询语句的同时访问活跃数据和 stretched 数据库中的历史信息，数据库引擎会将内部数据库和 Azure SQL 数据库无缝对接，返回查询结果，就像在同一个数据源中一样。该功能使得 DBA 的工作更容易了，他们可以将归档的历史信息转存到更廉价的存储介质上，而无须修改当前实际应用程序的代码。这样就可以使常用的内部数据库查询保持最佳性能状态。

10) 历史表

历史表(Temporal Table)会在基表中保存数据的旧版本信息。有了历史表功能，SQL Server 会在每次基表有行更新时自动将旧的数据版本迁移到历史表中。历史表在物理上是与基表独立的另一个表，但是与基表存在关联关系。如果已经构建或者计划构建自己的方法来管理行数据版本，那么应该先看看 SQL Server 2016 中新提供的历史表功能，然后再决定是否需要自行构建解决方案。

1.2.2　开发 DBA

自 SQL Server 2000 版本发布后，生产 DBA 与开发 DBA 的角色是合并在一起的。在大型

的数据库组织中，生产 DBA 归属于运营部门(由网络管理员和 Windows 支持人员构成)。如果将生产 DBA 放到开发小组中，就无法实现一些规章所要求的职责分离。

开发 DBA 在组织中也扮演着非常传统的角色。他们大多是以“开发人员”的身份出现，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，将数据库从一个版本升级为下一个版本。开发 DBA 与生产 DBA 不同，后者因为备份失败或其他类似问题在任何时间都要随时回应数据库所出现的问题，而开发 DBA 比较关注新版本中的以下内容。

- Transact-SQL(T-SQL)的增强包括内联指定基于磁盘的表的索引等功能。将兼容模式设置为至少 110 后，SELECT…INTO 语句可以与数据库并行操作。
- 对 SQL Server Data Tools for Business Intelligence(SSDT-BI)的更新包括支持针对多维模型创建 Power View 报表。其他对 SSDT-BI 的改进包括支持针对较早版本(2005+)的 Analysis Services 和 Reporting Services 创建项目。目前还不支持与 Integration Services 项目的向后兼容性。

开发 DBA 通常向开发小组汇报工作。开发小组接收来自业务分析师或其他开发人员的请求。从传统意义上讲，开发 DBA 没有生产数据库的修改权限。不过，开发 DBA 能够以只读方式访问数据库，以便在升级阶段进行调试。

1.2.3 商业智能 DBA

商业智能(BI)DBA 是随 SQL Server 功能的不断增加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 2016 改进了自 SQL Server 2012 以来最终用户的 BI 体验，并且通过 Office 365 带来的 Power BI 解决方案，以及现有的 BI 产品，提供了全新的扩展。

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，BI DBA 创建 SSIS 包，为用户支持数据提取、装载和转换(Extract Transform Load，ETL)过程。在大型组织中，BI DBA 创建 SSIS 包和 SSRS 报表。BI DBA 主要为有关 SSIS 程序包和 Analysis Services 多维数据集(或 SSAS 表格模型)的物理实现提供咨询。

BI DBA 具有下列职责：

- 实现有关 Analysis Services 多维数据集/表格模型和解决方案的建模并提供相关咨询。
- 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。

除了以上职责，BI DBA 还具有以下新功能：

- 使用 Power View 和 Power Pivot 快速发现数据。
- 使用列存储索引获得快速的查询响应。
- 使用 Power Query 轻松地发现、访问和组合各种数据源。
- 使用 Power Map 创建强大的地理空间可视化。
- 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 得到可信且一致的数据。
- 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

SQL Server 2016 在数据平台解决方案中有很多改进，这为用户带来了一种新的性能和可扩展性的突破，同时允许最终用户比以往任何时候更加快速、方便地探索和分析数据。新的功能(如内存 OLTP)为应用程序提供了巨大的性能优势，而不需要对应用程序的架构做任何改动。其他一些功能(如 AlwaysOn 可用性组)允许快速方便地扩展数据库应用程序。

若希望知道如何充分利用所有这些特性和功能，理解 SQL Server 的基本体系结构就十分重要。本节将介绍 SQL Server 2016 中的主要文件类型、文件管理、SQL Client 和系统数据库，还将概述架构、同义词和动态管理对象。最后，本节还将讲解数据类型、版本和许可选项。

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与以前的版本相比没什么改变。其取决于具体的类型。数据库文件有两个主要的功能，即保存数据、索引和数据库内的其他数据支持结构。事务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 区段的一部分。可通过 Transact-SQL 命令 create/alter index 的填充系数选项来指定每个数据页的填充情况。在 SQL Server 2016 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 8060B。但有一些例外，比如 text、ntext、image、varchar(max)、varbinary(max)以及定义数据表的一个 nvarchar(max)类型的数据列时，其存储数据的最大空间可达 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 8060B。不过，每个变长列达到 8KB，因此行的总尺寸大于 8060B。如果实际行尺寸超出 8060B，则会导致性能降级，因为现在逻辑行必须跨多个 8060B 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 中的数据库进行更改时，数据会写入日志，然后需要更改的页会被加载到内存中。之后更改将被写入这些页，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。这就是用户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)。

1.3.2 SQL Server Native Client

SQL Server Native Client 是 SQL Server 2005 自带的一种数据访问方法，在 SQL Server 2012 中得到了增强。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 的以下功能：

- 数据库镜像
- AlwaysOn 可读辅助路由
- 多活动结果集(Multiple Active Result Set，MARS)
- LocalDB 的原生客户端支持
- FileStream 支持
- 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
- 客户端连接中的服务主体名称(Service Principal Name，SPN)支持

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 Server 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)。

SQL Server 2016 安装的实际上是 SQL Server 2012 Native Client，因为并没有 SQL Server 2016 Native Client。其他重要的改变包括 SQL Server Native Client 中已不推荐使用 ODBC 驱动程序。因此，SQL Server Native Client 中的 ODBC 驱动程序不会再更新。不过，它有一个后继者，称为 Microsoft ODBC Driver 11 for SQL Server on Windows，SQL Server 2016 默认会安装该驱动程序。

1.3.3 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。

注意：

如果某些系统数据库被篡改或破坏，那么 SQL Server 将无法启动。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 中添加了 Resource 数据库。这个数据库包含 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\MSSQL14.MSSQLSERVER\MSSQL\Binn，每个实例只有一个 Resource 数据库。

注意：

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，那么需要改变此路径来进行匹配。另外，.MSSQLSERVER 是实例名。如果实例名也与此不同，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。自 SQL Server 2012 以来，升级到新服务补丁时，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)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 来连接该数据库。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语句，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;
```

注意：

不要将 Resource 数据库放在加密或压缩的驱动器中，这样做会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。通过运行下列查询语句(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。

注意：

尽量不要在 master 数据库中创建对象，如果要在其中创建对象，那么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，该数据库将被重新创建，其大小为其原始大小。由于该数据库每次都会重新创建，因

此不必备份它。对 tempdb 数据库中的对象进行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，用户可能收到以下错误信息之一：

- 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 都会将 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含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 数据库中添加自己的对象，那么应该把 model 数据库包含在备份中，或是应该维护所更改的脚本。

5. msdb 数据库

msdb 是系统数据库，它包含 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信息。该数据库存储了有关作业、操作员、警报、策略以及作业历史的全部信息。因为包含了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。分组的目的是便于管理，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进行适当的访问。

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，其中的 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 2016 的 AdventureWorks 示例数据库中(所有的 SQL Server 2016 示例都必须从 https://microsoft.con/zn-cn/download 网站单独下载和安装)。例如，使用两部分名称的示例查询如下所示：

```
USE AdventureWorks2016
GO
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，其实现方式在维护方面存在一些问题。如果拥有对象的用户要离开公司，就不能从 SQL Server 中删除该用户的登录，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相分离，从 SQL Server 2005 到 SQL Server 2016 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。通过这个抽象层可以改变一些物理实现，并将这些更改与使用者隔离开。下面的示例与连接服务器有关。表在另一个服务器上，该表需要连接到本地服务器上的表。该示例使用 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，若为 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，使用不同的连接服务器名，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进行任何更改。

注意：

同义词不能引用另一个同义词。object_id 函数返回同义词的 id 而非相关基本对象的 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 是接收参数的函数。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('AdventureWorks2016'),
FILE_ID('AdventureWorks_Data'));
```

SQL Server 2012 中提供了许多新的 DMV 和 DMF，利用它们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

SQL Server 2016 引入了下列新的 DMV 和 DMF：

- 与群集共享卷(Cluster Shared Volume，CSV)有关的动态管理视图和函数
- 与缓冲池扩展有关的动态管理视图和函数
- 与内存 OLTP 有关的动态管理视图和函数

1.3.7 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，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 Transact-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);
```

下面假设要为 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方法是连接 sys.objects 表和 sys.columns 表。在下面的代码中，有两个函数比较陌生。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()，它返回架构的标识值。在需要编写有关 SQL Server 元数据的报表时，该函数特别有用。

```
USE AdventureWorks2016;
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');
```

上述代码返回如下结果(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 和 text 等，用于存储字符数据。varchar 和 char 类型的主要区别在于数据填充。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 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 字节。SQL 将插入拖尾空格来填满这 20 个字符。

注意：

你可能想知道如果要节省空间，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，如果要存储两个字母形式的州名缩写，那么最好使用

char(2)。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。

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 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 2 字节。如果要将值 Brian 存储到 nvarchar 列，将使用 10 字节；而如果将之存储为 nchar(20)，就需要使用 40 字节。由于这些额外开销和空间的增加，因此应该避免使用 Unicode 列，除非确实有需要使用它们的业务或语言需求。应该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 列出了这些类型及其简单描述，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	n 字节
nchar(n)	n 为 1~4000 Unicode 字符	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	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、smallmoney 和 money。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 字节的空间。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/字节
bit	0、1 或 null	1
tinyint	0~255 的整数	1
smallint	−32 768~32 767 的整数	2
int	−2 147 483 648~2 147 483 647 的整数	4
bigint	−9 223 372 036 854 775 808~9 223 372 036 854 775 807 的整数	8
numeric(p,s)或 decimal(p,s)	$-10^{38}+1$~$10^{38}-1$ 的数值	最多 17
money	−922 337 203 685 477.580 8~922 337 203 685 477.580 7	8
smallmoney	−214 748.364 8~2 14 748.364 7	4

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可定义为 numeric(7,5)或 decimal(7,5)。如果将14.25插入 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 用于存储该数尾数的位数。SQL Server 对此只使用两个值。如果指定位数为 1~24，就使用 24。如果指定位数为 25~53，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及其简单描述，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/字节
float[(n)]	−1.79E+308~−2.23E−308,0,2.23E−308~1.79E+308	n≤8 n≥4
real( )	−3.40E+38~−1.18E−38,0,1.18E−38~3.40E+38	4

注意：

real 的同义词为 float(24)。

4. 二进制数据类型

varbinary、binary 和 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 引入了一种功能，可以在操作系统文件中通过 FileStream 存储选项来存储 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及其简单描述，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	n 字节
varbinary(n)	n 为 1~8000 十六进制数字	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 占用的存储空间为 4 字节，存储 1900 年 1 月 1 日—2079 年 6 月 6 日的时间，并且只精确到最近的分钟。datetime 数据类型占用的存储空间为 8 字节，存储 1753 年 1 月 1 日—9999 年 12 月 31 日的时间，并且精确到最近的 3.33 毫秒。

SQL Server 2012 引入了 4 种与日期相关的新数据类型：datetime2、datetimeoffset、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，表示的日期范围更广。时间总是用时、分、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 0~7，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 的有效值为 0~7。

表 1-5 列出了日期和时间数据类型及其简单描述，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/字节
date	1 年 1 月 1 日—9999 年 12 月 31 日	3
datetime	1753 年 1 月 1 日—9999 年 12 月 31 日，精确到最近的 3.33 毫秒	8
datetime2(n)	1 年 1 月 1 日—9999 年 12 月 31 日 范围为 0~7 的 n 指定小数秒	6~8
datetimeoffset(n)	1 年 1 月 1 日—9999 年 12 月 31 日 范围为 0~7 的 n 指定小数秒+/－偏移量	8~10
smalldatetime	1900 年 1 月 1 日—2079 年 6 月 6 日，精确到 1 分钟	4
time(n)	时:分:秒.9999999 范围为 0~7 的 n 指定小数秒	3~5

6.系统的其他数据类型

还有一些之前未见过的系统的其他数据类型，表 1-6 列出了这些数据类型。

表 1-6　系统的其他数据类型

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
SQL_Variant	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 值可以通过 newsequentialid()函数获得。这个函数返回的值对所有计算机来说都是唯一的。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

注意：

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的是 UTF-16 还是 UTF-8 来决定，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 2016 中，还可使用公共语言运行时(Common Language Runtime，CLR)中称为 SQL CLR 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 CLR 语言中的基本的类结构。

1.4 SQL Server 版本

SQL Server 2016 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的不同而不同。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 2016 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。更多信息可访问网址 www.microsoft.com/sqlserver。

如表 1-7 所示，SQL Server 2016 提供了 3 个主要的 SKU：

- 企业版包含 SQL Server 2016 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 BI 版的全部功能。
- BI (商业智能)版提供了 SQL Server 2016 中全套强大的 BI 功能，包括 Power Pivot、Power View、Data Quality Services 和 Master Data Services。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而设计的，提供了基本的数据库功能和基本的 BI 功能。SQL Server 2016 标准版中还新增了一些功能，如压缩。

表 1-7　SQL Server 2016 的版本

企业版(包含 BI 版的功能)	BI 版(包含标准版的功能)	标　准　版
关键任务和一级应用	公司和可扩展的分析报告	部门数据库
数据仓库	Power View 和 Power Pivot 允许自助分析	有限的商业智能项目
私有云和高度虚拟化的环境		
大型的、集中的或面向外部的商业智能		

注：Web 版本只能通过 Service Provider License Agreement(SPLA)使用。

表 1-8 总结了 SQL Server 2016 的各个版本的功能。

表 1-8　SQL Server 2016 各个版本的功能

功能	版本		
	标准版	BI 版	企业版
最大核数	16 核	数据库最多可以使用 16 核；对于 BI 功能，则为操作系统支持的最大核数	操作系统支持的最大核数
支持的内存	128GB	128GB	操作系统支持的最大空间
基本 OLTP	×	×	×
内存 OLTP			×
基本报表和分析	×	×	×
编程工具和开发人员的工具	×	×	×
可管理性(SSMS，基于策略的管理)	×	×	×
企业数据管理(数据质量，主数据服务)		×	×
自助式 BI(Power View，Power Pivot for SPS)		×	×
公司 BI(语义模型，高级分析)		×	×
高级安全性(高级审核，透明数据加密)			×
数据仓库(列存储索引，压缩，分区)			×
列存储索引			×
AlwaysOn			×
缓冲池扩展	×	×	×
虚拟许可	1 个虚拟机	1 个虚拟机	无限制

标准版中提供的是一些基本功能。BI SKU 中提供的是关键的企业 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，企业版是最佳选择。

从表 1-8 中可以看到，标准版对于数据库引擎最多只能支持 16 核。对于 BI 版，数据库最多可以使用 16 核，BI 功能(Analysis Services，Reporting Services)最多可以使用操作系统支持的最大核数。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 2016 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，可以实现最佳的可扩展性和高可用性。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，并使用这些许可将数据库部署到云端。

1.4.2　许可

SQL Server 2012 的许可模式发生了很大的变化。如果使用的不是通过 Microsoft Software Assurance 获取的许可，这种变化会对环境产生影响。本节只是概述了这些变化，并没有深入探

讨。更多相关细节可以咨询 Microsoft 客户经理。

SQL Server 2016 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，这为客户提供了多种好处，包括：

- SQL Server 2016 提供了高级的功能和支持。
  - 在主要供应商中，SQL Server 2016 仍然是明显值得信赖的引领者。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 2016，这可以节省大量成本。
- SQL Server 2016 针对云做了优化。
  - SQL Server 2016 是对虚拟化支持程度最高的数据库，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。
  - 完全支持本地的和基于云的混合解决方案。支持 SQL Server Data Files in Windows Azure、SQL Server Managed Backup to Windows Azure、从本地实例直接把数据库部署到 Windows Azure 虚拟机中的实例、将数据库备份到 Windows Azure Blob 存储，以及把副本添加到 Windows Azure 虚拟机中的 AlwaysOn 可用性组中。
- SQL Server 2016 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 的需求。
  - 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。

1. 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。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 2 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 2 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 2 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 2 个插槽，每个插槽 8 个核，需要 8 个许可包(16 个核的许可)。

2. 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 2016 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 2016 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。SQL Server 2016 还提供了一些令人兴奋的新功能，不仅允许扩展系统，还带来了前所未有的性能提升。将注意力集中到自己当前和将来所扮演的角色，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。

1.6 经典习题

1. SQL Server 2016 的常用版本有哪些？应用范围分别是什么？
2. SQL Server 2016 的优势是什么？
3. SQL Server 2016 由哪几个服务组成？