

# 第 3 章



## SQL Server 2016 数据库基础

2016年7月1日,微软公司发布了 SQL Server 数据库软件家族中最重要的一代产品,命名为 SQL Server 2016。SQL Server 2016 作为一款面向企业级应用的分布式关系数据库产品,在各行各业和各软件产品中得到了广泛的应用。SQL Server 2016 不仅延续了现有数据平台的强大能力,而且全面支持云技术。其提供了一个全面、灵活、可扩展的数据仓库管理平台,可以满足成千上万用户的海量数据管理需求,能够快速构建相应的解决方案,实现私有云与公有云之间数据的扩展与应用的迁移。SQL Server 2016 具有更安全、高性能、高级分析和可视化等强大功能,首次提供 R 语言与强大的商业智能 (Power BI) 功能,可用 R 语言打造智能应用程序,挖掘业务新价值并加以深入分析,不用再将数据发送到云端,即可用 Power BI 进行视觉化分析,并分享到移动设备上,协助用户在移动端、云端、社交网络与大数据四大趋势中快速掌握资料并进行及时的分析与决策,可以支持企业、部门以及个人等各种用户完成信息系统、电子商务、决策支持、商业智能等工作。本章将对 SQL Server 2016 系统进行概述,以使读者对该系统有整体的认识 and 了解,对 SQL Server 2016 系统在易用性、可用性、可管理性、可编程性、动态开发、安全性等方面有一个初步的理解,为后面各章的深入学习奠定坚实的基础。

### 3.1 SQL Server 2016 简介

#### 3.1.1 SQL Server 的发展历程

SQL Server 是一种广泛应用于网络业务数据处理的关系数据库管理系统,历经 20 多年发展到了今天的产品。表 3-1 概述了 SQL Server 的发展历程。

表 3-1 SQL Server 的发展历程

年份	发布版本	代码名称	说 明
1987	Sybase SQL Server	—	由 Sybase 公司发布
1989	SQL Server 1.0	Filipi	Microsoft 公司、Aston-Tate 公司参加到了 Sybase 公司的 SQL Server 系统开发中,只能运行于 OS/2 操作系统上的 16 位应用程序
1991	SQL Server 1.1	Pietro	只能运行于 OS/2 操作系统上的 16 位应用程序
1993	SQL Server 4.21a	SQLNT	由 Microsoft 公司和 Sybase 公司共同开发的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 NT 集成,界面易于使用并广受人们欢迎,标志着 Microsoft SQL Server 的真正诞生
1994			Microsoft 公司与 Sybase 公司终止合作关系
1995	SQL Server 6.0	SQL95	一种小型商业数据库,对核心数据库引擎做了重大的改写,性能得以提升,重要的特性得到增强
1996	SQL Server 6.5	Hydra	SQL Server 性能进一步改进
1998	SQL Server 7.0	Sphinx	一种 Web 数据库,再一次对核心数据库引擎进行了重大改写。该数据库介于基本的桌面数据库(例如 Microsoft Access)与高端企业级数据库(例如 Oracle 和 DB2)之间,为中小型企业提供了切实可行的可选方案
2000	SQL Server 2000	Shiloh	一种企业级数据库,SQL Server 在可扩展性和可靠性上有了很大的改进,其凭借优秀的数据处理能力和简单易用的操作,跻身世界三大企业级数据库(其他两大数据库为 Oracle 和 IBM DB2)之列,它提供了日志传送和索引视图等功能
2003	SQL Server 2000 64 位版	Liberty	
2005	SQL Server 2005	Yukon	对 SQL Server 的许多地方进行了改写,引入了 .NET Framework,并与 Microsoft Visual Studio 进行了集成。它是一个全面的数据库平台,不仅是大规模联机事务处理、数据仓库和电子商务应用的数据库平台,也是用于数据集成、分析和报表解决方案的商业智能平台。其提供了分区、数据库镜像、联机索引、数据库快照、复制、故障转移群集等功能
2008	SQL Server 2008	Katmai	该系统在可用性方面对数据库镜像进行了增强,可以创建热备用服务器,提供快速故障转移且保证已提交的事务不会丢失数据。在易管理性方面,增加了 SQL Server 审核功能,可以对各种服务器和数据库对象进行审核。在可编程性方面,增强的功能包括新数据存储功能、新数据类型、新全文搜索体系结构、对 Transact-SQL 所做的改进和增强
2010	Azure SQL Database	CloudDB	可以将本地数据库的数据和日志文件迁移和存储到 Azure 上
2010	SQL Server 2008 R2	Kilimanjaro (aka KJ)	新增数据中心版,最大支持 256 核

续表

年份	发布版本	代码名称	说 明
2012	SQL Server 2012	Denali	在管理、安全以及多维数据分析、报表分析等方面有进一步的提升,提供 AlwaysOn、列存储索引(支持高度聚合数据仓库查询,但不能使用集群,也不能更新)、增强的审计功能、大数据支持等功能
2014	SQL Server 2014	Hekaton	为用户的关键任务应用程序提供突破性的性能、可用性和可管理性,主要包括内置内存技术、混合云解决方案、备份加密、AlwaysOn 增强功能、延迟持续性、分区切换和索引生成、列存储索引、利用 SSDE 对高使用频率数据进行缓冲池扩展、增量统计信息等
2016	SQL Server 2016	—	通过 SQL Server 2016,可以使用可缩放的混合数据库平台生成任务关键型智能应用程序。此平台内置了需要的所有功能,包括内存中性能、高级安全性和数据库内分析。SQL Server 2016 版本新增了安全功能、查询功能、Hadoop 和云集成、R 分析等功能,以及许多改进和增强功能
2017	SQL Server 2017	—	SQL Server 2017 跨出了重要的一步,它力求通过将 SQL Server 的强大功能引入 Linux、基于 Linux 的 Docker 容器和 Windows,使用户可以在 SQL Server 平台上选择开发语言、数据类型、本地开发或云端开发,以及操作系统开发。其提供了可恢复的在线索引重建、图表数据库功能,以及用于多对多关系建模、R/Python 机器学习方面的功能等
2018	SQL Server 2018	Hekaton	SQL Server 2018 中最吸引人关注的特性就是内存在线事务处理(OLTP)引擎,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。它引入了另一种列存储索引,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。微软一直将 SQL Server 2018 定位为混合云平台,这意味着 SQL Server 数据库更容易整合 Windows Azure

### 3.1.2 SQL Server 2016 的主要功能

SQL Server 2016 拥有各种新的特性和增强,提供了突破性的性能、高级的安全性,以及更丰富、集成的报表和分析功能。SQL Server 2016 使用新的快速发布模型构建,包含了许多 Microsoft Azure SQL 数据库中最先在云中引入的特性。此外,SQL Server 2016 还包含将历史数据动态迁移到云的功能。SQL Server 2016 的新功能主要如下:

(1) 伸缩数据库(Stretch Database)。为了增强其性能,可将数据动态延伸至云计算平台与服务 Azure,以便于及时查看且始终处于安全保护。

(2) 实时业务分析与加速数据处理。其主要借助实时业务分析与内存中的联机事务处理(OLTP)功能提供实时数据分析并加速处理数据。

(3) 支持全程加密技术(Always Encrypted)。对数据的全程加密使得加密更便捷,数据的存储和应用都采用微软技术进行加密,确保数据库中的数据都进行加密保护,且无

须对应用重写。

(4) 增强的安全功能。层级安全性管控使客户基于用户特征控制数据访问,动态数据屏蔽(Dynamic Data Masking)保护数据。

(5) 提升可用性及灾难可恢复性。实现 AlwaysOn 高可用性和故障可恢复性,改进同步复制、事务处理协调器支持和负载均衡等。

(6) 更快的混合型 hybrid 备份。增强的云平台 Azure 混合备份功能,在 Azure 虚拟机中也可实现更快的备份和恢复。用户可将本地数据库中的数据和日志文件迁移和存储到 Azure 上。

(7) 为多种类型数据提供更好的支持。利用数据交换格式(JavaScript Object Notation,JSON)对数据的支持,可实现快捷解析和存储。

(8) 可用性和可扩展性得到较大提升。利用企业实时通信工具和分析服务等使性能得到提高,并强化信息管理。

(9) 内置高级分析(Built-in Advanced Analytics)、混合基(Poly Base)和移动商业智能(Mobile BI)。

(10) 多 TempDB 数据库文件。多个 TempDB 数据文件可在多核计算机中运行。

### 3.1.3 SQL Server 2016 的各版本和支持的功能

SQL Server 2016 分为 SQL Server 2016 企业版(Enterprise)、标准版(Standard)、Web 版、开发者版(Developer)和速成版(Express),它们的功能、作用和价格各不相同,如表 3-2 所示。其中,开发者版和速成版可免费下载。

表 3-2 SQL Server 2016 各版本的比较

版本分类	功 能
企业版	提供了全面的高端数据中心功能,性能极为快捷,虚拟化不受限制,还具有端到端的商业智能,可为关键任务工作负荷提供较高服务级别,支持最终用户访问深层数据。
标准版	提供了基本数据管理和商业智能数据库,使部门和小型组织能够顺利运行其应用程序,并支持将常用开发工具用于内部部署和云部署,有助于以最少的 IT 资源获得高效的数据库管理
Web 版	对于为从小规模至大规模 Web 资产提供可伸缩性、经济性和可管理性功能的 Web 宿主和 Web VAP 来说,SQL Server Web 版本是一项总拥有成本较低的选择
开发者版	支持开发人员基于 SQL Server 构建任意类型的应用程序。它包括企业版的所有功能,但有许可限制,只能用作开发和测试系统,而不能用作生产服务器。SQL Server 开发者版是构建 SQL Server 和测试应用程序的人员的理想之选
速成版	入门级的免费数据库,是学习和构建桌面及小型服务器数据驱动应用程序的理想选择。它是独立软件供应商、开发人员和热衷于构建客户端应用程序的人员的最佳选择

从 SQL Server 2016 开始,仅提供 64 位版本。本书以 SQL Server 2016 开发者版进行介绍,其可以运行在 Windows Server 2016、Windows Server 2012、Windows 10、Windows 8.1、Windows 8 等操作系统中。

### 3.1.4 SQL Server 2016 的服务器组件

SQL Server 2016 是一个提供了联机事务处理、数据仓库、电子商务应用的数据库、数据分析平台和解决方案,它的服务器组件主要包括数据库引擎(Database Engine, DE)、分析服务(Analysis Services, AS)、集成服务(Integration Services, IS)、报表服务(Reporting Services, RS)、主数据服务(Master Data Services, MDS)以及 R 服务(R Services)组件等。SQL Server 2016 各组件的组成结构如图 3-1 所示,通过选择不同的服务器类型来完成不同的数据库操作。本书仅讲解数据库引擎的相关技术。

用于操作、管理和控制的数据库引擎是整个系统的核心组件,其他所有组件都与其有着密不可分的联系。SQL Server 2016 主要组件之间的关系如图 3-2 所示。



图 3-1 SQL Server 2016 的服务器组件



图 3-2 SQL Server 2016 主要组件之间的关系

SQL Server 2016 的服务器组件及其对应的功能如下:

(1) 数据库引擎。数据库引擎是 SQL Server 2016 的核心组件,负责完成业务数据的操作、管理和控制等操作。其包括数据库引擎(用于存储、处理和保护数据、复制及全文搜索的核心服务)、用于管理关系数据和 XML 数据的工具,以及数据质量服务(Data Quality Services, DQS)的服务器。例如,创建数据库、创建表、创建视图、数据查询、访问数据库等操作都是由数据库引擎完成的。在通常情况下,使用 SQL Server 2016 实际上就是在使用数据库引擎。

(2) 分析服务。分析服务为商业智能提供了联机分析处理(Online Analytical Processing, OLAP)和数据挖掘功能,可以支持用户建立数据仓库和商业智能分析。使用分析服务,用户可以设计、创建和管理包含来自其他数据源数据的多维结构,通过对多维数据进行多角度分析,可以使管理人员对业务数据有更全面的理解。另外,通过使用分析服务,用户可以完成数据挖掘模型的构造和应用,实现知识发现、表示和管理。

(3) 集成服务。集成服务是一组图形工具和可编程对象,用于生成企业级数据集成和数据转换解决方案的平台,可以完成有关数据的提取、转换、加载等操作。如图 3-2 所示,数据库引擎、分析服务和报表服务就是通过集成服务来进行联系的。

(4) 报表服务。报表服务包括用于创建、管理和部署表格报表、矩阵报表、图形报表以及自由格式报表的服务器和客户端组件,可为用户提供支持 Web 的企业级报表功能。

报表服务还是一个可用于开发报表应用程序的可扩展平台。通过使用 SQL Server 2016 系统提供的报表服务,用户可以方便地定义和发布满足自己需求的报表。无论是报表的布局格式,还是报表的数据源,用户都可以轻松地实现。这种服务极大地便利了企业的管理工作,满足了管理人员高效、规范的管理需求。

(5) 主数据服务。主数据服务是针对主数据管理的 SQL Server 解决方案。用户可以配置主数据服务来管理任何领域(产品、客户、账户);在主数据服务中可包括层次结构、各种级别的安全性、事务、数据版本控制和业务规则,以及可用来管理数据的用于 Excel 的外接程序。其包括复制服务、服务代理、通知服务和全文检索服务等功能组件,共同构成完整的服务架构。

(6) R 服务。R 服务(数据库内)支持在多个平台上使用可缩放的分布式 R 解决方案,并支持使用多个企业数据源(例如 Linux、Hadoop 和 Teradata 等)。

### 3.1.5 SQL Server 2016 的主要管理工具

在实际应用中经常使用 SQL Server 2016 的管理工具,用户借助这些管理工具可以对系统进行快速、高效的管理。下面分别简要介绍。

(1) **SQL Server 管理平台(SQL Server Management Studio, SSMS)**。SSMS 是一个图形化的集成环境,它将查询设计器和服务管理器的各种功能组合到一个集成环境中,用于访问、配置、管理和开发 SQL Server 的相关工作。在 SSMS 中包含了大量的图形工具和丰富的脚本编辑器,使各种技术水平的开发人员和管理员都能使用 SQL Server。

(2) **SQL Server 数据工具(SQL Server Data Tools)**。SQL Server 2016 以前的版本将其称为商业智能开发工具(Business Intelligence Development Studio),其提供集成开发环境(Integrated Development Environment, IDE)以便为分析服务、报表服务和集成服务智能组件生成解决方案,该工具包括了一些项目模板。SQL Server 数据工具还包含“数据库项目”,为数据库开发人员提供集成环境,以便在 Visual Studio 内为任何 SQL Server 平台(包括本地和外部)执行其所有数据库设计工作。数据库开发人员可以使用 Visual Studio 中功能增强的服务器资源管理器,轻松创建或编辑数据库对象或执行查询。

(3) **SQL Server 配置管理器(SQL Server Configuration Manager)**。它用于管理与 SQL Server 相关的服务、配置 SQL Server 使用的网络协议,以及从 SQL Server 客户端计算机管理网络连接。在 SQL Server 配置管理器中集成了服务器网络使用工具、客户端网络使用工具和服务器管理等功能。

(4) **SQL Server 事件探查器(SQL Server Profiler)**。SQL Server 事件探查器提供了一个图形用户界面,用于监督、记录和检查数据库服务器的使用情况。使用该工具,管理员可以实时监控用户的活动状态。

(5) **数据库引擎优化顾问(Database Engine Tuning Advisor)**。数据库引擎优化顾问用来帮助用户分析工作负荷、提出优化建议等。即使用户对数据库的结构没有详细的了解,也可以使用该工具选择和创建最佳的索引、索引视图和分区等。

(6) **数据质量客户端**。SQL Server 2016 提供了一个非常简单和直观的图形用户界面,用于连接到 DQS 数据库并执行数据清理操作。它还允许用户集中监视在数据清理操

作过程中执行的各项活动。

(7) **连接组件**。安装用于客户端和服务端之间通信的组件,以及用于 DB-Library、ODBC 和 OLE DB 的网络库。

本书将主要围绕 SQL Server Management Studio 展开介绍。尽管本书重点介绍 SQL Server Management Studio 中的数据库引擎服务,但有了这方面的知识后,可以很容易地学习其他的服

## 3.2 SQL Server 2016 的登录

在 SQL Server 2016 中,一个 SQL Server 服务器又称为一个数据库实例。在同一台计算机上可以运行多个 SQL Server 2016 服务器,也就是多个数据库实例,简称“实例”。用“计算机名/实例名”来区分不同的命名实例。但一台计算机上只允许有一个默认实例,默认实例用“计算机名”表示。每个实例都提供了数据库引擎、分析服务、报表服务以及集成服务等。在一般情况下,要完成 SQL Server 的基本操作,例如创建和维护数据库,必须要启动数据库引擎。

### 3.2.1 启动数据库引擎

为了能有效控制用户对服务器资源的访问,需要对数据库引擎进行启动、停止、暂停和退出操作,可以使用 SQL Server 2016 配置管理器完成该项工作。

当完成 SQL Server 2016 相应版本的安装后,单击“开始”按钮,选择“所有程序”→Microsoft SQL Server 2016→“配置工具”→“SQL Server 2016 配置管理器”命令,即可打开 Sql Server Configuration Manager 对话框,如图 3-3 所示。

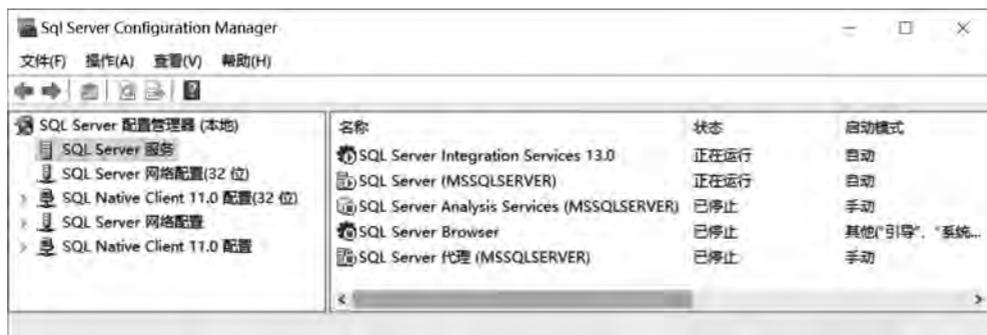


图 3-3 SQL Server Configuration Manager 对话框

在图 3-3 中,单击左侧的“SQL Server 服务”,在右侧显示该服务器的所有服务,例如 SQL Server(MSSQLSERVER),其中小括号里面的 MSSQLSERVER 是一个数据库命名实例。该实例是在安装 SQL Server 2016 的过程中选择的默认实例名称,用户在安装时可以进行修改。用户可以右击某一个服务,例如 SQL Server(MSSQLSERVER),在弹出的下拉菜单中选择“启动”“暂停”“停止”命令实现服务器的启动、暂停、停止操作。暂停服务是指不允许新的用户继续登录服务器,但是已经登录的用户依然可以不受影响地工作。

停止服务是指从内存中清除 SQL Server 2016 的所有服务器进程,除了不允许新的用户继续登录服务器外,已连接用户的操作也会被禁止。

### 3.2.2 SQL Server Management Studio 的启动与连接

SQL Server (MSSQLSERVER) 服务启动后,即可启动 SQL Server Management Studio 并连接到 SQL Server 服务器。在 Windows 系统桌面中,单击“开始”按钮,选择“所有程序”→Microsoft SQL Server 2016→SQL Server Management Studio 命令,就可以打开如图 3-4 所示的“连接到服务器”对话框。



图 3-4 “连接到服务器”对话框

(1) 在“服务器类型(T)”下拉列表框中列出了 SQL Server 2016 的所有服务,因为是进行数据管理工作,所以选择“数据库引擎”选项。其他还包括分析服务、报表服务和集成服务选项。

(2) 在“服务器名称(S)”下拉列表框中列出了当前网络中安装 SQL Server 服务器的所有计算机名称,这里选择当前计算机名即可。当然,使用 SQL Server 服务器的 IP 地址也可以,有时利用“.”或“(local)”表示本地计算机。

(3) 当用户登录数据库系统时,如何确保只有合法的用户才能登录到系统中,是一个最基本的安全性问题,也是数据库管理系统提供的基本功能。在 Microsoft SQL Server 2016 系统中,通过身份验证模式解决这个问题,共有 4 种身份验证模式,即 Windows 身份验证、SQL Server 身份验证、活动目录(Active Directory)密码身份验证、活动目录集成身份验证。在 Windows 身份验证模式中,用户通过 Microsoft Windows 用户账户连接时,SQL Server 使用 Windows 操作系统中的信息验证账户名和密码。一般不建议用户使用该种身份验证模式,而应使用 SQL Server 身份验证模式。在采用 SQL Server 身份验证时,需要内置的 SQL Server 的系统管理员 sa 的密码。sa 是一个默认的 SQL Server 登录名,拥有操作 SQL Server 系统的所有权限,该登录名不能被删除。在采用 SQL Server 身份验证模式安装 Microsoft SQL Server 系统之后,应该为 sa 指定一个密码。此时在“密码”输入框中输入初次安装时的密码即可。

单击“连接”按钮,即可进入 SQL Server Management Studio 开始数据库之旅,如图 3-5 所示。

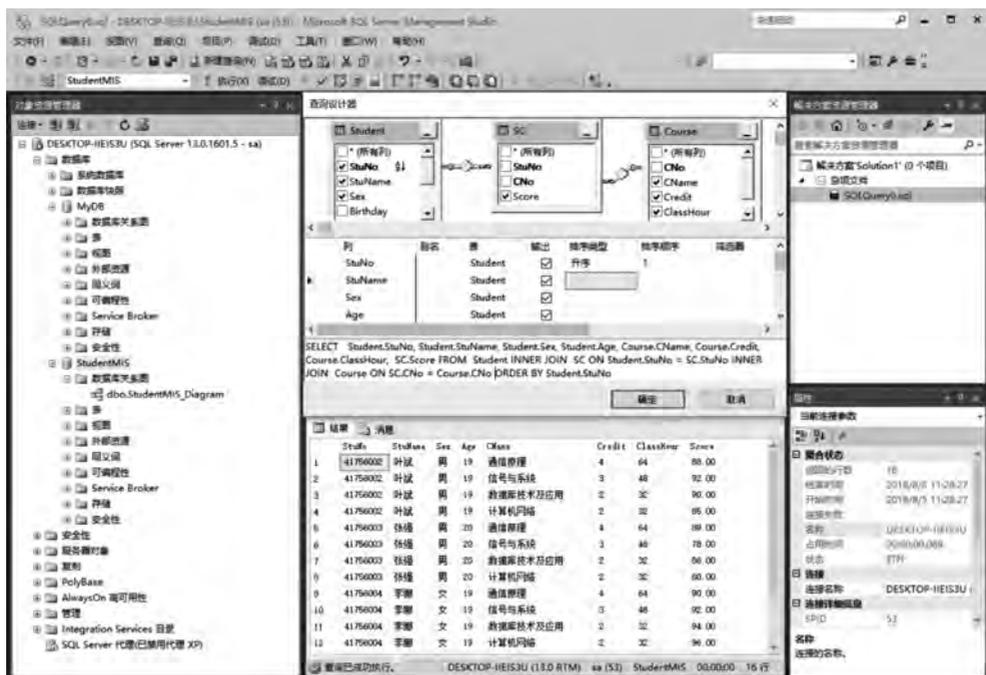


图 3-5 SQL Server Management Studio 界面

### 3.3 SQL Server Management Studio 简介

在 SQL Server 中,用于业务数据处理与管理有两种常用的操作方式:

(1) 通过 SQL Server Management Studio 的界面菜单方式进行操作。

(2) 用 SQL 语句及扩展的事务-结构化查询语言 (Transact-SQL) 进行操作。

Transact-SQL 是 SQL Server 的核心组件,在数据处理与管理等常用操作语句及其语法规则中极为重要,特别是在动态数据处理及系统运行中更为常用,将在第 4 章进行介绍。本节仅介绍 SQL Server Management Studio 的使用。

SQL Server Management Studio 是 SQL Server 2016 提供的一种新的集成环境,用于访问、配置、管理和开发 SQL Server 的所有组件。SQL Server Management Studio 组合了大量图形工具和丰富的脚本编辑器,极大地方便了技术人员和数据库管理员对 SQL Server 的各种操作。

SQL Server Management Studio 将 SQL Server 2005 以前版本所包含的企业管理器、查询分析器和 OLAP 分析管理器功能整合到单一的环境中。此外,SQL Server Management Studio 还可以和 SQL Server 的所有组件协同工作,例如报表服务、集成服务等。开发人员可以获得熟悉的体验,而数据库管理员可获得功能齐全的统一实用工具,其中包含易于使用的图形工具和多功能的脚本编辑器。

可以从程序组中启动 SQL Server Management Studio,启动该工具后的界面如图 3-5 所示。

SQL Server Management Studio 集成工作环境一般包括 4 个组件窗口,即对象资源管理器、查询编辑器、已注册的服务器、模板浏览器。

### 3.3.1 对象资源管理器

对象资源管理器是 SQL Server Management Studio 的一个组件,可连接到数据库引擎实例、分析服务、集成服务、报表服务。它提供了服务器中所有对象的视图,并具有可用于管理这些对象的用户界面。对象资源管理器的功能根据服务器的类型稍有不同,但一般都包括用于数据库的开发功能和用于所有服务器类型的管理功能。对象资源管理器与 SQL Server 2000 的企业管理器类似。该组件使用了类似于 Windows 资源管理器的树状结构,在左边的树结构图上,根结点是当前实例,子结点是该服务器的所有管理对象和可以执行的管理任务,分为“数据库”“安全性”“服务器对象”“复制”以及 PolyBase 和“AlwaysOn 高可用性”“管理”“集成服务目录”“SQL Server 代理”共九大类,如图 3-5 左边所示。

### 3.3.2 查询编辑器

在 SQL Server Management Studio 中,查询编辑器与 SQL Server 2000 的查询分析器类似,可以执行输入的 SQL 语句,执行结果会显示在屏幕下方。用户也可以使用图形化的方式进行数据库对象的拖拉操作,选择相应的显示字段,动态生成 SQL 语句,如图 3-5 中间所示。

在 SQL Server Management Studio 中,用户可输入 SQL 语句,执行语句并在“结果”窗口查看结果,如图 3-6 所示。用户也可以打开包含 SQL 语句的文本文件,执行语句并

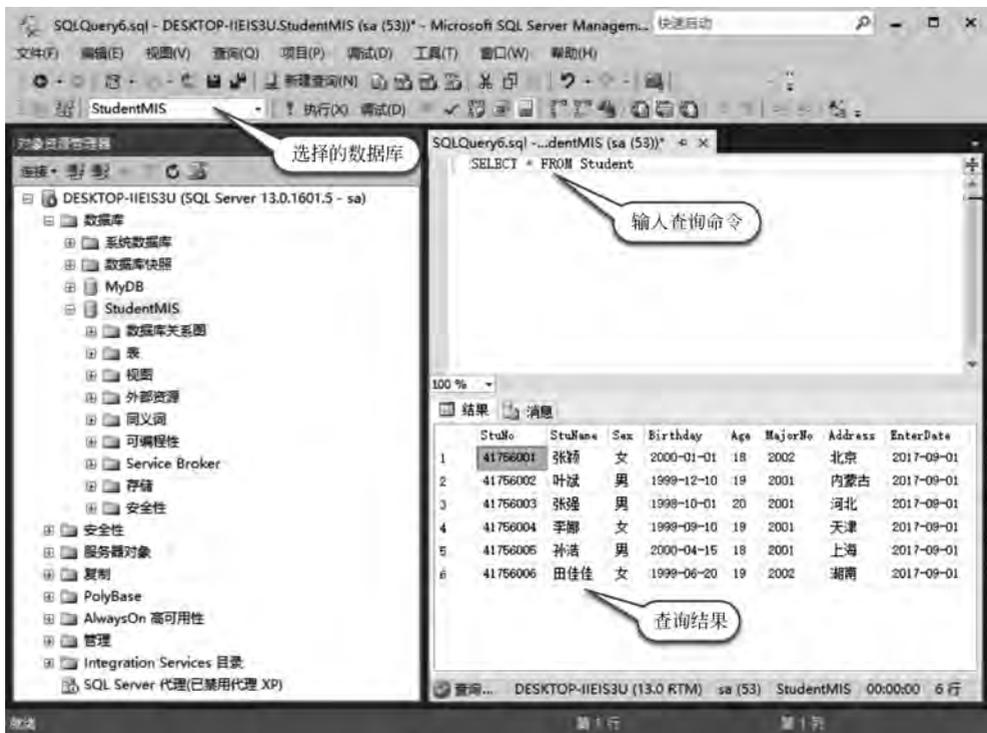


图 3-6 查询编辑器

在“结果”窗口中查看结果。SQL Server Management Studio 提供如下功能：用于输入 SQL 语句的自由格式文本编辑器；在 SQL 语句中使用不同的颜色，以提高复杂语句的易读性；对象浏览器和对象搜索工具，可以轻松查找数据库中的对象和对象结构；模板可用于加快创建 SQL Server 对象的 SQL 语句的开发速度；用于分析存储过程的交互式调试工具；以网格或自由格式文本窗口的形式显示结果；显示计划信息的图形关系图，用于说明内置在 SQL 语句执行计划中的逻辑步骤。

为了使文本消息和输出结果显示在同一窗口，需要设置输出结果的格式为“以文本格式显示结果”。步骤如下：进入 SQL Server Management Studio，选择“工具”菜单，然后选择“选项”命令，出现“选项”对话框，如图 3-7 所示，然后进行相应的设置。

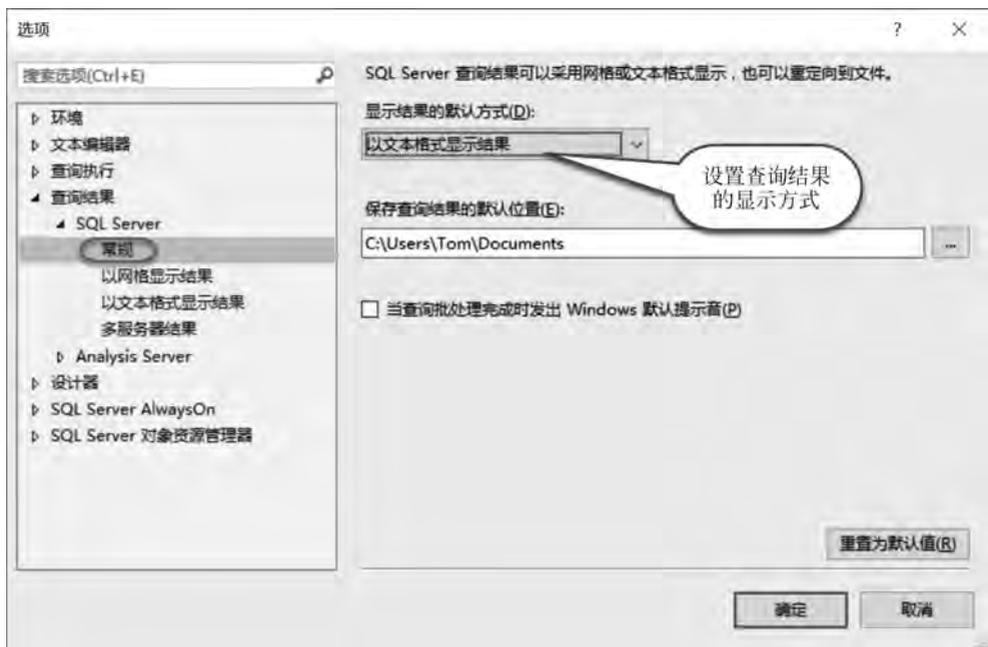


图 3-7 “选项”对话框

在设置输出结果的格式为“以文本格式显示结果”后，再次执行，界面如图 3-8 所示。

### 3.3.3 已注册的服务器

通过在 SQL Server Management Studio 的已注册的服务器组件中注册服务器，保存经常访问的服务器的连接信息。用户可以在连接前注册服务器，也可以在通过对象资源管理器进行连接时注册服务器。为了管理、配置和使用 SQL Server 2016 系统，必须使用 SQL Server Management Studio 工具注册服务器。注册服务器就是为 Microsoft SQL Server 客户机/服务器系统确定一个数据库所在的计算机，该计算机作为服务器可以为客户端的各种请求提供服务。如图 3-9 所示，可以看到注册的服务器名称是 desktop-iiis3u。

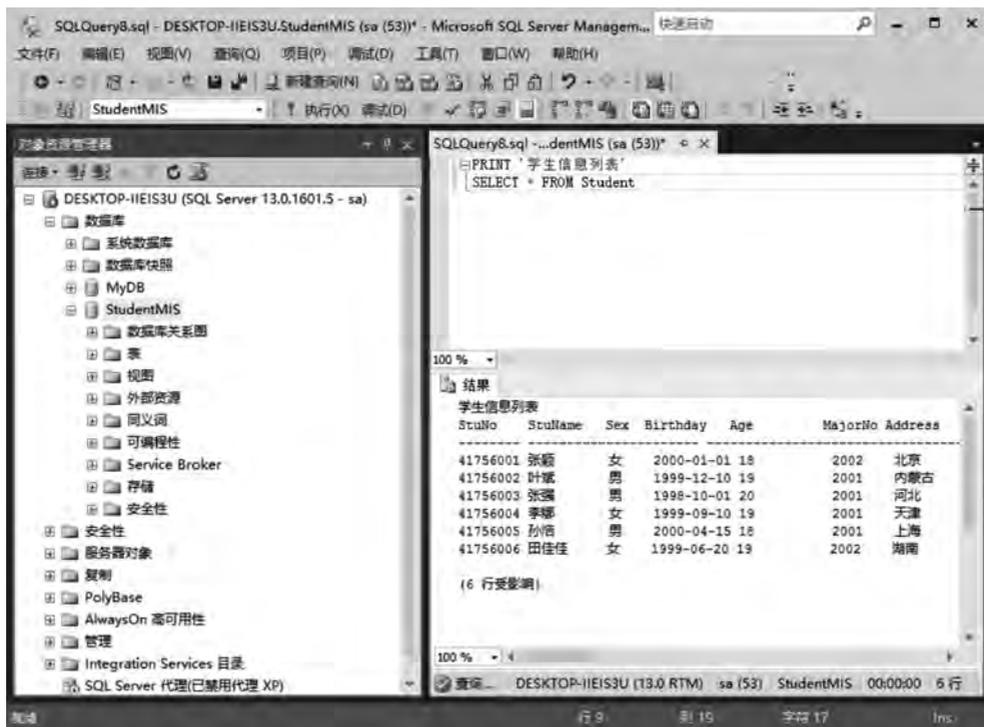


图 3-8 以文本格式显示查询结果

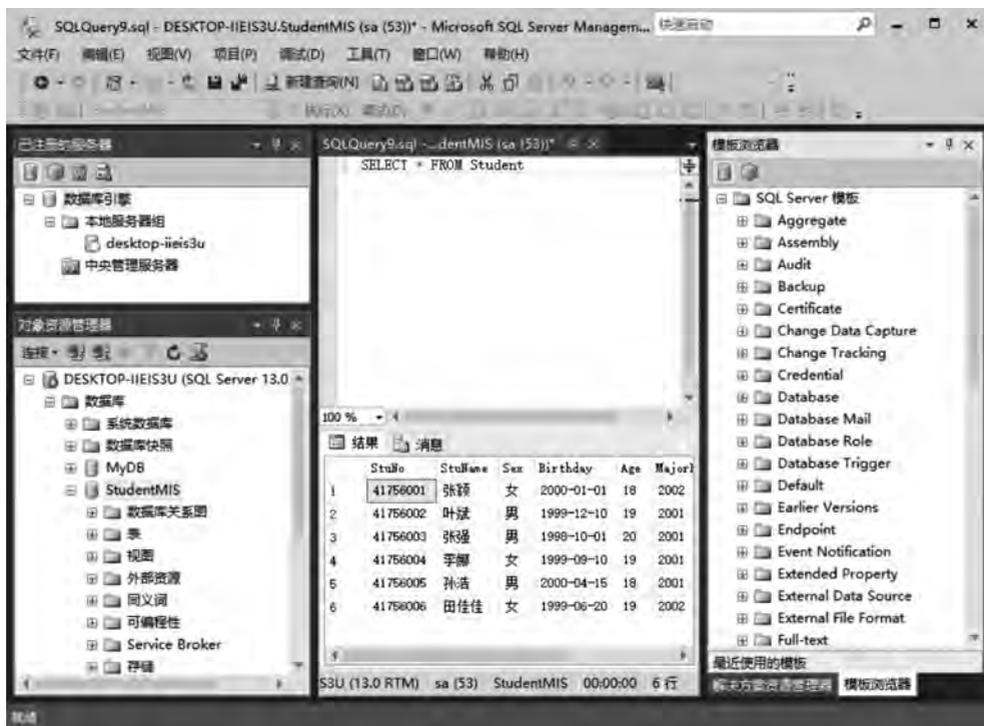


图 3-9 已注册的服务器

在本地计算机安装 SQL Server 2016 服务器后,第一次启动 SQL Server 2016 服务器时,SQL Server 2016 会自动完成本地数据库服务器的注册,但对于一台仅安装了 SQL Server 客户端的计算机而言,要访问 SQL Server 服务器的数据库资源,必须由用户来完成服务器的注册,注册服务器是进行服务器集中管理和实现分布式查询的前提。

启动 SQL Server Management Studio 工具,在“已注册的服务器”窗口中打开“数据库引擎”结点。右击“本地服务器组”结点,从弹出的快捷菜单中选择“新建服务器注册”命令,如图 3-10 所示。单击该命令,出现如图 3-11 所示的“新建服务器注册”对话框的“常规”选项卡。在“服务器名称”下拉列表框中既可以输入服务器名称,也可以选择一个服务器名称。从“身份验证”下拉列表框中可以选择身份验证模式,这里选择了“SQL Server 身份验证”。用户可以在“已注册的服务器名称”文本框中输入该服务器的显示名称。“连接属性”选项卡选择默认设置。在如图 3-11 所示的对话框中,单击“测试”按钮,可以对当前连接属性的设置进行测试。如果出现表示连接测试成功的信息框,那么当前连接属性的设置就是正确的。在完成连接属性的设置后,单击图 3-11 中的“保存”按钮,表示完成连接属性的设置操作。

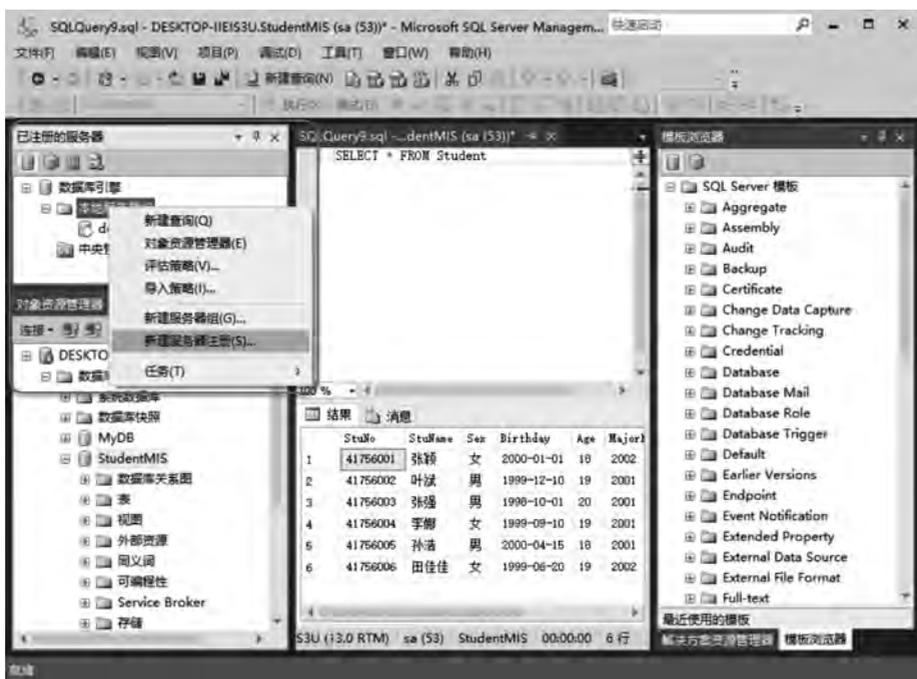


图 3-10 选择“新建服务器注册”命令

### 3.3.4 模板浏览器

SQL Server Management Studio 提供了大量包含用户提供的值(例如表名称)的参数脚本模板。使用该参数,可以只输入一次名称,然后自动将该名称复制到脚本中所有必要的位置。用户可以编写自己的自定义模板,以支持频繁编写的脚本。“模板浏览器”窗口如图 3-10 的右侧所示。



图 3-11 “新建服务器注册”对话框

### 3.4 SQL Server 2016 数据库的存储结构与分类

数据库是 SQL Server 中存储数据和数据库对象的容器。数据库对象是指存储、管理和使用数据的不同结构形式。

#### 3.4.1 SQL Server 2016 数据库的存储结构与文件

##### 1. SQL Server 2016 数据库的存储结构

SQL Server 2016 数据库的存储结构分为两种,即逻辑存储结构和物理存储结构。

(1) 数据库的逻辑存储结构。数据库的逻辑存储结构表示数据库中各数据之间的逻辑关系,说明数据库是由哪些性质的信息所组成。在如图 3-5 所示的 SQL Server Management Studio 集成工作环境左侧的“对象资源管理器”中可以看到 MyDB 数据库结点,包括数据库关系图、表、视图、外部资源、同义词、可编程性(存储过程、数据库触发器、函数等)、服务代理(Service Broker)、存储、安全性等若干数据库对象,这就是 SQL Server 数据库的逻辑结构。

(2) 数据库的物理存储结构。数据库的物理存储结构讨论数据库文件是如何在磁盘上存储的,数据库在磁盘上是以文件为单位存储的,由数据库文件和事务日志文件组成,一个数据库至少应该包含一个数据库文件和一个事务日志文件。例如,MyDB 数据库在物理存储上被映射成 4 个物理文件,文件名分别为 MyDB.mdf、MyDB\_log.ldf、MyDB\_secondary1.ndf、MyDB\_secondary2.ndf,如图 3-12 所示。

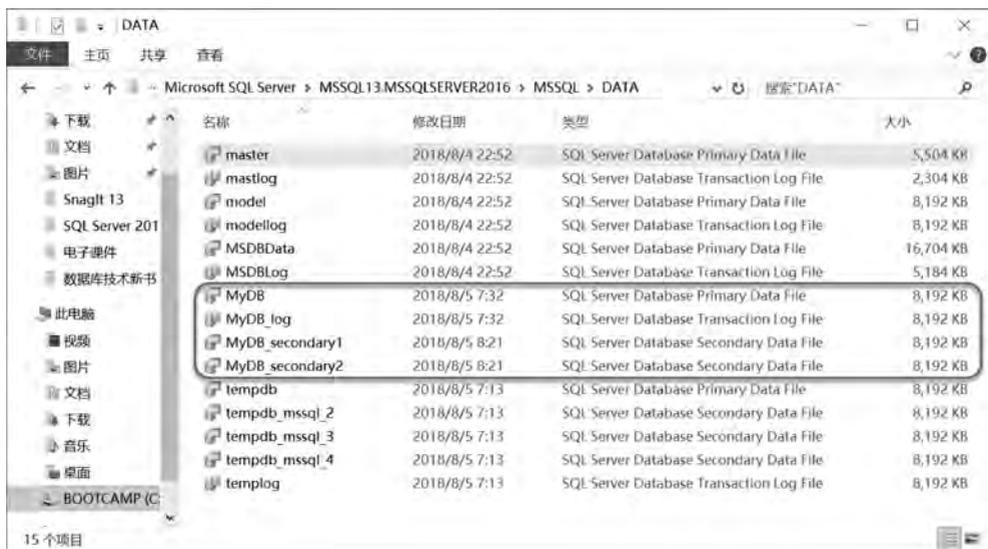


图 3-12 数据库的物理存储结构

## 2. SQL Server 2016 的数据库文件

SQL Server 的文件拥有两个名称,即逻辑文件名和物理文件名。当使用 Transact-SQL 命令语句访问某一个文件时,必须使用该文件的逻辑文件名。物理文件名是文件实际存储在磁盘上的文件名,而且可包含完整的磁盘目录路径。在 SQL Server 2016 中,数据库文件包括以下 3 种类型:

(1) 主数据文件。主数据文件是数据库的起点,包含数据库的启动信息,并指向数据库中的其他文件,是用来存放数据库部分或者全部数据和数据库对象的文件。每个数据库都有一个主数据文件,默认扩展名为 .mdf。

(2) 次要数据文件。次要数据文件是可选的,由用户定义并存储用户数据。除主数据文件以外的其他所有数据文件都是次要数据文件。如果主数据文件的大小超过了单个 Windows 文件的最大限制,可以使用次要数据文件继续增长。一个数据库可以没有次要数据文件,也可以同时拥有多个次要数据文件。次要数据文件的默认扩展名是 .ndf。采用主、次要数据文件来存储数据,容量可以无限制地扩充而不受操作系统文件的大小限制。用户可以将数据文件保存在不同的硬盘上,提高了数据处理的效率。

(3) 日志文件。日志文件用于记录所有事务以及每个事务对数据库所做的修改。日志文件包含着用于恢复数据库的所有日志信息。每个数据库必须至少有一个事务日志文件,当然也可以有多个。事务日志文件的默认扩展名是 .ldf。

SQL Server 2016 不强制使用 .mdf、.ndf 或者 .ldf 作为文件的扩展名,但建议使用这些扩展名帮助标识文件的用途。

## 3. SQL Server 2016 的数据库文件组

为了方便数据的管理和分配,可以将数据库对象和文件一起分成文件组。文件组就

是文件的逻辑集合。例如在某个数据库中,3个文件(data1.ndf、data2.ndf和data3.ndf)分别创建在3个不同的磁盘驱动器中,然后为它们指定一个文件组group1。以后,所创建的表可以明确指定放在文件组group1上。对该表中数据的查询将分布在这3个磁盘上,因此可以通过执行并行访问来提高查询性能。

在创建表时,不能指定将表放在某个文件上,只能指定将表放在某个文件组上。因此,如果希望将某个表放在特定的文件上,那么必须通过创建文件组来实现。通常有3种类型的文件组:

(1) 主文件组。主文件组(PRIMARY文件组)包含主数据文件和任何没有明确分配给其他文件组的其他文件。数据库的所有系统表都被分配到主文件组中。当主文件组的存储空间用完之后,将无法向系统表中添加新的目录信息,一个数据库有一个主文件组。

(2) 用户定义文件组。用户定义文件组是通过在CREATE DATABASE或ALTER DATABASE语句中使用FILEGROUP关键字指定的任何文件组,其目的在于数据分配,以提高数据表的读/写效率。

(3) 默认文件组。各数据库都有一个被指定的默认文件组,容纳所有在创建时没有指定文件组的表、索引以及text、ntext和image数据类型的数据。

日志文件不包括在文件组内。日志空间与数据空间分开管理。同时,一个文件不可能是多个文件组的成员,而且一个文件或文件组只由一个数据库使用。表、索引和大型对象数据可以与指定的文件组相关联。

### 3.4.2 SQL Server 2016 数据库的分类

SQL Server 2016 系统提供了3种类型的数据库,即系统数据库、用户数据库和示例数据库。

#### 1. 系统数据库

系统数据库存放SQL Server 2016的系统级信息,例如系统配置、数据库的属性、登录账户、数据库文件、数据库备份、警报、作业等信息。SQL Server 2016使用这些系统级信息管理和控制整个数据库服务器系统,如图3-13所示。SQL Server 2016在安装时创建了5种系统数据库,即master、model、msdb、resource和tempdb。

##### 1) master 数据库

master数据库是SQL Server最重要的系统数据库,是整个数据库服务器的核心。它记录了SQL Server系统级的所有信息,这些系统级的信息包括服务器配置信息、登录账户信息、数据库文件信息、SQL Server初始化信息



图 3-13 系统数据库示意图

等,这些信息影响整个 SQL Server 系统的运行。用户不能直接修改该数据库,永远也不要再在 master 数据库中创建对象,如果在其中创建对象,则可能需要更频繁地进行备份。如果损坏了 master 数据库,则 SQL Server 无法启动。master 数据库是 SQL Server 的默认数据库。用户使用 SQL Server Management Studio 登录后,新建的查询是针对 master 数据库的。用户可以在下拉列表中修改当前可用的数据库。

### 2) model 数据库

model 数据库是一个在 SQL Server 创建新数据库时充当模板的系统数据库。该数据库存储了可以作为模板的数据库对象和数据。当创建用户数据库时,系统自动把该模板数据库中的所有信息复制到用户新建的数据库中,使得新建的用户数据库初始状态下具有与 model 数据库一致的对象和相关数据,从而简化了数据库的初始创建和管理操作。对 model 数据库进行的修改(例如数据库大小、排序规则、恢复模式和其他数据库选项)将应用于以后创建的所有数据库。

### 3) msdb 数据库

msdb 数据库是代理服务数据库,包含 SQL Server 代理、日志传送、SQL Server 集成服务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、报警、任务调度以及作业历史的全部信息,这些信息可以用于自动化系统的操作。

### 4) resource 数据库

resource 数据库是一个只读数据库,包含了 SQL Server 2016 系统中的所有信息。

### 5) tempdb 数据库

tempdb 数据库是一个临时数据库,类似于操作系统的分页文件。它用于存储用户创建的临时对象或中间结果、数据库引擎需要的临时对象和版本信息。tempdb 数据库由整个系统的所有数据库使用,不管用户使用哪个数据库,他们建立的所有临时表和存储过程都存储在 tempdb 上。实际上,它只是 SQL Server 的临时工作空间,SQL Server 关闭后,该数据库中的内容被清空。tempdb 数据库在每次重启 SQL Server 时被重新创建。

## 2. 用户数据库

用户数据库是指由用户创建并使用的数据库,主要用于存储用户使用的数据信息。用户数据库由用户定义,且由永久存储表和索引等数据库对象的磁盘空间构成,空间被分配在一个或多个操作系统文件上。本书将主要介绍用户数据库的创建、修改、删除等操作。系统数据库与用户数据库的结构如图 3-14 所示。

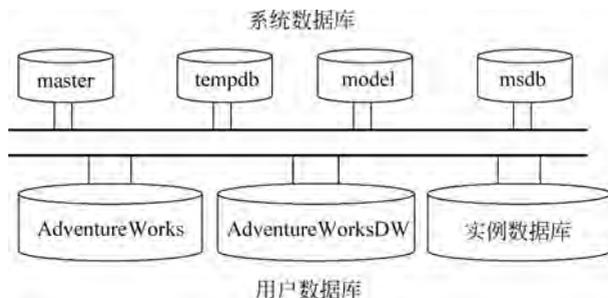


图 3-14 系统数据库与用户数据库的结构

### 3. 示例数据库

示例数据库是一种实用的学习数据库的范例,SQL Server 引入 Adventure Works Cycles 公司的 Adventure Works 示例数据库供用户学习。在默认情况下,SQL Server 2016 不安装示例数据库,需要进行单独安装和设置。

通常,每个 SQL Sever 实例包括 4 个系统数据库(master、model、msdb 和 tempdb)以及一个或多个用户数据库。

#### 3.4.3 SQL Server 2016 数据库的状态

SQL Server 2016 数据库总是处于一个特定的状态中,这些状态包括联机(ONLINE)、脱机(OFFLINE)、还原(RESTORING)、恢复(RECOVERING)、恢复挂起(RECOVERY PENDING)、可疑(SUSPECT)、应急(EMERGENCY),如表 3-3 所示。若要确认数据库的当前状态,可以选择 sys.databases 目录视图中的 state\_desc 列或 DATABASEPROPERTYEX 函数中的 status 属性。

表 3-3 数据库的状态

状 态	定 义
ONLINE	联机状态,可以对数据库进行访问。即使可能尚未完成恢复的撤销阶段,主文件组仍处于在线状态
OFFLINE	脱机状态,数据库无法使用。数据库由于显式的用户操作而处于离线状态,并保持离线状态直到执行了其他的用户操作。例如,可能会让数据库离线以便将文件移至新的磁盘。在完成移动操作后,使数据库恢复在线状态
RESTORING	正在还原主文件组的一个或多个文件,或正在脱机还原一个或多个辅助文件。数据库不可用
RECOVERING	正在恢复数据库。恢复进程是一个暂时性状态,恢复成功后数据库将自动处于在线状态。如果恢复失败,数据库将处于可疑状态。数据库不可用
RECOVERY PENDING	SQL Server 在恢复过程中遇到了与资源相关的错误。数据库未损坏,但是可能缺少文件,或系统资源限制可能导致无法启动数据库。数据库不可用,需要用户另外执行操作来解决问题,并让恢复进程完成
SUSPECT	至少主文件组可疑或可能已损坏。在 SQL Server 启动过程中无法恢复数据库。数据库不可用,需要用户另外执行操作来解决问题
EMERGENCY	用户更改了数据库,并将其状态设置为 EMERGENCY。数据库处于单用户模式,可以修复或还原。数据库标记为 READ_ONLY,禁用日志记录,并且仅限 sysadmin 固定服务器角色的成员访问。EMERGENCY 主要用于故障排除。例如,可以将标记为“可疑”的数据库设置为 EMERGENCY 状态。这样可以允许系统管理员对数据库进行只读访问。只有 sysadmin 固定服务器角色的成员才可以将数据库设置为 EMERGENCY 状态

脱机与联机是针对数据库的当前状态来说的,当一个数据库处于可操作、可查询的状态时就是联机状态,而一个数据库尽管可以看到其名字出现在数据库结点中,但对其不能

执行任何有效的数据库操作时就是脱机状态。

在数据库管理及软件开发过程中,经常会出现对当前数据库进行迁移的操作,而在联机状态下,SQL Server Management Studio 工具是不允许复制数据库文件的。若在数据库复制过程中需要暂停当前的联机数据库,就可以在“对象资源管理器”中选择指定的“数据库”,例如 StudentMIS 数据库,然后右击,在弹出的快捷菜单中选择“任务”→“脱机”命令。在完成对脱机状态的数据库的复制后,要将其恢复为可用状态,可以选择指定的“数据库”,右击,在弹出的快捷菜单中选择“任务”→“联机”命令来实现数据库联机。

#### 3.4.4 SQL Server 2016 数据库的分离与附加

系统管理员在进行系统维护之前、发生硬件故障之后或者更换硬件时都需要对数据库进行迁移,这就需要使用 SQL Server 的分离和附加操作了。同时,对于学生在实验室创建的数据库、表和数据等,由于公用机房使用还原系统,下次开机后数据自动丢失,需要每次实验结束后将数据库及其数据保存到 U 盘供后续实验使用,也需要使用 SQL Server 的分离和附加操作。

如果存在下列任何情况,不能分离数据库:

(1) 已复制并发布的数据库。如果进行复制,则数据库必须是未发布的。用户必须通过运行 `sp_replicationdboption` 禁用发布后才能分离数据库。

(2) 数据库中存在数据库快照。此时必须首先删除所有数据库快照,然后才能分离数据库。

(3) 该数据库正在某个数据库镜像会话中进行镜像。除非终止该会话,否则无法分离该数据库。

(4) 数据库处于可疑状态。

(5) 该数据库是系统数据库。

##### 1. 数据库的分离及保存

分离数据库是指移动保存完整的数据库及其数据文件和事务日志文件,同时将数据库的定义从 SQL Server 的数据库引擎中删除,但并不会删除数据库存储在磁盘上的数据库文件。实际分离数据库的操作步骤如下:

(1) 启动 SQL Server Management Studio,在“对象资源管理器”中选择指定的“数据库”,例如 StudentMIS 数据库,然后右击,在弹出的快捷菜单中选择“任务”→“分离”命令,如图 3-15 所示。

(2) 出现“分离数据库”对话框,如图 3-16 所示,勾选“删除连接”复选框,然后单击“确定”按钮即可分离该数据库。

(3) 保存数据库。找到分离数据库的数据文件(StudentMIS.mdf)和事务日志文件(StudentMIS\_log.ldf),复制到 U 盘或指定的位置即可。



(2) 启动 SQL Server Management Studio,在“对象资源管理器”中右击“数据库”选项,弹出快捷菜单,选择“附加”命令,出现“附加数据库”对话框,如图 3-17 所示。



图 3-17 “附加数据库”对话框

(3) 单击“添加”按钮,出现“定位数据库文件”对话框,如图 3-18 所示。从中选择要附加的数据库的主数据文件 StudentMIS. mdf,然后单击“确定”按钮,返回“附加数据库”对话框。



图 3-18 选择要附加的数据库的主数据文件 StudentMIS. mdf

(4) 在“要附加的数据库”区域和“‘StudentMIS’数据库详细信息”区域显示相关的信息,如图 3-19 所示。



图 3-19 完成附加数据库文件的选择

(5) 确认无误后,单击“确定”按钮,即可把所选的 StudentMIS 数据库添加到当前 SQL Server 实例上。

**说明:** 如果附加数据库失败,可能是当前用户对于 StudentMIS 数据库的权限不够。右击 StudentMIS.mdf 文件,打开文件属性对话框,选择“安全”选项卡,然后选中 Authenticated Users 组,单击“编辑”按钮,选择“完全控制”权限即可。

### 3.5 Transact-SQL 程序设计基础

Transact-SQL 语言是使用 SQL Server 的核心。在 SQL Server 2016 中,与 SQL Server 实例通信的所有应用程序都通过将 Transact-SQL 语句发送到服务器,实现数据的查询、操纵和控制等功能。因此 Transact-SQL 是 SQL Server 与应用程序之间的语言,是 SQL Server 对应用程序开发的接口语言。每一条 Transact-SQL 语句都包含一系列元素,例如标识符、数据类型、变量、运算符、函数、表达式等,因此本节将对 Transact-SQL 语言程序设计的各个元素进行简要介绍。

### 3.5.1 SQL 与 Transact-SQL

#### 1. SQL 概述

SQL 是结构化查询语言(Structured Query Language)的英文缩写,是一种操作关系型数据库的语言。作为一种操作关系型数据库的标准语言,SQL 自问世以来得到了广泛的应用,不仅是著名的大型商用数据库产品 Oracle、DB2、Sybase、SQL Server 支持它,很多开源的数据库产品(例如 PostgreSQL、MySQL)也支持它,甚至一些小型的产品(例如 Access)也支持 SQL。近些年蓬勃发展的 NoSQL 系统最初是宣称不再需要 SQL 的,后来不得不修正为 Not Only SQL 来“拥抱”SQL。

SQL 最早由 IBM 圣约瑟实验室的 Boyce 和 Chamberlin 于 1974 年提出,并作为 IBM 公司研制的关系数据库管理系统原型 System R 的一部分付诸实施。当时它称为 SEQUEL,后简称为 SQL。该语言简洁、功能丰富、易学易用,不仅具有数据定义、数据操纵和数据控制功能,还有着强大的查询功能。现在 SQL 已经成为关系数据库的标准语言,并且发展了 7 个主要标准。

(1) SQL-86。1986 年,美国国家标准化组织(America National Standards Institute, ANSI)正式发表了编号为 X3.135-1986 的 SQL 标准,并且在 1987 年得到了国际标准化组织(International Standards Organization, ISO)的认可,被命名为 ISO/IEC 9075:1987。

(2) SQL-89。1989 年,ANSI 正式发表了编号为 X3.135-1989 的 SQL 标准,而 ISO 将其命名为 ISO/IEC 9075:1989。

(3) SQL-92(也称 SQL2)。1992 年,ANSI 正式发表了编号为 ANSI X3.135-1992 的 SQL 标准,而 ISO 将其命名为 ISO/IEC 9075:1992。

(4) SQL:1999(也称 SQL3)。1999 年,ISO 在 SQL2 的基础上推出了 ISO/IEC 9075:1999,并增加了对象关系特征和许多其他的新功能。从 SQL:1999 开始,标准简称中的短横线(-)被换成了冒号(:),而且标准制定的年份也改用了 4 位数字。前一个修改的原因是 ISO 标准习惯上采用冒号,而 ANSI 标准一直采用短横线;后一个修改的原因是标准的命名也遇到了 2000 年问题。

(5) SQL:2003。2003 年,ISO 推出了 ISO/IEC 9075:2003 标准。

(6) SQL:2008。2008 年,ISO 推出了 ISO/IEC 9075:2008 标准。

(7) SQL:2011。2011 年,ISO 推出了 ISO/IEC 9075:2011 标准。

现在各大数据库厂商提供不同版本的 SQL。这些版本的 SQL 不仅都包括原始的 ANSI 标准,而且还在很大程度上支持 SQL-92 和 SQL:1999 标准。这使不同的数据库系统之间的互操作有了可能。SQL 语言集数据定义、数据操纵、数据查询和数据控制功能于一体,主要特点包括:

##### 1) 多功能综合统一

交互式查询语言功能强大、简单易学,而且集数据定义、数据操纵、数据查询、数据控制和附加语言于一体。数据库的主要功能是通过数据库支持的数据语言来实现的,SQL 语言的核心包括如下数据语言:

(1) 数据定义语言(Data Definition Language, DDL)。数据定义语言用于定义和管理数据库及其对象,是对关系模式一级的定义,包括数据库、基本表、视图、索引等的创建、修改和删除操作。

(2) 数据操纵语言(Data Manipulation Language, DML)。数据操纵语言用于对数据库中数据的操作,包括插入(INSERT)新数据、删除(DELETE)旧数据、修改(UPDATE)已有数据等。

(3) 数据查询语言(Data Query Language, DQL)。数据查询语言按一定的查询条件从数据库对象(基本表或视图)中检索(SELECT)符合条件的数据。

(4) 数据控制语言(Data Control Language, DCL)。数据控制语言主要用于权限和安全管理,用来控制对数据库中数据的操作,包括基本表和视图等对象的授权、完整性规则的描述、事务开始和结束控制语句等。

(5) 其他附加的语言。这些附加语言主要用于辅助命令语句的操作、标识、理解和使用,包括标识符、数据类型、常量、变量、运算符、函数、表达式、批处理、流程控制、错误处理和注释等。

SQL 语言集这些功能于一体,语言风格统一,可以独立完成数据库生命周期中的全部活动,包括定义关系模式、建立数据库、查询和更新数据、数据库重构、数据库安全控制等一系列操作要求,这就为数据库应用系统开发提供了良好的环境。

#### 2) 高度非过程化

使用 SQL 语言进行数据操作,用户只需提出“做什么”,而不必指明“怎么做”,用户只需将要求用 SQL 语句提交给系统,系统会自动完成所需的操作。这不仅极大地减轻了用户负担,而且有利于提高数据独立性。

#### 3) 面向集合的操作方式

SQL 操作的对象和结果都是集合(关系),插入、删除、修改的对象和查找的结果均为元组的集合。通常,用关系(二维数据表)表示数据处理操作更快捷、方便。

#### 4) 灵活的使用方法

SQL 语言既是自含式语言,又是嵌入式语言。作为自含式语言,它能够独立地用于联机交互的使用方式,用户可以在终端键盘上直接输入 SQL 命令对数据库进行操作。作为嵌入式语言,SQL 语句能够嵌入高级语言(例如 Java)程序中,供程序员设计程序时使用。在两种方式下,SQL 语言的语法结构基本上是一致的。这种统一的语法结构提供了两种不同的使用方式,为用户提供了极大的灵活性与方便性。

#### 5) 语言简洁、易学易用

SQL 语言功能极强,且十分简洁,完成数据定义、数据操纵、数据查询、数据控制的核心功能只用了 10 个关键字,其中数据定义 3 个(CREATE、DROP、ALTER)、数据操纵 3 个(INSERT、UPDATE、DELETE)、数据查询 1 个(SELECT)、数据控制 3 个(GRANT、DENY、REVOKE)。SQL 用关键字、表名、属性名等组合而成的一条语句(SQL 语句)来描述操作的内容,语法简单,接近英语口语,因此易学易用。

## 2. Transact-SQL 概述

Transact-SQL 语言是微软公司在 SQL Server 系统中使用的事务-结构化查询语言,

支持标准的 SQL,但对 SQL 语言进行了扩展。Transact-SQL 对 SQL 的扩展主要包含以下 4 个方面:

(1) 增加了流程控制语句。

(2) 加入了局部变量、全局变量、表达式、函数等许多新概念,可以写出更复杂的查询语句。

(3) 增加了新的数据类型,处理能力更强。

(4) 增加了事务管理语言,主要用于事务管理操作。

事务是指用户定义的一个数据库操作序列,这些操作“要么都做,要么都不做”,是一个不可分割的工作单位。在数据库中执行操作时,经常需要多个操作同时完成或同时撤销。例如,将账户 A 的资金转账到账户 B,需要两个更新操作(账户 A 的余额减少,账户 B 的余额增加相应的数额),这就属于事务管理,在执行过程中或者两个更新操作都做,或者都不做,避免数据不一致。在 SQL Server 2016 中,可用 COMMIT 语句提交事务,也可用 ROLLBACK 语句撤销。

Transact-SQL 语言是一种交互式查询语言,既允许用户直接查询存储在数据库中的数据,也可以将 Transact-SQL 语句嵌入某种高级程序设计语言中使用,例如可嵌入 C#、C++ 或 Java 等语言中,具有功能强大、易学易用的特点。对于数据库中数据集的操作来说,Transact-SQL 比其他高级语言更加简单、高效,而且也具有了其他高级语言的特点,因此得到了广泛的应用。

### 3. Transact-SQL 的执行方式

在 SQL Server 2016 系统中提供了多种图形界面和命令行工具,用户可以使用不同的方法来访问数据库。但是这些工具的核心却是 Transact-SQL 语言。在 SQL Server 2016 中主要使用 SQL Server Management Studio 工具来执行 Transact-SQL 语言编写的查询语句,用于交互地设计和测试 Transact-SQL 语句、批处理和脚本。Transact-SQL 对于 SQL Server 非常重要,SQL Server 中使用图形界面和命令行工具能够完成的所有功能都可以使用 Transact-SQL 实现。因此,本书主要以 Transact-SQL 为主线进行介绍。

若想使用 SQL Server Management Studio 工具运行编写的 Transact-SQL 语句,可先在“对象资源管理器”中选中要运行 Transact-SQL 语句的数据库或者数据库中的对象,然后单击“新建查询”按钮或者按 Alt+N 组合键,SQL Server Management Studio 将新建一个空白查询编辑器窗口,用户可在此编写 Transact-SQL 语句并执行。在执行时需要注意的是,若用户在编辑器中选中部分 Transact-SQL 脚本,SQL Server Management Studio 将只运行选中的脚本。若用户没有在查询编辑器窗口中选择任何脚本,SQL Server Management Studio 将运行该窗口中的所有 Transact-SQL 脚本。

在 SQL Server Management Studio 中还支持对大多数数据库对象(例如表、视图、同义词、存储过程、函数、触发器等)生成操作 Transact-SQL 语句,该功能可减少开发人员反复编写 Transact-SQL 语句的工作,极大地提高了工作效率。例如要生成查询 Student 表的 SQL 语句,只需在“对象资源管理器”中找到该表,在该表上右击,在弹出的快捷菜单

中选择“编写表脚本为”→“SELECT 到”→“新查询编辑器窗口”命令,如图 3-20 所示。

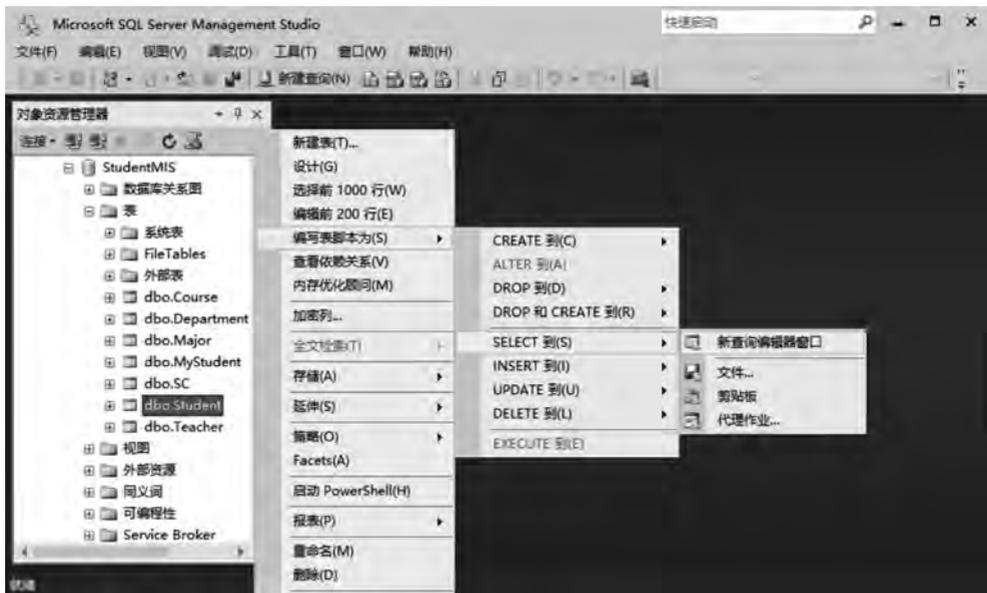


图 3-20 为表生成查询 Transact-SQL 语句

SQL Server Management Studio 可自动生成该表的查询语句。单击工具栏中的 **! 执行(X)** 按钮,运行该语句,将在主区域的下方显示运行结果。自动生成的查询语句和运行结果如图 3-21 所示。通过同样的操作,不仅可以自动生成查询语句,还可以自动生成表的创建、插入、更改和删除等操作的 Transact-SQL 语句。

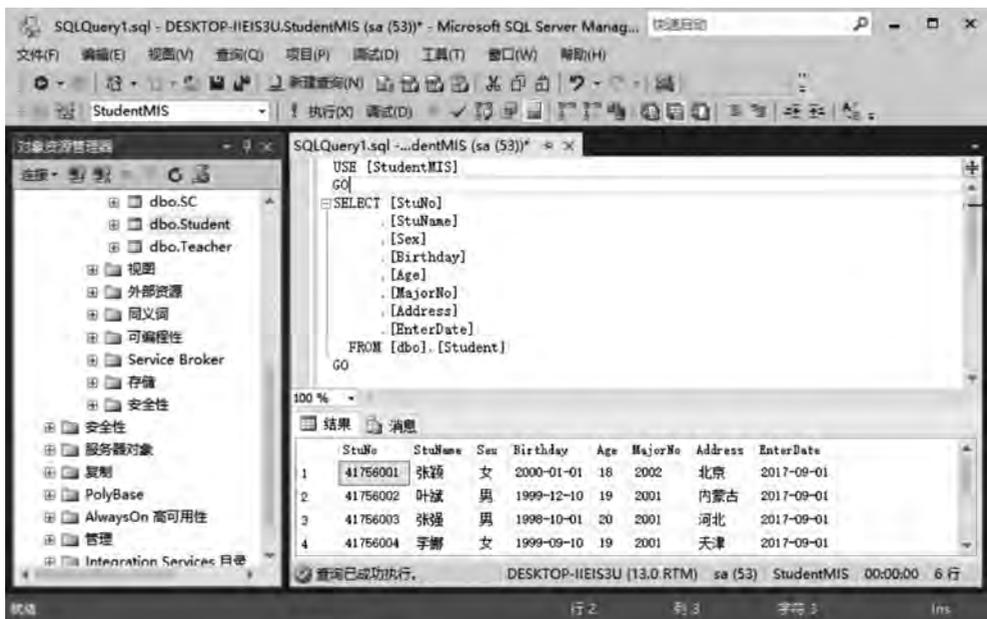


图 3-21 自动生成的查询语句及其执行结果

### 3.5.2 SQL Server 2016 的标识符

在 SQL Server 2016 中,标识符(Identifier)就是指用来定义服务器、数据库、数据库对象(表、视图、索引、触发器、存储过程等)和变量等名称的字符串,不区分大小写。按照标识符的使用方法,标识符可以分为常规标识符和分隔标识符两种。

#### 1. 常规标识符

常规标识符合标识符的格式规则,又称规则标识符,就是不需要使用分隔标识符进行分隔的标识符。在 Transact-SQL 语句中使用常规标识符时不用将其分隔。

常规标识符的格式规则如下:

- (1) 第一个字符必须是 Unicode 标准 3.2 所定义的字母(例如 a~z 和 A~Z 以及来自其他语言的字母字符)、下画线(\_)、at 符号(@)和数字符号(#)。
- (2) 后续字符可以是 Unicode 标准 3.2 所定义的字母、来自基本拉丁字母或其他国家/地区脚本的十进制数字、下画线(\_)、at 符号(@)、美元符号(\$)、数字符号(#)。
- (3) 标识符不能使用 Transact-SQL 的保留关键字,例如函数名 max、min、asc 等。
- (4) 标识符内不允许嵌入空格或其他特殊字符,例如?、%、&、\* 等。
- (5) 常规标识符和分隔标识符包含的字符数必须在 1~128。对于本地临时表而言,标识符最多可以有 116 个字符。

**注意:**在 SQL Server 中,某些处于标识符开始位置的符号具有特殊意义。以 at 符号(@)开始的标识符表示局部变量或参数;以双 at 符号(@@)开始的标识符表示全局变量;以一个数字符号(#)开始的标识符表示临时表或过程;以双数字符号(##)开始的标识符表示全局临时对象。

#### 2. 分隔标识符

分隔标识符是使用[ ]、'或"等起到分隔作用的符号来限定的标识符。在 Transact-SQL 语句中,对不符合所有标识符规则的标识符必须进行分隔。符合标识符格式规则的标识符可以分隔,也可以不分隔,二者是等效的。

当标识符中包含 SQL Server 关键字或包含了内嵌的空格和其他不是规则规定的字符时,要使用分隔符中的括号([ ])将标识符括起来。例如,下面语句中的 Student Table 和 in 均不符合标识符规则,其中 Student Table 中间出现了空格,而 in 为 Transact-SQL 的保留关键字,因此必须使用分隔符进行分隔:

```
SELECT * FROM [Student Table] WHERE [in] = 5
```

分隔标识符在下列情况下使用:

- (1) 当在对象名称或对象名称的组成部分中使用保留关键字时。注意不要使用保留关键字作为对象名称。从 SQL Server 早期版本升级的数据库可能含有标识符,这些标识符包括早期版本中未保留而在 SQL Server 2016 中保留的关键字。用户可用分隔标识符引用对象直到可改变其名称。

(2) 当使用未被列为合法标识符的字符时。SQL Server 允许在分隔标识符中使用当前代码页中的任何字符,但是不加选择地在对象名称中使用特殊字符将使 SQL 语句和脚本难以阅读和维护。

分隔标识符的格式规则如下:

(1) 分隔标识符可以包含与常规标识符相同的字符数(1~128 个,不包括分隔符字符)。本地临时表标识符最多可以包含 116 个字符。

(2) 标识符的主体可以包含当前代码页内字母(分隔符本身除外)的任意组合。例如,分隔标识符可以包含空格、对常规标识符有效的任何字符以及代字号(~)、连字符(-)、惊叹号(!)、左括号({)、百分号(%)、右括号(})、插入号(^)、撇号(')、and 号(&)、句号(.)、左圆括号(())、反斜杠(\)、右圆括号())、重音符号(˘)等。

### 3. 数据库对象命名规则

数据库对象的名称被看成是该对象的标识符。SQL Server 中的每一内容都可带有标识符,服务器、数据库和数据库对象(例如表、视图、列、索引、触发器、过程、约束及规则等)都有标识符。大多数数据库对象要求带有标识符,但对于有些对象(例如约束),标识符是可选项。在 SQL Server 2016 中,一个数据库对象的全称语法格式为:

```
[server_name].[database_name].[schema_name].object_name
```

其中,server\_name 指定连接的服务器名称或远程服务器名称。对于 database\_name,如果对象驻留在 SQL Server 的本地实例中,则指定 SQL Server 数据库的名称;如果对象在连接服务器中,则 database\_name 将指定 OLE DB 目录。对于 schema\_name,如果对象在 SQL Server 数据库中,则指定包含对象的架构的名称;如果对象在连接服务器中,则 schema\_name 将指定 OLE DB 架构名称。object\_name 为对象的名称。

在实际使用时,使用全称比较烦琐,因此经常使用简写格式。若要省略中间结点,请使用句点来指示这些位置。数据库对象的引用格式如表 3-4 所示。

表 3-4 数据库对象的引用格式

对象引用格式	说 明
server.database.schema.object	4 个部分的名称
server.database..object	省略架构名称
server..schema.object	省略数据库名称
server...object	省略数据库和架构名称
database.schema.object	省略服务器名
database..object	省略服务器和架构名称
schema.object	省略服务器和数据库名称
object	省略服务器、数据库和架构名称

在上面的简写格式中,没有指明的部分使用默认设置值。server: 本地服务器; database: 当前数据库; schema: 包含该对象的架构的名称。

**【例 3-1】** 一个用户名为 sa 的用户登录到 MyServer 服务器上,并使用 MyDB 数据库。

使用下述语句创建一个 MyTable 表:

```
CREATE TABLE MyTable(column1 int, column2 char(20))
```

则表 MyTable 的全称就是 MyServer. MyDB. dbo. MyTable。

#### 4. Transact-SQL 语句使用的语法规则

Transact-SQL 语句使用的语法规则如下:

- (1) “< >”(尖括号)中的内容表示“必选项”,不可省略,不需输入尖括号。
- (2) “[ ]”(方括号)中的内容为可选项,省略时系统取默认值,无须输入方括号。
- (3) { } (大括号)中的内容表示“必选项”,无须输入大括号。
- (4) “|”(或/)分隔尖括号、方括号或大括号中的语法项,表示相邻前后两项只能选择其中一项。
- (5) [ , … n ] 指示前面的项可以重复  $n$  次,且每一项由逗号分隔。
- (6) [ … n ] 指示前面的项可以重复  $n$  次,且每一项由空格分隔。
- (7) 一条较长语句可分成多行书写且以分号 (;) 结尾,但是在同一行中不允许写多条语句。在 SQL Server 中,SQL 语句是逐条执行的,每条语句以“;”结尾。
- (8) 在一个关键字的中间不能加入空格或换行符。
- (9) 在 Transact-SQL 中,保留关键字(Transact-SQL 中系统预留或事先定义好的关键字)、命令和语句的书写不区分大小写,但是插入表中的数据是区分大小写的。关键字不能被缩写,也不能分行。虽然 Transact-SQL 语言不区分大小写,但是本书中的关键字使用大写,用户定义的标识符使用小写。
- (10) 在书写各种 Transact-SQL 语句时,所用的标点符号(例如空格、括号、逗号、分号、圆点(英文句号))等都应是英文半角,若写成中文符号或全角符号,将会在执行命令时出错。

### 3.5.3 SQL Server 2016 的数据类型

在 SQL Server 中,每个表或视图中的字段、常量、Transact-SQL 程序中的变量、表达式、存储过程中的参数等都有其各自的数据类型,指定对象的数据类型相当于定义该对象的下列 4 种属性:对象所含的数据类型,例如字符、整数或二进制数;数据的长度;数值精度;数值中的小数位数。其中,后两种属性仅用于数值数据类型。

在 SQL Server 中提供了两类数据类型供用户选择,即系统数据类型和用户自定义数据类型。系统数据类型由 SQL Server 提供系统数据类型集,定义可供使用的所有数据类型,用户可直接使用上述数据类型。用户定义数据类型是用户根据实际需要在系统基本数据类型的基础上定义自己的数据类型,是出于系统可扩展性的需要和考虑。

SQL Server 提供的系统数据类型主要包括数值型、字符型、日期型等。

## 1. 整数数据类型

整数数据类型包括 tinyint、smallint、int、bigint，均为精确数值型，如表 3-5 所示。

表 3-5 整数数据类型

数据类型	描 述	存储空间
tinyint	用于存储 0~255 的整数	1 字节
smallint	$-2^{15}(-32768) \sim 2^{15}-1(32767)$ 的整数, 其中一个二进制位表示正负号	2 字节
int	$-2^{31}(-2147483648) \sim 2^{31}-1(2147483647)$ 的整数, 其中一个二进制位表示正负号	4 字节
bigint	$-2^{63}(-9223372036854775808) \sim 2^{63}-1(9223372036854775807)$ 的整数	8 字节

## 2. 浮点数据类型

浮点数据类型用来表示有小数部分的数据, 根据所使用的存储空间, 可以分为 float、real、decimal、numeric、money、smallmoney, 如表 3-6 所示。其中, decimal 和 numeric 属于精确数值型, 而 float 和 real 为近似数值型, money、smallmoney 属于货币型。

表 3-6 浮点数据类型

数据类型	描 述	存储空间
float(n)	范围为 $-1.79E+308 \sim -2.23E-308, 0, 2.23E-308 \sim 1.79E+308$	由 $n$ 决定
real()	范围为 $-3.40E+38 \sim -1.18E-38, 0, 1.18E-38 \sim 3.40E+38$	4 字节
decimal(p,s)	$-10^{38}+1 \sim 10^{38}-1$ 的数值, $p$ 变量指定精度, 取值范围为 $1 \sim 38$ ; $s$ 变量指定小数位数, 取值范围为 $0 \sim p$	最多 17 字节
numeric(p,s)	$-10^{38}+1 \sim 10^{38}-1$ 的数值, 功能上等价于 decimal	最多 17 字节
money	$-922337203685477.5808 \sim 922337203685477.5807$	8 字节
smallmoney	$-214748.3648 \sim 214748.3647$	4 字节

float(n)中的  $n$  是用于存储该数尾数的位数。SQL Server 对此只使用两个值。如果指定位于  $1 \sim 24$ , SQL Server 就使用 24。如果指定  $25 \sim 53$ , SQL Server 就使用 53。当指定 float()时(括号中为空), 默认为 53。real 类型等价于 float(24)。decimal 和 numeric 数值数据类型可存储小数点右边或左边的变长位数。 $p$  表示精度(Precision), 定义了最多可以存储的十进制数字的总位数, 包括小数点左、右两侧的位数, 精度值的范围是  $1 \sim 38$ , 默认精度为 18。 $s$ (Scale)是小数点右侧可以存储的十进制数字的最大位数。当精度为  $1 \sim 9$  时, 存储空间为 5 字节; 当精度为  $10 \sim 19$  时, 存储空间为 9 字节; 当精度为  $20 \sim 28$  时, 存储空间为 13 字节; 当精度为  $29 \sim 38$  时, 存储空间为 17 字节。money 和 smallmoney 类型用来存储货币型数据, 精确到它们所代表的货币单位的 1%。当表中使用货币型数据时, 必须在前面加上货币符号(\$)。若货币为负数, 则需要在符号 \$ 后面加上负号(-)。例如, \$-100.5 是正确的货币表示形式。money 的数据范围和 bigint 相同, 不同的是 money 有 4 位小数。smallmoney 与 int 的关系也是如此。

### 3. 字符数据类型

字符数据类型包括 `varchar`、`char`、`nvarchar`、`nchar`、`text` 以及 `ntext`，如表 3-7 所示。这些数据类型用于存储字符串。当 SQL 语句中含有字符串的时候，需要像 'abc' 这样，使用单引号 (') 将字符串括起来，用来标识这是一个字符串。

表 3-7 字符数据类型

数据类型	描述	存储空间
<code>char(n)</code>	n 的取值范围为 1~8000 个字符	n 字节
<code>nchar(n)</code>	n 的取值范围为 1~4000 个 Unicode 字符	2n 字节
<code>varchar(n)</code>	n 的取值范围为 1~8000 个字符	n 字节+2 字节额外开销
<code>varchar(max)</code>	最多为 $2^{31}-1$ (2147483647) 个字符	n 字节+2 字节额外开销
<code>nvarchar(n)</code>	n 的取值范围为 1~4000 个 Unicode 字符	2n 字节+2 字节额外开销
<code>nvarchar(max)</code>	最多为 $2^{30}-1$ (1073741823) 个 Unicode 字符	2n 字节+2 字节额外开销
<code>text</code>	最多为 $2^{31}-1$ (2147483647) 个字符	n 字节
<code>ntext</code>	最多为 $2^{30}-1$ (1073741823) 个 Unicode 字符	2n 字节

`varchar` 和 `char` 类型的主要区别是数据填充。如果有一个表的列名为 Name，且数据类型为 `varchar(8)`，同时将值 Tom 存储到该列中，则物理上只存储 3 字节。如果在数据类型为 `char(8)` 的列中存储相同的值，将使用全部 8 字节，SQL 将在 Tom 后面插入半角空格来填满 8 个字符。所以，称 `char` 为定长字符类型，称 `varchar` 为可变长字符类型。

`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 Tom 存储到 `nvarchar` 列，它将使用 6 字节；而如果将它存储为 `nchar(20)`，则需要使用 40 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。

`text` 数据类型用于在数据页内外存储大型字符数据。用户应尽可能少地使用这两种数据类型，因为可能影响性能，但可在单行的列中存储多达 2GB 的数据。与 `text` 数据类型相比，更好的选择是使用 `varchar(max)` 类型，因为将获得更好的性能。另外，`text` 和 `ntext` 数据类型在 SQL Server 的一些未来版本中将不可用，因此从现在开始还是最好使用 `varchar(max)` 和 `nvarchar(max)`，而不是 `text` 和 `ntext` 数据类型。

### 4. 日期和时间数据类型

用户以字符串的形式输入日期和时间类型数据，系统也以字符串形式输出日期和时间类型数据。SQL Server 支持的日期和时间数据类型如表 3-8 所示。

表 3-8 日期和时间数据类型

数据类型	描述	精确度	存储空间
date	9999年1月1日~12月31日	1天	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	3.33ms	8字节
datetime2(n)	9999年1月1日~12月31日,0~7的n指定小数秒	100ns	6~8字节
datetimeoffset(n)	9999年1月1日~12月31日 0~7的n指定小数秒+/-偏移量	100ns	8~10字节
smalldatetime	1900年1月1日~2079年6月6日,精确到1分钟	1分钟	4字节
time(n)	小时:分钟:秒,9999999,0~7的n指定小数秒	用户指定 小数位数	3~5字节

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 日的时间,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 日的时间,且精确到最近的 3.33 毫秒。当 SQL 语句中含有日期的时候,需要使用单引号将其括起来。日期的格式有很多种('15 Jan 2018'或者'18/01/15'等),本书统一使用'2018-01-15'这种'年-月-日'的格式。

SQL Server 2016 有 4 种与日期相关的新数据类型,即 datetime2、datetimeoffset、date 和 time。用户通过 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、秒形式来存储。用户可以定义末尾带有可变参数的 datetime2 数据类型,例如 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。其有效值为 0~9,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。该时区偏移量最大为 +/-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,这是一直需要的一个功能。time 数据类型只存储时间。它也支持 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。

## 5. 二进制数据类型

二进制数据类型用于存储二进制数据,例如图形文件、Word 文档或 MP3 文件等,值为十六进制的 0x0~0xf,包括 bit、binary、varbinary、varbinary(max) 和 image,如表 3-9 所示。

表 3-9 二进制数据类型

数据类型	描述	存储空间
bit	0、1 或 NULL	1 字节
binary(n)	n 为 1~8000 的十六进制数字	n 字节
image	最多为 $2^{31}-1$ (2147483647) 十六进制数字	每字符 1 字节
varbinary(n)	n 为 1~8000 的十六进制数字	n 字节+2 字节额外开销
varbinary(max)	最多为 $2^{31}-1$ (2147483647) 十六进制数字	每字符 1 字节+2 字节额外开销

## 6. 其他数据类型

除了以上数据类型,SQL Server 2016 还提供了一些新的数据类型,如表 3-10 所示。

表 3-10 其他数据类型

数据类型	描述	存储空间
NULL	表示什么也没有,不同于空格。按 Ctrl+0 组合键可在单元格中输入 NULL	
sql_variant	可包含除 text、ntext、image、timestamp、xml、varchar(max)、nvarchar(max)、varbinary(max)以及用户定义的数据类型之外的其他任何系统数据类型的值	8016 字节
table	存储用于进一步处理的数据集。其定义类似于 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中	取决于表定义和存储的行数
uniqueidentifier	包含全局唯一标识符(Globally Unique Identifier, GUID)。GUID 值可以从 Newid() 函数获得。一般用作主键的数据类型,是由硬件地址、CPU 标识、时钟频率所组成的随机数据,在理论上每次生成的 GUID 都是全球唯一的。尽管存储为 16 位的二进制值,但它显示为 char(36)	16 字节
rowversion	每一次对数据表的更改,SQL Server 都会更新一个内部的序列数,这个序列数就保存在 rowversion 字段中。所有 rowversion 列的值在数据表中是唯一的,并且每张表中只能有一个包含 rowversion 字段的列存在。使用 rowversion 作为数据类型的列,其字段本身的内容是无自身含义的,这种列主要是作为数据是否被修改过,更新是否成功的作用列	8 字节
timestamp	timestamp 时间戳数据类型和 rowversion 有一定的相似性,当插入或者修改行时,自动生成的唯一的二进制数字的数据类型,通常用于时间戳。在创建表时只需提供数据类型即可,不需要为 timestamp 所在的数据列提供列名。	8 字节
cursor	游标数据类型允许在存储过程中创建游标变量,游标允许一次一行地处理数据,这个数据类型不能用作表中的列数据类型	
XML	可以以 Unicode 或非 Unicode 形式存储	最多 2GB
hierarchyid	表示树层次结构中的位置	1~892 字节+2 字节额外开销

### 3.5.4 常量、变量、运算符

#### 1. 常量

常量是指在程序运行过程中其值保持不变的量。常量是表示一个特定数据值的符号,也称为文字值或标量值。根据不同的数据类型,常量可分为字符型常量、整型常量、日期常量、实型常量、货币常量和唯一标识(uniqueidentifier)常量。对于非数值型常量需要使用单引号。

### 1) 字符型常量

字符型常量括在单引号内并包含字母(a~z、A~Z)、数字(0~9)以及特殊字符,例如感叹号(!)、at符号(@)和数字符号(#)。如果单引号中的字符串包含一个嵌入的单引号,可以使用两个单引号表示嵌入的单引号。对于嵌入在双引号中的字符串则没有必要这样做。例如,'This is a book'是一个字符型常量。字符型常量包括 ASCII 字符型常量和 Unicode 字符型常量两种。

(1) ASCII 字符型常量。用单引号括起来,由 ASCII 字符构成的字符串,例如'Tom'。

(2) Unicode 字符型常量。通常在常量前面有一个 N 标识符,例如 N'Tom'(其中的 N 在 SQL-92 标准中表示国际语言,要求必须大写)。

### 2) 整型常量

整型常量通常表示整数,主要包括二进制整型常量、十进制整型常量和十六进制整型常量。整型常量不用引号括起来,且不能包含小数。二进制常量如 100;十进制常量如 18;十六进制常量具有前缀 0x,例如 0x1a。

### 3) 日期时间型(datetime)常量

日期时间型常量是表示日期或时间的常量,要求用单引号将所表示的日期或时间括起来。例如,'December 5, 2018','12/5/2018','14:30:24'等。

### 4) 实型常量

decimal 常量由没有用引号括起来并且包含小数点的数字字符串来表示。例如 123.45。float 和 real 常量使用科学记数法来表示,例如 1.5E5。

### 5) 货币常量

货币常量以前缀为可选的小数点和可选的货币符号的数字字符串来表示。以货币符号开头,例如 \$100.5、¥12.0。SQL Server 不强制分组,每隔 3 个数字插入一个逗号进行分隔。

### 6) uniqueidentifier 常量

uniqueidentifier 常量是表示全局唯一标识符的字符串,可以使用字符或二进制字符串格式指定,是 SQL Server 根据计算机网络适配器地址和主机时钟产生的唯一号码生成的全局唯一标识符。例如,0xff19966f868b11d0b42d00c04fc964ff、'6F9619FF-8B86-D011-B42D-00C04FC964FF'。

## 2. 变量

变量是指在程序运行过程中其值可以发生改变的量,包括用户自己定义的局部变量和系统提供的全局变量两种。在局部变量前面有一个@字符,由用户自定义和使用;在全局变量名称前面有两个@字符,由系统定义和维护。

### 1) 局部变量

局部变量由用户自定义,仅在声明它的批处理、存储过程或者触发器中有效。在批处理结束后,局部变量将变成无效。局部变量用于保存特定类型的单个数据值的对象。在 Transact-SQL 语言中,可以使用 DECLARE 语句声明变量。在声明变量时需要注意为变量指定名称,且名称的第一个字符必须是@;指定该变量的数据类型和长度;在默认

情况下将该变量值设置为 NULL。其语法格式如下：

```
DECLARE { @local_variable data_type}[, ...n]
```

各参数含义如下：

(1) @local\_variable 是局部变量的名称。变量名必须以 at 符号(@)开头,并符合 SQL Server 标识符命名规则。

(2) data\_type 是任何由系统提供的或用户定义的数据类型。变量不能是 text、ntext 或 image 数据类型。

用户可以在一个 DECLARE 语句中声明多个变量,变量之间使用逗号分隔。例如：

```
DECLARE @maxprice float, @pub char(12)
```

局部变量的使用也是必须先声明,然后再赋值,默认初值是 NULL。通常有两种为变量赋值的方式,即使用 SET 语句为变量赋值,和使用 SELECT 语句选择列表中当前所引用值为变量赋值。其语法格式如下：

```
SET @local_variable = expression[, ...n]  
SELECT { @local_variable = expression[, ...n] [FROM 子句] [WHERE 子句]}
```

其中,@local\_variable 为定义的局部变量名称,expression 为一表达式。如果省略了 FROM 子句和 WHERE 子句,则使用 SELECT 语句和 SET 语句给变量赋值,二者等价。

**【例 3-2】** 首先定义两个变量,并分别使用 SET 和 SELECT 为其赋值,然后使用这两个变量查询价格小于 40 且出版社为“国防工业出版社”的书籍的信息。

```
DECLARE @maxprice float, @pub char(12)  
SET @maxprice = 40  
SELECT @pub = '国防工业出版社'  
SELECT * FROM booklist WHERE price < @maxprice AND publisher = @pub
```

SELECT @local\_variable 通常用于将单个值返回到变量中。例如,如果 expression 为列名,则返回多个值。如果 SELECT 语句返回多个值,则将返回的最后一个值赋给变量。如果 SELECT 语句没有返回行,变量将保留当前值。如果 expression 是不返回值的标量子查询,则将变量设为 NULL。一般来说,应该使用 SET,而不是 SELECT 给变量赋值。

**【例 3-3】** 以消息的方式返回出版社数据库中图书的总数。

```
DECLARE @Number int  
SELECT @Number = count(*) FROM booklist  
PRINT '图书总数为: ' + @Number  
GO
```

## 2) 全局变量

系统全局变量是 SQL Sever 系统定义并提供赋值的变量,用于跟踪和记录服务器范围和特定会话期间的信息,不允许用户显式定义、赋值或修改,也就是说用户既不能定义全局变量,也不能使用 SET 语句对全局变量进行赋值。在 SQL Server 2016 中提供了 33 个全局变量,方便用户监测或了解 SQL Server 服务器活动状态,例如,@@ERROR 返回最后一个 Transact-SQL 语句错误的错误号、@@VERSION 返回 SQL Server 的版本信息、@@IDENTITY 返回最后插入的标识值,对于详细内容可查阅联机帮助。

## 3. 运算符

运算符是一种符号,用来指定要在一个或多个表达式中执行的操作。SQL Server 提供的运算符包括算术运算符、赋值运算符、位运算符、比较运算符、逻辑运算符、字符串连接运算符和一元运算符。运算符在表达式中起到连接变量、常量和函数的作用,同时在连接过程中存在一定的优先级。下面将对 SQL Server 中的运算符的分类和运算符的优先级进行简要论述。

### 1) 运算符的分类

(1) 算术运算符。算术运算符用于两个表达式执行数学运算,这两个表达式可以是任何数值数据类型。在 SQL Server 中,算术运算符包括+(加)、-(减)、\*(乘)、/(除)和%(取模)。取模运算返回一个除法运算的整数余数。例如  $7\%3=1$ ,这是因为 7 除以 3,余数为 1。

(2) 赋值运算符。Transact-SQL 有一个赋值运算符,即等号(=)。它将表达式的值赋给另外一个变量。

**【例 3-4】** 下面的 SQL 语句先声明一个变量,然后将一个取模运算的结果赋给该变量。

```
DECLARE @MyResult int
SET @ MyResult = 7 % 3
```

(3) 位运算符。位运算符用于对两个表达式执行位操作,这两个表达式可以是整型数据或者二进制数据。位运算符包括&(按位与)、|(按位或)、^(按位异或)。Transact-SQL 首先把整型数据转换为二进制数据,然后再对二进制数据进行按位运算。

(4) 比较运算符。比较运算符用于比较两个表达式的大小,表达式可以是字符、数字或日期数据,并可用在查询语句的 WHERE 或 HAVING 子句中。比较运算符的计算结果为布尔数据类型,它们根据测试条件的输出结果返回 TRUE、FALSE 或 UNKNOWN。SQL Server 提供的比较运算符包括>(大于)、<(小于)、=(等于)、<=(小于或等于)、>=(大于或等于)、!=(不等于)、<>(不等于)、!<(不小于)、!>(不大于)。

(5) 逻辑运算符。逻辑运算符用于把多个逻辑表达式连接起来进行测试,以获得其真实情况。其返回带有 TRUE、FALSE 或 UNKNOWN 的布尔数据类型。SQL Server 共提供了 10 个逻辑运算符,如表 3-11 所示。

表 3-11 逻辑运算符

逻辑运算符	含 义
ALL	如果一组比较关系的值都为 TRUE,才返回 TRUE
AND	如果要比较的两个布尔表达式的值都为 TRUE,才返回 TRUE
ANY	只要一组比较关系中有一个值为 TRUE,就返回 TRUE
BETWEEN	只有操作数在定义的范围,才返回 TRUE
EXISTS	如果在子查询中存在,就返回 TRUE
IN	如果操作数在所给的列表表达式中,则返回 TRUE
LIKE	如果操作数与模式相匹配,则返回 TRUE
NOT	对所有其他的布尔运算符的值取反
OR	只要比较的两个表达式有一个为 TRUE,就返回 TRUE
SOME	如果一组比较关系中有一些为 TRUE,则返回 TRUE

(6) 字符串连接运算符。字符串连接运算符为加号(+),可以将两个或多个字符串合并或连接成一个字符串,还可以连接二进制字符串。

(7) 一元运算符。一元运算符是指只有一个操作数的运算符。SQL Server 提供的一元操作符包括+(正)、-(负)和~(按位非)。+和-运算符表示数据的正和负,可以对所有的数据类型进行操作。~运算符返回一个数的补数,只能对整型数据进行操作。

#### 2) 运算符的优先级

运算符的优先级决定了运算符与变量、常量和函数相结合和执行运算的先后顺序,执行的顺序可能严重地影响所得到的值。在 SQL Server 中,运算符的优先级从高到低如表 3-12 所示,如果两个运算符优先级相同,则按照从左到右的顺序进行运算。使用括号可以提高运算符的优先级,首先对括号中的内容进行求值,从而产生一个值,然后括号外的运算符才可以使用这个值。如果有嵌套的括号,则处于最里面的括号最先计算。

表 3-12 运算符的优先级

优先级	运 算 符
1	+(正)、-(负)、~(按位非)
2	*(乘)、/(除)、%(取模)
3	+(加)、+(连接)、-(减)
4	=、>、<、>=、<=、<>、!=、!>、!<(比较运算符)
5	^(按位异或)、&(按位与)、 (按位或)
6	NOT
7	AND
8	ALL、ANY、BETWEEN、IN、LIKE、OR、SOME
9	=(赋值)

### 3.5.5 函数

函数是指具有完成某种特定功能的程序片段,在 Transact-SQL 编程中也可理解为能完成一定功能的 SQL 语句集合,其处理结果称为返回值,处理过程称为函数体。SQL

Server 与其他程序设计语言一样,提供了丰富的内置函数,而且允许用户自定义函数。所以,SQL Server 2016 支持两种函数类型,即内置函数和用户自定义函数,利用这些函数可以方便地实现各种运算和操作。

### 1. SQL Server 2016 提供的内置函数

为了使用户对数据库进行查询和修改时更加方便,SQL Server 2016 提供了丰富的具有执行某些运算功能的内置函数,可以分为 14 大类,如表 3-13 所示。

表 3-13 SQL Server 2016 提供的内置函数的种类和功能

函数种类	主要功能
聚合函数	对一组值进行运算,返回一个汇总值
字符串函数	对字符串输入值进行运算,然后返回一个字符串或数字值
数学函数	对输入值进行数学运算,然后返回数字值
日期时间函数	对日期和时间输入值进行运算,然后返回字符串、数字或日期和时间值
系统函数	执行运算后返回 SQL Server 实例中有关值、对象和设置的信息
行集函数	返回可在 SQL 语句中像表引用一样使用的对象
排名函数	对分区中的每一行均返回一个排名值
文本和图像函数	对文本或图像输入值进行运算,然后返回有关值的信息
配置函数	返回当前配置选项配置的信息
加密函数	支持加密、解密、数字签名和签名验证等操作
游标函数	返回有关游标状态的信息
安全函数	返回有关用户和角色的信息
系统统计函数	返回系统的统计信息
元数据函数	返回有关数据库和数据库对象的信息

这里主要介绍比较常用的几类内置函数,即聚合函数、字符串函数、数学函数、日期时间函数、系统函数以及其他函数。

#### 1) 聚合函数

聚合函数又称为统计函数。所有聚合函数均为确定性函数,只要使用一组特定输入值(数值型)调用聚合函数,该函数就会返回同类型的单个计算结果。聚合函数通常和 SELECT 语句中的 GROUP BY 子句一起使用。SQL Server 提供的聚合函数如表 3-14 所示。

表 3-14 聚合函数

函数	说明
AVG([ALL DISTINCT]表达式)	计算表达式中各项的平均值。其中,ALL 表示对所有值求平均,DISTINCT 表示排除表达式中的重复值项
SUM([ALL DISTINCT]表达式)	计算表达式中所有值项的和,它忽略 NULL 值项
MAX([ALL DISTINCT]表达式)	返回表达式中的最大值项
MIN([ALL DISTINCT]表达式)	返回表达式中的最小值项
COUNT({[ALL DISTINCT]表达式} * )	返回一个集合中的项数,返回值为整型
COUNT_BIG({[ALL DISTINCT]表达式} * )	返回一个集合中的项数,返回值为长整型

续表

函 数	说 明
CHECKSUM_AGG([ALL DISTINCT]表达式)	返回一个集合的校验和
STDEV(表达式)	返回表达式中所有数值的统计标准偏差
STDEVP(表达式)	返回表达式中所有数值的填充统计标准偏差
VAR(表达式)	返回表达式中所有数值的统计方差
VARP(表达式)	返回表达式中所有数值的填充统计方差
GROUPING(表达式)	指示是否聚合 GROUP BY 列表中的指定列表表达式。在结果集中,如果 GROUPING 返回 1,则指示聚合;如果返回 0,则指示不聚合。如果指定了 GROUP BY,则 GROUPING 只能用在 SELECT、HAVING 和 ORDER BY 子句中
BINARY_CHECKSUM(表达式)	返回一个根据表达式遍历表的所有行的二进制校验和值,可用于检测表中行的更改

在所有聚合函数中,除了 COUNT() 函数以外,聚合函数均忽略空值。

## 2) 字符串函数

字符串函数实现对字符串的操作和运算。SQL Server 2016 提供的字符串函数如表 3-15 所示。

表 3-15 字符串函数

函 数	说 明
ASCII(character_expression)	返回字符表达式中最左侧的字符的 ASCII 代码值
CHAR(integer_expression)	将 integer_expression 转换为字符。对于控制字符,可以使用 CHAR() 函数输入。例如,CHAR(9) 表示制表符,CHAR(10) 表示换行符,CHAR(13) 表示回车符
CHARINDEX(character_expression1, character_expression2[ start_location])	返回指定的表达式 character_expression1 在表达式 character_expression2 中的开始位置。其中,参数 start_location 指出在表达式 character_expression2 中开始搜索的起始位置,如 start_location 的值为 0、-1 或者省略,搜索从表达式 character_expression2 的起始位置开始。返回值类型为 int
DIFFERENCE(character_expression1, character_expression2)	比较两个字符串表达式的差异。返回值为 0~4
LEFT(character_expression, integer_expression)	返回字符串表达式 character_expression 中左边的 integer_expression 个字符。返回值类型为 varchar
LEN(string_expression)	返回字符串的长度,并包括字符串尾部的空格。返回值类型为 int
LOWER(character_expression)	将 character_expression 中的所有大写字母转换成小写字母
LTRIM(character_expression)	将 character_expression 中的前导空格删除
NCHAR(integer_expression)	返回 integer_expression 所代表的 Unicode 字符

续表

函 数	说 明
PATINDEX('%pattern%',expression)	返回 expression 中 pattern 首次出现的位置
REPLACE('string_expression1','string_expression2','string_expression3')	将字符串表达式 string_expression1 中所有的 string_expression2 字符串替换为 string_expression3
QUOTENAME('character_string'['quote_character'])	给字符串 character_string 添加上定界符,以构成 SQL Server 中有效的定界标识符
REPLICATE(character_expression,integer_expression)	将 character_expression 重复 integer_expression 次,组成一个字符串
REVERSE(character_expression)	将 character_expression 中的字符逆向排列组成字符串
RIGHT(character_expression,integer_expression)	返回 character_expression 中右边的 integer_expression 个字符
RTRIM(character_expression)	将字符串表达式 character_expression 中的尾部空格删除
SOUNDEX(character_expression)	返回一个四字符代码,说明字符串读音的相似性
SPACE(integer_expression)	返回一个由空格组成的字符串,空格的个数为 integer_expression; 如果 integer_expression 的值为负,返回 NULL
STR(float_expression[,length[,decimal]])	返回由数字数据转换来的字符数据
SUBSTRING(expression,start,length)	返回字符表达式、二进制表达式、文本表达式或图像表达式的一部分。start 是指定子字符串开始位置的整数
UPPER(character_expression)	将 character_expression 中的所有小写字母转换成大写字母

### 3) 数学函数

数学函数对数字表达式进行数学运算并返回运算结果。数学函数可对 SQL Server 系统提供的数字数据 (decimal、integer、float、real、money、smallmoney、smallint 和 tinyint) 进行处理。在默认情况下,对 float 数据类型数据的内置运算的精度为 6 个小数位;传递到数学函数的数字将被解释为 decimal 数据类型;可用 CAST() 或 CONVERT() 函数将数据类型更改为其他数据类型,例如 float 类型。SQL Server 2016 提供的数学函数如表 3-16 所示。

表 3-16 数学函数

函 数	说 明
ABS(numeric_expression)	求 numeric_expression 表达式的绝对值
ACOS(float_expression)	求 float_expression 表达式的反余弦
ASIN(float_expression)	求 float_expression 表达式的反正弦
ATAN(float_expression)	求 float_expression 表达式的反正切
ATN2(float_expression1,float_expression2)	求 float_expression1/float_expression2 的反正切
CEILING(numeric_expression)	求大于等于 numeric_expression 表达式的最小整数
COS(float_expression)	求 float_expression 表达式的余弦
COT(float_expression)	求 float_expression 表达式的余切
DEGREES(numeric_expression)	将弧度 numeric_expression 转换为度
EXP(float_expression)	求 float_expression 表达式的指数

续表

函 数	说 明
FLOOR(numeric_expression)	求小于等于 float_expression 表达式的最大整数
LOG(float_expression)	求 float_expression 表达式的自然对数
LOG10(float_expression)	求 float_expression 表达式以 10 为底的对数
PI()	返回圆周率 $\pi$ , 值为 3.14159265358979
POWER(numeric_expression, y)	求 numeric_expression 的 y 次方
RADIANS(numeric_expression)	将度 numeric_expression 转换为弧度
RAND([seed])	返回 0 到 1 的随机浮点数, 可以用整数 seed 来指定初值
ROUND(numeric_expression, length[, function])	求表达式 numeric_expression 的四舍五入值和截断值, 四舍五入或截断后保留的位数由 length 指定; function 参数说明 ROUND() 函数执行的操作, 数据类型必须是 tinyint, smallint 或 int; 当 function 的数值为 0 或省略时, 执行四舍五入操作, 否则执行截断操作
SIGN(numeric_expression)	求表达式 numeric_expression 的符号值
SIN(float_expression)	求表达式 float_expression 的正弦
SQUARE(float_expression)	求表达式 float_expression 的平方
SQRT(float_expression)	求表达式 float_expression 的平方根
TAN(float_expression)	求表达式 float_expression 的正切

像 ABS()、CEILING()、DEGREES()、FLOOR()、POWER()、RADIANS() 和 SIGN() 函数, 返回值的数据类型和输入值的数据类型相同。而三角函数和其他函数, 包括 EXP()、LOG()、LOG10()、SQUARE() 和 SQRT(), 将输入值的数据类型转换成 float 类型, 并且返回值为 float 类型。

#### 4) 日期时间函数

日期时间函数用于处理输入的日期和时间数值, 并返回一个字符串、数字或者日期时间数值。SQL Server 2016 提供的日期时间函数如表 3-17 所示。

表 3-17 日期时间函数

函 数	说 明
DATEDD(datepart, number, date)	返回 datetime 类型数值, 其值为 date 值加上 datepart 和 number 参数指定的时间间隔
DATEDIFF(datepart, startdate, enddate)	返回 startdate 和 enddate 的时间间隔, 其单位由 datepart 参数决定
DATENAME(datepart, date)	返回 date 参数对应的字符串, 其格式由 datepart 确定
DATEPART(datepart, date)	返回 date 参数对应的整数值, 其格式由 datepart 确定
DAY(date)	返回 date 参数的日数, 返回值数据类型为 int
GETDATE()	按照 SQL Server 规定的格式返回系统当前的日期和时间
GETUTCDATE()	返回 datetime 类型数值, 其值表示当前的格林威治时间
MONTH(date)	返回 date 参数的月份, 返回值数据类型为 int
YEAR(date)	返回 date 参数的年份, 返回值数据类型为 int

在日期时间函数中, datepart 参数指定了时间的单位。在 SQL Server 2016 中, datepart 的取值如表 3-18 所示。

表 3-18 datepart 的取值

datepart 的取值	缩 写	含 义
Year	yy 或 yyyy	年
quarter	qq 或 q	季
Month	mm 或 m	月
dayofyear	dy 或 y	年日期(1~366)
Day	dd 或 d	日
Week	wk 或 ww	周
Hour	hh	时
minute	mi 或 n	分
second	ss 或 s	秒
millisecond	ms	毫秒
dayofweek	dw 或 w	周日期(1~7)

### 5) 系统函数

系统函数返回有关 Microsoft SQL Server 的设置和对象等信息。SQL Server 为 DBA 和用户提供了—系列系统函数。通过调用这些系统函数可以获得有关服务器、用户、数据库状态等的系统信息。例如,HOST\_NAME()返回运行 SQL Server 的计算机的名字,APP\_NAME()返回当前会话的应用程序名称等。更多函数参见联机帮助。

### 6) 元数据函数

元数据函数返回有关数据库和数据库对象的信息,所以元数据函数都具有不确定性。常用的元数据函数如表 3-19 所示。

表 3-19 常用的元数据函数

函 数	说 明
COL_LENGTH(表名,列名)	返回列的定义长度(以字节为单位)
COL_NAME(表标识号,列标识号)	根据指定的对应表标识号和列标识号返回列的名称
DB_ID([数据库名称])	返回数据库标识(ID)号
DB_NAME([数据库的标识号])	返回数据库名称

### 7) 配置函数

配置函数实现返回当前配置选项设置的信息的功能,常用的配置函数如表 3-20 所示。

表 3-20 常用的配置函数

函 数	说 明
@@DBTS()	返回当前数据库的当前 timestamp 数据类型的值
@@LANGUAGE()	返回当前所用语言的名称
@@MAX_CONNECTIONS()	返回 SQL Server 实例允许同时进行的最大用户连接数
@@TEXTSIZE()	返回 SET 语句中的 TEXTSIZE 选项的当前值
@@VERSION()	返回当前的 SQL Server 安装版本、处理器体系结构、生成日期和操作系统

### 8) 系统统计函数

在 SQL Server 2016 中,通常以全局变量的形式来表达系统统计函数,常用的系统统计函数如表 3-21 所示。

表 3-21 常用的系统统计函数

函 数	说 明
@@ CONNECTIONS()	返回 SQL Server 自上次启动以来尝试的连接数
@@CPU_BUSY()	返回 SQL Server 自上次启动后的工作时间
@@IDLE()	返回 SQL Server 自上次启动后的空闲时间
@@PACK_RECEIVED()	返回 SQL Server 自上次启动后从网络读取的输入数据包数
@@TOTAL_READ()	返回 SQL Server 自上次启动后读取磁盘的次数

### 9) 其他常用函数

(1) ISDATE(expression)用于判断指定表达式是否为一个合法的日期。如果输入 expression 是 datetime 或 smalldatetime 数据类型的有效日期或时间值,则返回 1,否则返回 0。

(2) ISNULL(check\_expression, replacement\_value)判断 check\_expression 的值是否为空,如果是,则返回 replacement\_value 的值;如果不是,则返回 check\_expression 的值。

(3) ISNUMERIC(expression)确定表达式是否为有效的数值类型。

(4) PRINT(字符串表达式)向客户端返回用户定义消息。

(5) CAST(expression AS data\_type[(length)])将一种数据类型的表达式转换为另一种数据类型的表达式。

(6) CONVERT(data\_type[(length)], expression[, style])将一种数据类型的表达式转换为另一种数据类型的表达式,但 CONVERT()比 CAST()的功能更加强大。

## 2. SQL Server 2016 的自定义函数

为了扩展性和方便用户,SQL Server 2016 提供了用户自定义函数功能。用户自定义函数是由一个或多个 Transact-SQL 语句组成的子程序,可用于封装代码以便重新使用。自定义函数可以接受 0 个或多个输入参数,其返回值是一个临时表或一个数值。需要特别指出的是,自定义函数不支持输出参数,如果要使用输出参数,可以使用存储过程。

在 SQL Server 中,用 CREATE FUNCTION 语句创建自定义函数,每个完全合法的用户自定义函数名必须唯一。根据函数返回值形式的不同可创建 3 类自定义函数,即标量值自定义函数、内联表值自定义函数和多语句表值自定义函数。

### 1) 标量值自定义函数

标量值自定义函数的返回值是一个确定类型的标量值,其返回值类型为除 text、ntext、image、timestamp 和 table 类型之外的任意类型,即标量值自定义函数返回的是一个数值。

定义标量值自定义函数的语法结构如下:

```
CREATE FUNCTION 函数名称(@参数 1 类型 1,[@参数 2 类型 2,...,@参数 n 类型 n])
RETURNS 返回值类型
[WITH ENCRYPTION]
[AS]
BEGIN
    函数体语句序列
    RETURN 返回值
END
```

WITH 子句指出了创建函数的选项,如果 ENCRYPTION 参数被指定,则创建的函数是被加密的,函数定义的文本将以不可读的形式存储在 syscomments 表中,任何人都不可查看该函数的定义,包括函数的创建者和管理员。

### 2) 内联表值自定义函数

内联表值自定义函数是以表形式返回一个值,也就是说返回一个表的数据。内联表值自定义函数没有 BEGIN...END 语句块中包含的函数体,而是直接使用 RETURNS 子句,其中包含的 SELECT 语句将数据从数据库中筛选出形成一个表。使用内联表值自定义函数可提供参数化的视图功能。

内联表值自定义函数的语法结构如下:

```
CREATE FUNCTION 函数名称(@参数 1 类型 1,[@参数 2 类型 2,...,@参数 n 类型 n])
RETURNS TABLE
[WITH ENCRYPTION]
[AS]
RETURN (查询语句)
```

### 3) 多语句表值自定义函数

多语句表值自定义函数可看作标量值和内联表值自定义函数的结合体。此类函数的返回值是一个表,但与标量值自定义函数一样,有一个 BEGIN...END 语句块中包含的函数体,返回值的表中的数据是由函数体中的语句插入的。因此,其可以进行多次查询,对数据进行多次筛选与合并,弥补了内联表值自定义函数的不足。

同时,可以使用 ALTER FUNCTION 语句修改自定义函数,使用 DROP FUNCTION 语句删除用户自定义函数。其语法参见联机帮助。

## 3.5.6 表达式

表达式是标识符、值和运算符的组合,是指由常量、变量或函数等通过运算符按规则连接起来的有意义的式子。表达式的运算常在“列与列”或者“变量”间进行。在 SQL Server 中,表达式主要分为 4 类,即数学表达式、字符串表达式、比较表达式和逻辑表达式。下面分别对这 4 类表达式进行说明。

### 1. 数学表达式

数学表达式用于各种数字变量的运算。数字变量的类型有 int、smallint、tinyint、

float、real、money 和 smallmoney 等。用于数学表达式的符号主要为算术运算符。

## 2. 字符串表达式

字符串表达式由字母、符号或数字组成。在字符串表达式中,用“+”来实现字符或字符串的连接。在数据类型中,可用于字符串加法的数据类型有 char、varchar、nvarchar、text 和可以转换为 char 或 varchar 的数据类型。

## 3. 比较表达式

比较表达式用于两个表达式的比较,其执行优先级和数学表达式一样,可以用“( )”来人为设置。

## 4. 逻辑表达式

在 SQL Server 的逻辑表达式中有 3 种连接符,即 AND、OR、NOT。

(1) AND 表达式。当所有表达式的值为真时,其逻辑表达式的值才为真;如果有一个返回值为“假”,则表达式的值为“假”。

(2) OR 表达式。只要有一个子表达式的返回值为“真”,则其逻辑表达式的值即为“真”。

(3) NOT 表达式。当表达式的值为“真”时,进行 NOT 运算后,其表达式的值为“假”,反之亦然。

### 3.5.7 注释

注释是指程序代码中不执行的文本字符串,也称为注解。在 Transact-SQL 程序中,注释语句主要用于对程序语句的解释说明并增加可读性,有助于对源程序语句的理解、修改和维护,系统对注释语句不执行。注释通常用于记录程序名称、作者姓名和主要代码更改的日期。注释可用于描述复杂计算或解释编程方法。SQL Server 支持两种类型的注释语句,即单行注释语句和多行注释语句。

#### 1. 单行注释语句

单行注释语句也称为行注释语句,通常放在一行语句的后面,用于对本行语句进行具体说明,是以--(双连字符)开始的若干字符。这些注释字符可与要执行的代码处在同一行,也可另起一行。从双连字符开始到行尾均为注释。对于多行注释,必须在每个注释行的开始使用双连字符。

例如以下 Transact-SQL 语句就包括单行注释语句,注释对 SQL 语句的执行没有任何影响。

```
-- 查询学生信息  
SELECT * FROM Student
```

## 2. 多行注释语句

多行注释语句也称为块注释语句,通常放在程序(块)的前面,用于对程序功能、特性和注意事项等方面进行说明。对于多行注释,必须使用开始注释字符对( /\* )开始注释,使用结束注释字符对( \*/ )结束注释,在注释行上不应出现其他注释字符。这些注释字符可与要执行的代码处在同一行,也可另起一行,甚至在可执行代码内。从开始注释对( /\* )到结束注释对( \*/ )之间的全部内容均视为注释部分。

## 3.6 本章知识点小结

SQL Server 2016 是微软公司具有重要意义的数据库新产品。其作为最新研发的新一代旗舰级数据库和分析平台,突出高级分析和丰富可视化,并融合了关键创新功能。本章首先介绍了 SQL Server 的发展历程,以及 SQL Server 2016 的主要功能及特点;然后简要介绍了 SQL Server 2016 的服务器组件、主要管理工具以及数据库的存储结构,重点介绍了 SQL Server Management Studio 的使用。如果想要使用某个 SQL Server 2016 服务器所提供的资源,首先必须要保证相关的服务已启动,并已经成功登录。SQL Server 2016 支持 4 种身份验证模式,即 Windows 身份验证、SQL Server 身份验证、活动目录密码身份验证、活动目录集成身份验证。Windows 验证模式是使用 Windows 的验证机制;SQL Server 验证模式则是在 Windows 验证的基础上,输入 SQL Server 的用户名和密码的验证模式。Windows 验证模式适用于 Windows 组,它适用于命名管道的 RPC 网络库。SQL Server 验证模式适用于所有的网络库。

结构化查询语言(SQL)具有语言简洁、易学易用、高度非过程化、一体化等特点,是目前广泛使用的数据库标准语言。本章最后介绍了 Transact-SQL 程序设计基础,包括 SQL Server 标识符的命名规范、SQL Server 2016 支持的数据类型、常量、变量、运算符、函数、表达式等,为后续章节的学习打下了良好的基础。利用 Transact-SQL 语言所提供的功能,用户可以方便地进行数据库及其对象的创建、管理和维护工作。

## 3.7 习题

1. 简述 SQL Server 2016 的组件及其功能。
2. SQL Server Management Studio 的功能有哪些?
3. 如何进行服务器的注册?
4. 数据库的存储结构分为哪两种?其含义分别是什么?
5. SQL Server 2016 的系统数据库有哪些?各自的作用是什么?
6. 简述如何利用 SQL 语句增加、修改和删除数据库。
7. Transact-SQL 和 SQL 的关系是什么?
8. 数据定义语言的类型和作用是什么?
9. 数据操纵语言的类型和作用是什么?

10. 数据控制语言的类型和作用是什么？
11. Transact-SQL 的标识符必须遵循哪些原则？
12. Transact-SQL 语言主要由哪几部分组成？各部分的功能是什么？
13. 标识符有哪几种？
14. 标识符的命名和使用规则分别是什么？
15. 什么是常量？什么是变量？它们的种类有哪些？
16. 常量和变量的区别是什么？
17. 什么是局部变量？什么是全局变量？如何标识它们？
18. 什么是函数？SQL Server 提供的常用函数分为哪几类？它们的功能分别是什么？
19. SQL Server 提供了哪些种类的运算符？运算符的优行级是如何排列的？
20. SQL Server 2016 数据库分为哪几种类型？
21. 数据库文件类型有哪些？