

第5章

数据库管理系统SQL Server 2019

学习目标

- 理解 SQL Server 2019 体系结构和数据库引擎的作用；
- 掌握 SQL Server 2019 常用管理工具及其操作；
- 掌握 SQL Server 2019 数据库服务器的配置和连接方法；
- 掌握 SQL Server 2019 组成数据库的各种对象的类型和作用；
- 掌握 SQL Server 2019 提供的系统内置数据类型；
- 熟练使用 SQL Server Management Studio 图形化工具管理数据库和数据库基本对象的方法。

重点：重点掌握表、视图、索引的创建和维护，特别是数据完整性约束的定义。

难点：完整性约束定义的方法。

SQL Server 2019 是 Microsoft 公司 2019 年 11 月发布的新版本(15.0)，是在早期版本的基础上构建，旨在将 SQL Server 发展成一个平台，以提供开发语言、数据类型、本地或云环境以及操作系统选项。此版本的 SQL Server 专注于 Hadoop、Apache Spark 等分布式文件系统之间的数据交换，与大数据的连接，在易用性、可伸缩性、安全性、可靠性等方面的优异性能，能够为企事业单位的信息管理系统提供一个全面、安全、可靠的数据库平台，满足各种类型单位构建网络数据库的需求，使其成为客户构建、管理商业数据库的最佳选择方案之一。因此，SQL Server 2019 数据库管理系统正被越来越多的用户使用，已成为企业级数据库管理系统的主流产品。

SQL Server 2019 作为一款面向企业级应用的关系数据库产品，以其强大的功能、简便的操作和可靠的安全性，赢得了很多用户的认可，在各行各业和各种软件产品中得到了广泛的应用。作为全新的企业级信息平台，SQL Server 2019 不仅延续了现有信息平台的强大能力，支持云技术，而且引入了大数据群集和智能化数据管理等新特性。本章主要介绍 SQL Server 2019 的基础知识和使用方法，包括服务器管理、常用管理工具的使用以及数据库和数据库对象的管理与操作。

5.1 SQL Server 2019 简介

作为世界数据库三大巨头之一的 SQL Server，其高效的数据处理、强大的功能、简易而统一的界面操作受到众多软件厂商和企业的青睐。SQL Server 2019 不仅延续现有数据平台的强大能力，更是一款面向数据云服务的信息平台，实现企业内部与外部的数据集成，实

现私有云与公有云之间数据的扩展与应用的迁移。同时,新版本还引入了大数据集群和智能化数据管理等新特性,以满足不同人群对数据以及信息的需求。针对大数据以及数据仓库,SQL Server 2019 提供从 TB 到数百 TB 全面端到端的解决方案。

5.1.1 SQL Server 2019 的体系结构

SQL Server 2019 功能很多,但总体来说可以分为两大类:实例功能和共享功能模块。

实例功能包含数据库引擎服务、分析服务,其中,数据库引擎服务包含 SQL Server 复制、机器学习服务和语言扩展、全文和语义提取搜索、数据质量服务;共享功能包括数据质量客户端、机器学习服务器(独立)、集成服务、主数据服务等。SQL Server 2019 的体系结构如图 5-1 所示。

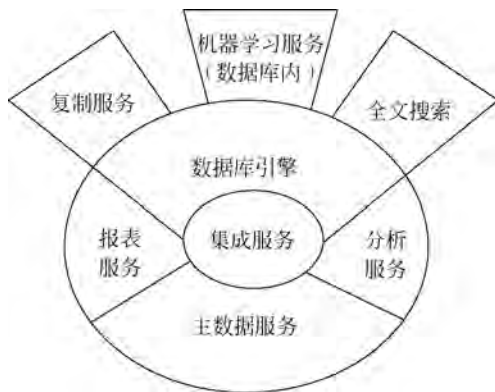


图 5-1 SQL Server 2019 的体系结构

1. 数据库引擎

数据库引擎(SQL server database engine,SSDE)是 SQL Server 2019 用于存储、处理和保护数据的核心服务,也是传统意义上的数据库管理系统。使用数据库引擎可以创建用于联机事务处理或联机分析处理的关系数据库,包括创建用于存储数据的基本表和用于查看、管理和保护数据安全的数据对象。例如,查询数据、创建数据库、创建表、视图及索引和存储过程等操作,都是由数据库引擎完成的。在大多数情况下,使用数据库系统,实际上就是使用数据库引擎。SQL Server 数据库引擎服务包括 SQL Server 复制、全文和语义提取搜索,以及使用关系数据运行 Python 和 R 脚本的机器学习服务等可选功能。

2. 分析服务

分析服务(SQL server analysis services,SSAS)包括一些工具,可用于创建和管理联机分析处理(OLAP)以及数据挖掘应用程序。也就是说,SSAS 为各种商业智能提供联机分析处理和数据挖掘功能,可以支持用户建立数据库,使用分析服务,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,还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,以更有效的方式提供给决策分析者。其中,联机分析处理承载多维数据库,将数据存储在高维数据集中,而数据挖掘提供了分析数据集的方法,用这种方法可以找出数据中的一些非显性模式。例如,在电子商务系统中,可以使用分析服务完成对客户购物的数据挖掘分析,发现隐藏在大量数据中的模式和关系,获取更多有价值的信息,从而使决策者

更加合理地安排不同商品的管理。

3. 报表服务

报表服务(SQL server reporting services, SSRS)是基于服务器的,能为用户提供支持Web方式的企业级报表功能,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。报表服务包含一整套用于创建、管理和传送报表的工具以及允许开发人员在自定义应用程序中集成或扩展数据和报表处理的API,通过图形方式或编程生成,以.rdl文件格式存储在SQL Server的报表服务数据库。报表服务工具在Microsoft Visual Studio环境中工作,并与SQL Server工具和组件完全集成。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。使用报表服务可以从包含关系数据源、多维数据源和基于XML的数据源中获取报表的内容,能用自己需求的不同格式创建数据表格、图形等各种样式的报表,并可以通过Web连接来查看和管理这些报表。

4. 集成服务

集成服务(SQL server integration services, SSIS)是一个用于生成高性能数据集成和工作流解决方案的数据平台,几乎可以在任何类型的数据源之间移动数据,是SQL Server的数据提取—转换—加载(ETL)工具。使用集成服务可以解决复杂的业务问题,具体表现为:管理SQL Server对象和数据,复制或下载文件,发送电子邮件以响应事件,执行FTP执行等。也可以提取和转换来自多种源的数据,如Oracle、XML文档、文本文件等数据源中的数据或者用它来清理、聚合、合并、复制数据。总之,数据转换、收集来自许多不同数据源的数据或搜集可用分析服务进行分析的数据仓库数据,集成服务在这些操作中非常有用。

5. 主数据服务(master data services, MDS)

MDS是针对主数据管理的SQL Server解决方案。可以配置MDS来管理任何领域(产品、客户、账户);MDS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于管理数据的用于Excel的外接程序。

6. 机器学习服务

机器学习服务(数据库内)支持使用企业数据源的分布式、可缩放的机器学习解决方案。机器学习服务器(独立)支持在多个平台上部署分布式、可缩放机器学习解决方案,并可使用多个企业数据源,包括Linux和Hadoop。在SQL Server 2016中,支持R语言。SQL Server 2019(15.x)支持R和Python。

7. 复制服务

SQL Server复制服务通常可用于移动数据。可以将数据和数据库对象从一个数据库复制和分发到另一个数据库,然后,在数据库之间进行同步,以保持它们的一致性。只要有网络,都可以使用复制服务把数据分发到不同的位置,包括移动用户。

8. 全文搜索

SQL Server的全文搜索可以将SQL Server表中纯字符的数据以词或短语的形式执行全文查询。一旦创建了全文搜索功能,SQL Server查询就可以搜索全文搜索索引,并返回高性能字符串的字符索引。全文搜索与SQL中的Like语句不同,它是先将数据库中的文本数据创建索引,然后,根据特定语言的规则对词和短语进行搜索,其速度快,形式灵活,使用方便。

5.1.2 SQL Server 2019 的主要亮点

作为全新的企业级信息平台,SQL Server 2019 不仅延续了现有信息平台的强大能力,支持云技术,而且引入了 大数据群集和智能化数据管理等新特性,还为 SQL Server 数据库引擎、分析服务、SQL Server 机器学习服务、Linux 上的 SQL Server 和主数据服务提供了附加功能和改进。特别是 SQL Server 2019 引入的大数据集群,将为使用大数据、大数据集和 AI 提供良好的支持。

当代企业通常掌管着庞大的数据资产,这些数据资产由托管在整个公司的孤立数据源中的各种不断增长的数据集组成。利用 SQL Server 2019 大数据群集,可以从所有数据中获得近乎实时的分析,该群集提供了一个完整的环境来处理包括机器学习和 AI 功能在内的大量数据。

总之,SQL Server 2019 已成为数据库领域稳定性、可靠性、安全性最高和应用性最广泛的数据库管理系统。其主要亮点如下。

1. 分析所有类型的数据

使用内置有 Apache Spark 的 SQL Server 2019,跨关系、非关系、结构化和非结构化数据进行查询,从所有数据中获取见解,从而全面了解业务情况。

2. 灵活选择语言 and 平台

通过开源支持,可以灵活选择语言和平台。在支持 Kubernetes 的 Linux 容器上或在 Windows 上运行 SQL Server。

3. 依靠行业领先的性能

利用突破性的可扩展性和性能,改善数据库的稳定性并缩短响应时间,而无须更改应用程序。让关键型应用程序、数据仓库和数据湖实现高可用性。

4. 安全性持续领先,值得信赖

该数据库过去九年来被评为漏洞最少的数据库,可实现安全性和合规性目标。可以使用内置功能进行数据分类、数据保护以及监控和警报,实现快人一步。

5. 更快速地做出更好的决策

使用报表服务的企业报告功能在数据中找到问题的答案,并通过随附的 Power BI 报表服务器,使用户可以在任何设备上访问丰富的交互式 Power BI 报表。

5.2 SQL Server 2019 常用管理工具

SQL Server 2019 提供了各种帮助数据库管理员和开发人员提高工作效率的工具,通过这些工具可以完成数据库的配置、管理和开发等任务。因此,在使用 SQL Server 2019 之前,认识各种工具及其特性是非常重要的。

5.2.1 SQL Server Management Studio

SQL Server Management Studio(SSMS)是 SQL Server 2019 中最重要的一个集成环境管理工具,用于访问、配置、控制、管理和开发 SQL Server 的所有组件。它继承了 SQL Server 低版本的操作风格,将早期的 SQL Server 2000 中所包含的企业管理器、查询分析器

和 Analysis Manager 功能整合到同一环境中,使得 SQL Server 中的所有组件协同工作,形成了用于数据库管理的功能丰富的图形工具与脚本编辑器,为开发人员及管理人员提供对 SQL Server 的访问。作为开发和管理 SQL Server 数据库对象的有效工具,SQL Server Management Studio 可以完成对 SQL Server 2019 的管理,例如,管理 SQL Server 服务器,创建和管理数据库,创建与管理表、视图、存储过程、角色、规则等数据库对象以及用户自定义的数据类型,备份数据库等。

下面简要介绍 SQL Server Management Studio 的使用方法。

1. SQL Server Management Studio 的启动

在 SQL Server 2019 中,SQL Server Management Studio 需要单独下载并安装,安装完成后可以通过如下步骤启用 SQL Server Management Studio。

(1) 开始启动。执行“开始”→“程序”→Microsoft SQL Server Tools 18→SQL Server Management Studio 18 命令,打开如图 5-2 所示的“连接到服务器”对话框(由于 SQL Server Management Studio 是客户端工具,通过 SQL Server Management Studio 管理和操作 SQL Server 服务,需要先连接服务器)。



图 5-2 “连接到服务器”对话框

(2) 连接服务器。选择服务器的类型是数据库引擎,服务器的名称是安装运行了数据库服务器的计算机机器名(XHR 是笔者主机的名称)或 IP,该名由系统自动查找并显示。如果安装数据库时使用的不是默认实例,而是命名实例,则服务器名称还要包括实例名。

连接服务器的属性设置。单击图 5-2 的“选项”按钮,可以对要连接的服务器进行属性设置,如网络协议、网络数据包大小、连接超时值选项等,如图 5-3 所示。

(3) 身份验证。如果在安装时配置了 sa 的登录密码,可以使用 SQL Server 身份验证,在用户名中输入 sa,然后输入设置的密码,单击“连接”按钮,连接到指定的 SQL Server 服务器。与服务器连接之后的 SQL Server Management Studio 的集成环境窗口如图 5-4 所示。

2. SQL Server Management Studio 的基本操作

SQL Server Management Studio 管理工具采用微软统一的界面风格,是 SQL Server 2019 图形使用界面的集成管理环境,由一个或多个子窗口组成,可以通过同一个工具来访问、设置、管理和开发 SQL Server 组件。由图 5-4 可以看出,SQL Server Management Studio 管理工具类似 Windows 文件资源管理器,窗口最上面分别是菜单栏和工具栏,左侧



图 5-3 连接服务器的属性设置

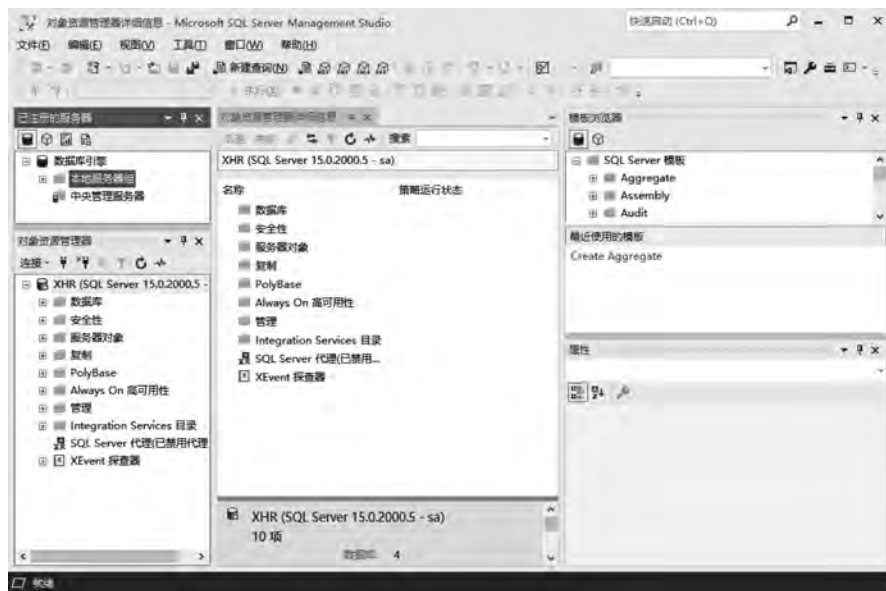


图 5-4 SQL Server Management Studio 集成环境窗口

是已注册的服务器,对象资源管理器窗口,所有已经连接的数据库服务器及其对象以树状结构显示在该窗口中。中间区域是 SSMS 的主要区域,SQL 语句的编写、表的创建、数据表的展示等都在该区域完成。主区域采用选项卡的方式在同一区域实现这些功能。右侧是模板浏览器、属性区域等,模板浏览器提供多项常用操作的模板,属性区域用于查看和修改某对象的属性作用。

SSMS 是一个功能强大而且灵活的工具,由于集成了很多窗口,将所有窗口的属性对话

框都打开的话,就会占用整个屏幕。为了更好地利用屏幕空间,可以关闭、隐藏或移动这些组件对话框。通过菜单栏上的“视图”菜单可以进行定制。

SSMS 的主要窗口介绍如下。

(1) 已注册服务器窗口。已注册服务器窗口用于显示所有已注册的服务器,可以在此添加和删除服务器。

(2) 对象资源管理器窗口。对象资源管理器是服务器中所有数据库对象的树形结构视图,用于管理服务器的相关对象项目,包括数据库、安全性、服务器对象、复制等。用户可以通过该窗口操作数据库,如创建、修改、删除数据库、表、视图等数据库对象,创建登录用户和授权,进行数据库的备份和恢复操作。

(3) 模板资源管理器。SQL Server 为了便于用户使用,提供了多项常用操作的模板。如数据库创建、数据库备份等,这些模板都集中在“模板资源管理器”中。用户可以根据需要选择对应的模板,再修改模板提供的代码来完成所需的操作。如双击某个模板就会自动打开查询分析器,显示模板对应操作的代码。

(4) 查询编辑器窗口。SSMS 集成了用于编写 Transact-SQL(T-SQL)查询语句的窗口查询编辑器。查询编辑器窗口是一个提供了图形界面的查询管理工具,它与 SQL Server 2000 中的查询分析器类似,是数据库管理员或开发人员执行 T-SQL 语句的工具。在开发和维护应用系统时,查询编辑器窗口是最常用的工具之一。其具体启动过程如下:

在图 5-4 所示窗口的工具栏中单击“新建查询”按钮,在 SSMS 主窗口的右边产生一个新的查询编辑器代码窗口,如图 5-5 所示。其中右上方是 SQL 代码区域,用于输入 SQL 的语句;右下方是结果区域,也称结果窗格,用于显示结果和分析结果。多次单击“新建查询”按钮,将会出现多个查询编辑器代码窗口,单击“查询编辑器代码”窗口上的选项卡可以选择不同的查询编辑器代码窗口来编辑 T-SQL 语句。

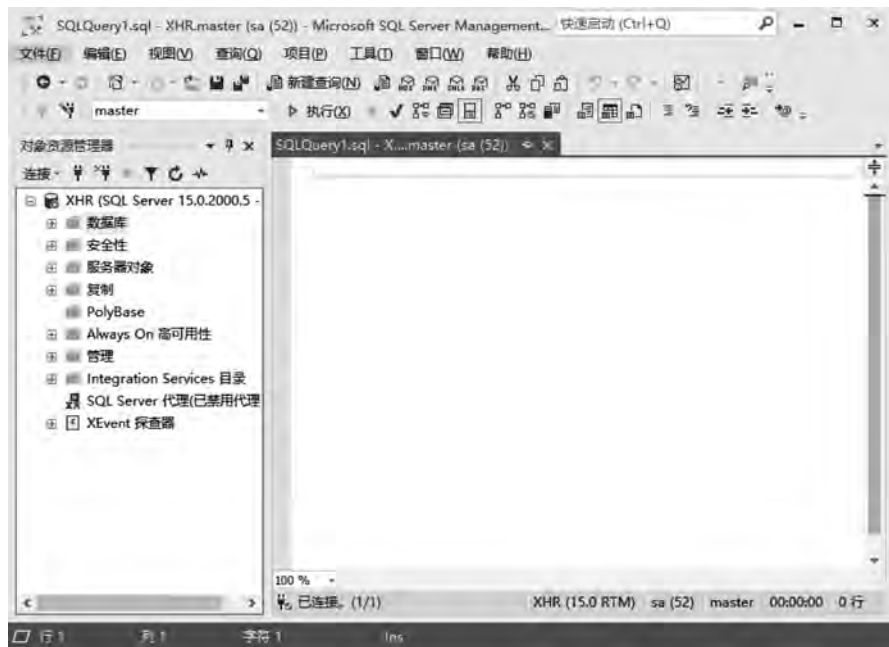


图 5-5 查询编辑器代码窗口

要使用查询编辑器代码窗口,用户必须掌握 T-SQL 语句。在查询编辑器代码窗口中输入 T-SQL 语句,输入完毕后,单击“SQL 查询编辑器”工具栏上的“分析”按钮 ,检查输入 T-SQL 语句是否有语法错误,如果有语法错误,则进行修改;如果语句分析正确,则单击工具栏的“执行”按钮 ,可以执行该 T-SQL 语句,并打开查询“结果窗格”,显示结果。

对于 T-SQL 的执行结果,在“结果窗格”中可以通过执行菜单“查询”→“将结果保存到”命令,选择有不同的输出形式,常用的输出结果的形式有以文本形式显示结果、以网格形式显示结果,将结果保存到文件。

注意: SSMS 中各窗口和工具栏的位置并不是固定的,用户可以根据自己的喜好将窗口拖动到主窗体的任何位置,甚至悬浮脱离主窗体。

5.2.2 SQL Server 2019 的配置工具

SQL Server 2019 的配置工具包括 SQL Server Configuration Manager、Reporting Services 配置、Notification Services 命令提示、SQL Server 错误和使用情况报告。

1. SQL Server Configuration Manager

SQL Server Configuration Manager(SQL Server 配置管理器)是一个管理工具,用于管理 SQL Server 的服务,网络配置和客户端配置,如 SQL Server 客户端计算机连接服务器端计算机的连接配置。SQL Server Configuration Manager 实际上是将 SQL Server 2000 中的“服务管理器”“服务器网络实用工具”和“客户端网络实用工具”三个工具集成在了同一个工具中,可以完成三个工具所能完成的工作。

完成了 SQL Server 2019 的安装后,首要的问题是配置 SQL Server 2019。执行“开始”→“程序”→Microsoft SQL Server 2019→“Microsoft SQL Server 2019 配置管理器”命令,启动 SQL Server 配置管理器(SQL Server Configuration Manager),如图 5-6 所示。SQL Server 在服务器后台要运行许多不同的服务。SQL Server 配置管理器中可配置的项目包括 SQL Server 服务、SQL Server 网络配置、SQL Native Client 11.0 配置等项。



图 5-6 SQL Server“配置管理器”窗口

1) SQL Server 2019 的服务配置

SQL Server 服务可以对 SQL Server 2019 提供的各项服务进行管理,包括启动、停止、暂停等。配置 SQL Server 2019 服务的步骤如下。

(1) 在图 5-6 所示的 SQL Server 配置管理器的左窗格中,单击“SQL Server 服务”结点,如图 5-7 所示。在右窗格中以列表的形式展示当前计算机中所安装的可配置的 SQL

Server 2019 服务项目及服务状态、启动模式、登录身份、进程 ID 和服务类型。列表中的服务项目与安装时所选择的功能项目相对应。



图 5-7 SQL Server 2019 服务

(2) 右击需要设置的服务项,在弹出的快捷菜单中选择“属性”命令,如图 5-8 所示。打开服务项的属性对话框,在该对话框中可以修改服务的登录身份、启动模式和其他高级选项,其中“登录”选项卡如图 5-9 所示。



图 5-8 选择“属性”命令



图 5-9 “登录”选项卡

2) SQL Server 2019 的网络配置

SQL Server 2019 的网络配置可以用来配置服务器端的网络协议和连接选项。SQL Server 服务允许通过多种网络协议来响应客户端的请求。

SQL Server 2019 网络配置的步骤如下。

(1) 在图 5-6 所示的配置管理器中,单击左窗格的“SQL Server 网络配置”,将其展开可以看到“MSSQLSERVER 的协议”结点,窗口右边列出了当前实例所应用的协议及其运行状态,如图 5-10 所示。



图 5-10 配置 SQL Server 协议

(2) 如果要启用某一项网络协议,可以右击该协议,在弹出的快捷菜单中可以启用或禁用该协议,配置该协议的属性。如果要对选中的网络协议进行设置,可右击该网络协议,在弹出的菜单中选择“属性”命令,在打开的协议属性对话框中进行设置,其中,TCP/IP 是应用最广的协议,TCP/IP 属性对话框如图 5-11 所示。



图 5-11 TCP/IP 属性配置

3) SQL Native Client 配置

SQL Native Client 配置与 SQL Server 2019 网络配置相似,不同的是,该工具配置的是客户端连接服务时的协议,并且可以配置协议的顺序,创建客户端的别名。

在图 5-2 所示的配置管理器中,单击左窗格的“SQL Native Client 的网络配置”,展开后可以配置 SQL Server 2019 客户端协议,如启用、设置协议顺序等及根据协议设置一个预定义的客户端和服务端之间连接的别名。

2. SQL Server 错误和使用情况报告

通过设置 SQL Server 错误和使用情况报告,可以将错误报告发送到微软公司错误报告服务器。SQL Server 2019 的错误和使用情况报告的方式有两种:一是将 SQL Server 2019 的所有组件和实例的错误报告发送到 Microsoft 公司或错误报告服务器;二是将 SQL Server 2019 的所有组件和实例的功能使用情况报告发送到 Microsoft。Microsoft 公司希望可以收集到错误信息和使用情况,以便改进 SQL Server。

启动 SQL Server 2019 的错误和使用情况报告的操作步骤如下。

(1) 执行“开始”→“程序”→Microsoft SQL Server 2019→“SQL Server 错误和使用情况报告”命令,打开“错误和使用情况报告设置”窗口。

(2) 单击“选项”按钮,在窗口下方会出现组件和实例列表,用户可以根据需要选择是否使用情况报告或错误报告。选择想要的发送方式,单击“确定”按钮完成设置。

3. Reporting Services 配置

Reporting Services 配置也就是报表服务器配置,其作用是配置和管理 SQL Server 2019 的报表服务器。

5.2.3 SQL Server Profiler

SQL Server Profiler 是用于从服务器上跟踪和记录 SQL Server 2019 事件的工具。能够通过监控数据库引擎实例或 Analysis Services 实例的运行状态,来识别影响性能的事件,提高系统运行的可靠性。捕获后的事件保存在一个跟踪文件中,通过事件探查器来创建管理事件跟踪文件,根据这个跟踪文件,可以分析有问题的查询并找到问题的所在,查找导致 SQL Server 运行缓慢的查询,捕获导致某个问题的 T-SQL 语句,以及监视 SQL Server 的性能。

启动 SQL Server Profiler 的方法如下。

方法一:在 SSMS 窗口,选择菜单命令“工具”→SQL Server Profiler 命令。

方法二:执行“开始”→“程序”→Microsoft SQL Server Tools 18→SQL Server Profiler 18 命令。

创建跟踪的操作方法步骤如下。

(1) 打开 SQL Server Profiler,在菜单栏中执行“文件”→“新建跟踪”命令。

(2) 打开“连接到服务器”对话框,连接到服务器后,打开“跟踪属性”窗口。“跟踪属性”窗口有两个选项卡,其中“常规”选项卡可以设置跟踪名称、使用模板、保存到文件的地址和名称等属性,如图 5-12 所示;“事件选择”选项卡可以设置跟踪的事件和事件列,对每个事件,可以选择需要监视的信息,如计算机名、用户名、命令文本、CPU 的使用情况等,如图 5-13 所示。

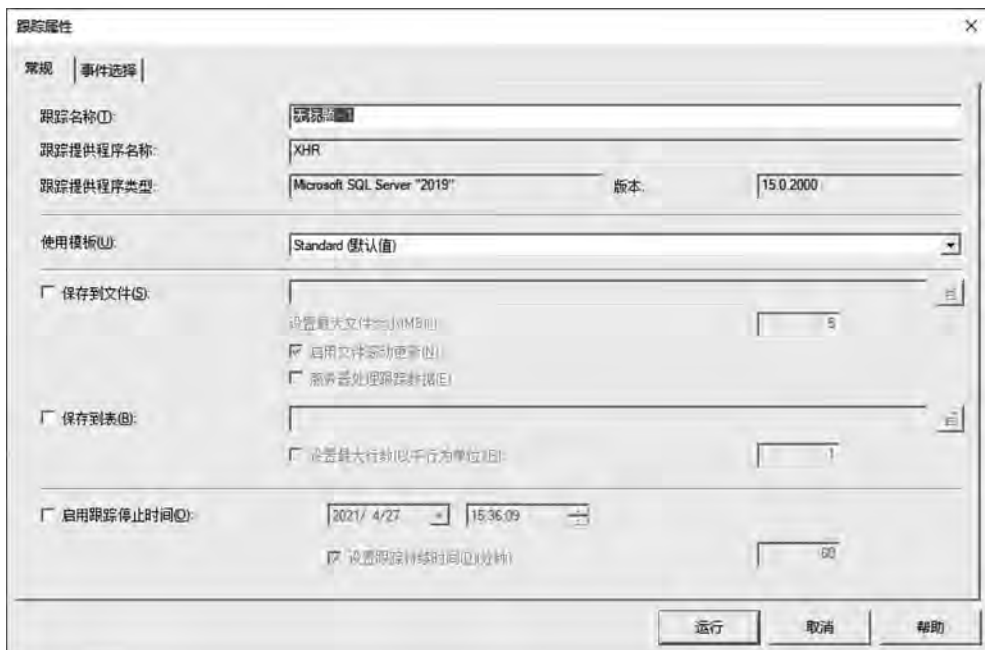


图 5-12 “跟踪属性”的“常规”选项卡



图 5-13 “跟踪属性”的“事件选择”选项卡

(3) 选择完毕后,单击“运行”按钮,启动跟踪事件的变化情况,并在跟踪窗口中显示出来,可以根据结果分析出现问题的原因。

5.2.4 数据库引擎优化顾问

数据库引擎是用于存储、处理和保护数据的核心服务。数据库引擎优化顾问是对 SQL Server 服务器应用过程中承受的工作负荷进行分析、提出优化方案的工具。数据库引擎优化顾问可以让数据库管理员不必精通数据结构、T-SQL,也可以完成对数据库的优化。

执行“开始”→“程序”→Microsoft SQL Server Tools 18→“数据库引擎优化顾问”命令,打开“数据库引擎优化顾问”程序,与 SQL Server 服务器建立连接后,则会出现“数据库引擎优化顾问”窗口,如图 5-14 所示。在该窗口中,可以设置会话的名称、工作负荷所用的文件或表,选择要分析的数据库和表,然后单击“开始分析”按钮,进行分析。分析完毕后可以看到 SQL Server 2019 给出的优化建议及优化报告。

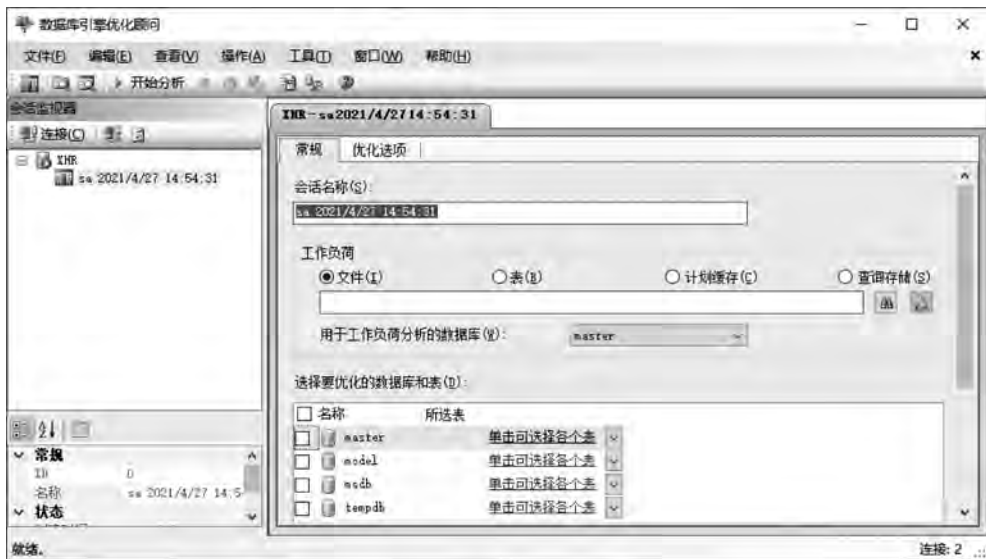


图 5-14 “数据库引擎优化顾问”窗口

5.3 SQL Server 2019 服务器的管理

SQL Server 2019 是服务器级的系统软件,它以服务形式响应客户端数据处理请求,对外提供数据存储、维护和管理等各种服务。通过 SQL Server Management Studio 工具,可以将本地或远程的 SQL Server 服务器注册到本地 SQL Server Management Studio 中,对 SQL Server 服务器和服务器中的资源进行管理。

5.3.1 服务器组的创建与删除

在多 SQL Server 服务器实例的应用环境中,可以根据管理的实际需要,将大量的服务器按照不同的用途和类型组织在几个易于管理的组中,以提高管理的效率。

1. 创建服务器组

创建服务器组可以将众多的已注册的服务器进行分组化管理。而通过注册服务器,可

以存取服务器连接信息,以供在连接服务器时使用。

可以在 SSMS 中创建服务器组,并将服务器放在该服务器组中。创建服务器组的操作步骤如下。

(1) 执行“开始”→“程序”→Microsoft SQL Server Tools 18→SQL Server Management Studio 18 命令。

(2) 打开如图 5-2 所示“连接到服务器”对话框,单击对话框的“取消”按钮,打开如图 5-15 所示的 SSMS 管理工具。

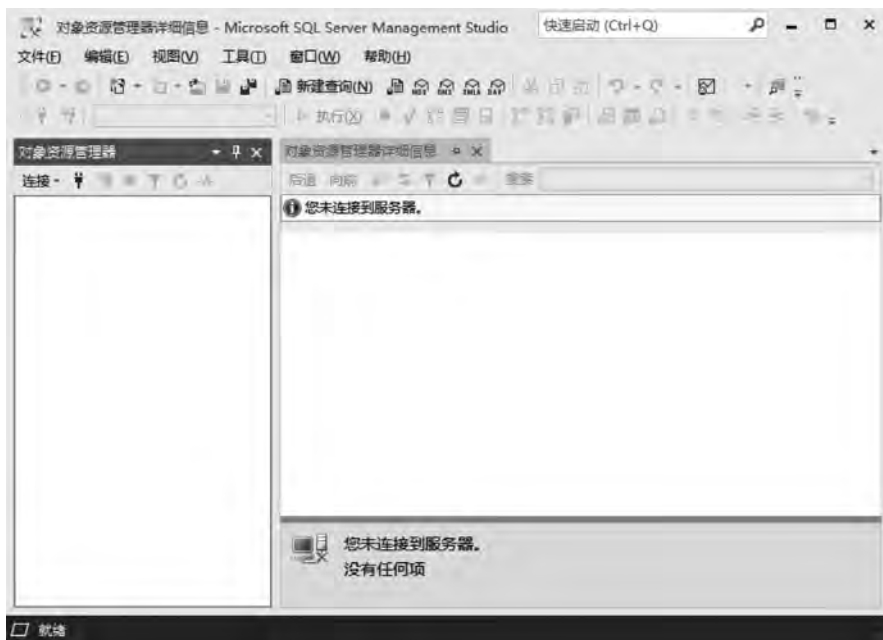


图 5-15 SQL Server Management Studio 管理工具

(3) 执行 SSMS 中的菜单“视图”→“已注册的服务器”命令,则在 SSMS 管理工具中出现“已注册的服务器”窗口,在“已注册的服务器”窗口中选择相应的服务器类型“数据库引擎”→“本地服务器组”,右击“本地服务器组”,在快捷菜单中选择“新建服务器组属性”命令,弹出“新建服务器组属性”属性对话框。

(4) 依次输入新建的服务器组名称 Group1,如图 5-16 所示,输入服务器组的描述信息。



图 5-16 “新建服务器组属性”对话框

(5) 单击“确定”按钮,成功地创建了一个服务器组 Group1,该服务器组没有数据库服务器,结果如图 5-17 所示。



图 5-17 新建的 Group1 服务器组

2. 删除服务器组

删除服务器组的具体操作步骤如下。

(1) 按照创建服务器组时打开的“已注册服务器”窗口的步骤打开“已注册服务器”窗口。

(2) 选择需要删除的服务器组,在弹出的菜单中选择“删除”命令。

(3) 弹出“确认删除”对话框,单击“是”按钮即可完成服务器组的删除。

注意: 在删除服务器组的同时,也会将该组内所注册的服务器一同删除。

5.3.2 服务器的注册、删除和连接

注册服务器是为 SQL Server 客户/服务器确定一台数据库所在的机器,该机器作为服务器,可以响应客户端的各种请求。

1. 注册服务器

SQL Server 2019 可以管理多个不同的服务器实例,为了让 SQL Server 管理工具实现对后台数据库的管理,必须对需要进行管理的本地或远程服务器进行注册。在注册服务器时必须指定服务器名称、登录到服务器时使用的安全类型,注册服务器后如果建立了服务器分组方案,可以将该服务器加入对应的服务器组中。

注册服务器就是将服务器实例的信息添加并保存在 SQL Server Management Studio 中。注册服务器的具体操作步骤如下。

(1) 按照创建服务器组时打开的“已注册服务器”窗口的步骤打开“已注册服务器”窗口。

(2) 右击“已注册服务器”窗口的空白处或右击新建的服务器组 Group1,在弹出的快捷菜单中选择“新建”→“服务器注册”命令。

(3) 打开“新建服务器注册”对话框,该对话框中有“常规”“连接属性”、Always Encrypted 和“其他连接参数”四个选项卡。

“常规”选项卡用于设置服务器类型、名称、登录时身份验证方式,已注册的服务器名称等信息,如图 5-18 所示。

“连接属性”选项卡用于设置所要连接服务器中的数据库、使用的网络协议、连接时等待建立连接的时间、连接后等待任务执行的时间等信息。

Always Encrypted 设置。Always Encrypted 用于保护敏感数据的功能。启用 Always Encrypted 可以允许客户端应用程序对敏感数据进行加密和解密,而 SQL Server 服务器端数据库内只能看到加密后的数据,从而有效实现了敏感数据与高权限之间的隔离。



图 5-18 “常规”选项卡

其他连接参数。除了对前三个选项卡进行设置外,如果还需要设置更多连接参数时,可以在“其他连接参数”选项卡中输入需要的参数。

设置完成后,单击“测试”按钮测试是否与所注册的服务器连接,如果成功连接,会弹出“连接测试成功”提示信息的对话框。

单击“确定”按钮。返回“新建服务器注册”对话框,单击“保存”按钮,确定注册,在 SQL Server Management Studio 窗口中会出现新注册成功的服务器图标。

2. 删除服务器

删除服务器具体操作步骤如下。

(1) 在“已注册服务器”窗格中选择需要删除的服务器,在弹出的菜单中选择“删除”命令。

(2) 在弹出的“确认删除”对话框中单击“是”按钮,即可完成服务器的删除。

3. 服务器的连接

在对象资源管理器中,单击工具栏的“连接”下拉按钮,在下拉菜单中选择要连接的服务器类型(如数据库引擎),如图 5-19 所示,打开“连接到服务器”窗口,根据要连接的服务器在注册时设置的信息,正确选择服务器类型、名称和身份验证模式。单击“连接”按钮,连接成功后,在 SQL Server Management Studio 窗口中会出现所连接的数据库服务器上的各个数

数据库实例及各自的数据库对象。



图 5-19 选择连接的服务器类型

4. 服务器选项的配置

为了确保 SQL Server 服务拥有足够的资源,调整 SQL Server 服务的运行行为,进而取得整体性能的优化,就需要对 SQL Server 所需资源进行配置。

通过 SQL Server Management Studio 中配置服务器选项,具体操作步骤如下。

在“对象资源管理器”窗口中,右击需要配置的服务器,在快捷菜单中选择“属性”命令,在如图 5-20 所示的“服务器属性”窗口中完成各项配置。各选项包括常规项配置、内存项、处理器、安全性、连接等。



图 5-20 “服务器属性”窗口

常规项配置：由系统安装时设置或由服务器的硬件、操作系统类型决定。通过常规选项卡,可以了解当前服务器的基本情况。

内存项配置：可以设置 SQL Server 使用的内存。包括最小服务内存、最大服务内存、创建索引占用的内存、每次查询占用的最小内存等。

处理器：为 SQL Server 分配处理器资源。

安全性：用于设置与 SQL Server 服务登录身份验证等相关的安全性设置。

5.3.3 启动和关闭 SQL Server 2019 服务器

通常情况下,SQL Server 服务器被设置为自动启动模式,在系统启动后,会以 Windows 后台服务的形式自动运行。但某些服务器的配置被更改后必须重新启动才会生效,此时,需要数据库管理员先关闭服务器,再重新启动服务器。SQL Server 2019 的服务器可以通过 SQL Server Management Studio、SQL Server Configuration Manager 管理工具和后台三种方式进行启动或关闭。

1. 在 SSMS 中关闭、启动服务器

在成功连接到 SQL Server 2019 数据库服务器后,打开 SQL Server Management Studio 窗口,可以对服务进行各种管理。

1) 关闭服务器

在“对象资源管理器”窗格中,右击需要关闭的服务器,在弹出的菜单中选择“停止”命令,出现“是否确实要停止服务器”的提示信息,单击“是”按钮,即可关闭选中的服务器,并停止相应的服务。服务器关闭后,其左侧的图标带有红色矩形的停止符号.

2) 启动服务器

启动服务器的操作与关闭服务类似,右击需要启动的服务器后,弹出的快捷菜单中选择“启动”命令,即可启动选中的服务器。服务器启动后,其左侧的图标带有绿色箭头的运行符号.

2. 在 SQL Server Configuration Manager 中关闭和启动服务器

利用 SQL Server Configuration Manager 启动和关闭服务器其操作类似于 SQL Server 2019 的服务配置,按照 SQL Server 2019 服务配置的步骤,打开“SQL Server 2019 服务”结点,在右侧窗格中右击需要关闭的服务器,在弹出的快捷菜单中(见图 5-8)选择“停止”命令,即可关闭选中的服务器。要启动服务器,只要右击需要启动的服务器,在弹出的快捷菜单中选择“启动”命令即可。

3. 通过后台启动和关闭服务器

后台关闭和启动服务器是指用户可以通过“控制面板”进行启动或关闭 SQL Server 服务。其操作步骤是,执行“开始”→“控制面板”,在打开的“控制面板”窗口中双击“管理工具”选项,打开的“管理工具”窗口,在“管理工具”窗口中,双击“服务”选项,打开的“服务”窗口,找到并右击需要启动或关闭的 SQL Server 2019 服务,在弹出的快捷菜单中选择“启动”或“停止”命令,即可启动或关闭服务器。

5.4 SQL Server 2019 数据库的创建与管理

数据库是 SQL Server 2019 最基本的操作对象之一,数据库在运行过程中,能否及时、

准确地为各个应用程序提供所需的数据,关系到系统的性能。数据库的创建、删除、修改、查看、收缩、分离和附加是 SQL Server 2019 的最基本操作,是进行数据库管理与开发的基础。

本节主要介绍 SQL Server 数据库与架构,数据库的组成和存储,数据库的创建与数据库的管理等。

5.4.1 SQL Server 数据库与架构

SQL Server 2019 中的数据库可分为系统数据库、示例数据库和用户数据库。示例数据库是系统为了让用户学习和理解 SQL Server 2019 而设计的。SQL Server 2019 的示例数据库仍然是 Adventure Works。但安装包中并没有提供示例数据库,如果需要使用,则需在安装完 SQL Server 后安装示例数据库。用户数据库是用户根据事务管理需求创建的数据库,如图书管理数据库、商品销售数据库等,而系统数据库是 SQL Server 内置的,主要用于系统管理,是在安装系统 SQL Server 2019 时自动安装的。

1. 系统数据库

SQL Server 2019 中主要包括 master、model、tempdb 和 msdb 四个系统数据库。

1) master 数据库

master 数据库由一些系统表组成,这些系统表负责跟踪整个数据库系统安装和随后创建的其他数据库,对其他的数据库实施管理和控制。作为 SQL Server 2019 中最重要的系统数据库,它是整个数据库服务器的核心功能,同时记录了 SQL Server 中所有系统级的信息,如 SQL Server 的初始化信息、所有的登录账户信息、所有的系统配置设置信息以及用户数据库信息。

如果在计算机上安装了 SQL Server 2019,那么,系统首先会建立一个 master 数据库来记录系统的有关登录账户、系统配置、数据库文件等初始化信息,如果用户在这个 SQL Server 系统中建立一个用户数据库,系统马上将用户数据库的有关用户管理、文件配置、数据库属性等信息写入 master 数据库。系统根据 master 数据库中的信息来管理系统和其他数据库,如果 master 数据库信息被破坏,则 SQL Server 将无法启动。由于 master 数据库对系统来说至关重要,所以,随时都应该保存一个当前环境的备份。

2) model 数据库

model(模板)数据库是 SQL Server DBMS 为用户创建数据库提供的模板,它包含了用户数据库中应该包含的所有系统表,即新建的数据库中的所有内容都是从模板数据库中复制过来。当用户创建数据库时,系统会自动地把 model 数据库中的内容复制到新建的用户数据库中。如果 model 数据库被修改了,那么以后创建的所有数据库都将继承这些修改。

因此,利用 model 数据库的模板特性,通过更改 model 数据库的设置,并将经常使用的数据库对象复制到 model 数据库中,可以简化数据库及其对象的创建、设置工作,为用户节省大量的时间。

3) tempdb 数据库

tempdb 数据库是一个临时数据库,用于保存所有的临时表、临时数据以及临时创建的存储过程。使用 SQL Server 系统时,经常会产生一些临时表和临时数据库对象等。例如,用户在数据库中修改表的某一行数据时,在修改数据库这一事务没有被提交的情况下,系统内就会有该数据的新、旧版本之分,修改后的数据表往往构成了临时表,所以,系统要提供一

个空间来存储这些临时对象,这就是 tempdb 数据库。

因为 tempdb 数据库中记录的信息都是临时的,每当连接断开时,所有临时表和存储过程都将自动丢弃,所以每次启动时 tempdb 数据库里都是空的,上一次的临时数据库都被清除掉了,需要重新创建。默认情况下,SQL Server 在运行时 tempdb 数据库会根据需要自动增长。但是,与其他数据库不同,每次启动数据库引擎时,它会重置初始大小。

4) msdb 数据库

msdb 数据库是代理服务数据库,通常由 SQL Server 代理用来管理警报和作业。当多个用户在使用一个数据库时,经常会出现多个用户对同一个数据的修改而造成数据不一致的现象,或是用户对某些数据和对象的非法操作等。为了防止上述现象,SQL Server 提供了一套代理程序,代理程序能够按照系统管理员的设定监控上述现象,及时向系统管理员发出警报。当代理程序调度警报作业、记录操作时,系统要用到或实时产生许多相关信息,这些信息一般存储在 msdb 数据库中。

2. 常用的数据库对象

SQL Server 数据库中的数据在逻辑上被组织成一系列对象,当用户连接到数据库后,该用户所看到的是逻辑对象,而不是物理的数据库文件。数据库对象指具体数据库管理的内容,也就是存储、管理和使用的不同结构形式,包括数据库关系图、表、视图、同义词、可编程性、代理服务、存储和安全性八类。如图 5-21 所示的窗口的对象资源管理器中,可以看到 SQL Server 将服务器的数据库组织成一个树形逻辑结构,在该结构中有若干结点,每个结点又包括很多子结点,它们代表与该特定数据库有关的不同类型的对象。

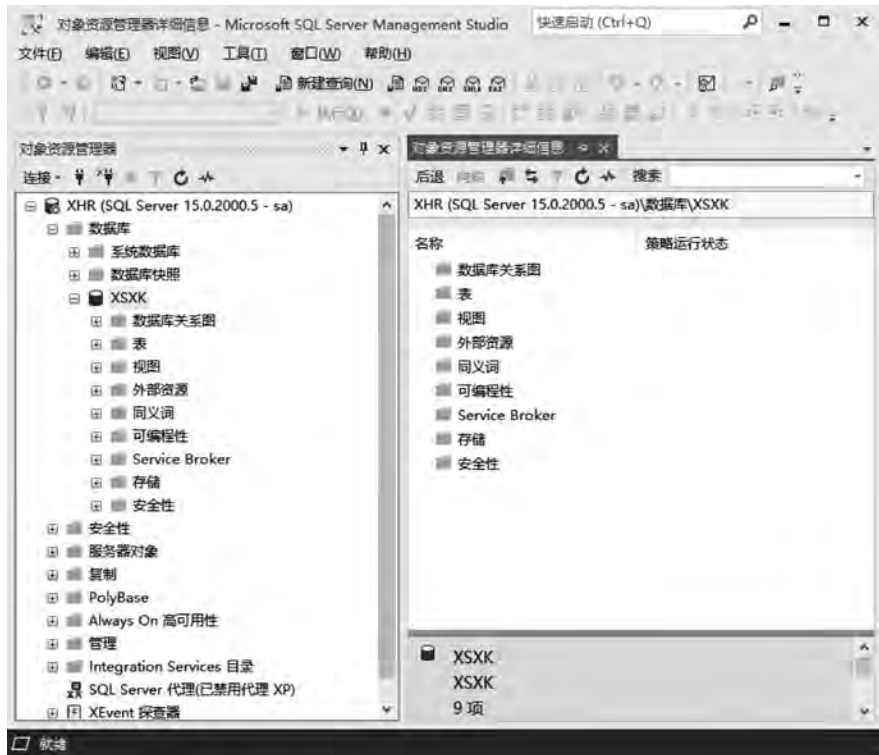


图 5-21 SQL Server 2019 的数据库对象

3. 数据库对象的架构

架构(schema)是 SQL-99 规范的概念,可以用来群组数据库对象,如表、视图等,类似于.NET Framework 的命名空间。架构的作用是将多个数据库对象归属到架构中,以解决用户与数据库对象之间因从属关系而引起的管理问题。SQL Server 架构通常使用在大型数据库的多个表,除了可以有效分类表外,还可以避免名称重复的问题。

从 SQL Server 2005 起,架构不再等效于数据库用户,现在,每个架构都是独立于创建该数据库用户而存在,架构与用户的分离方便了数据库的管理,架构与数据库用户是不同的命名空间。在 SQL Server 2005 及以后的版本中,架构既是一个容器,又是一个命名空间,是一种允许用户对数据库对象进行分类的容器对象,是形成单个命名空间的数据库对象的集合。命名空间是一个集合,其中的每个元素的名称都是唯一的。架构也类似于文件系统中的文件夹,作为容器可以保存和放置下层对象。因此,在同一架构中不能有相同类型、相同名称的数据库对象。例如,为了避免名称冲突,同一架构中不能有两个同名的表,两个表只有在位于不同的架构中时才可以同名。

架构对如何引用数据库对象具有很大的影响,在 SQL Server 2019 中,数据库对象除了在命名时需要遵循命名规则之外,在引用时,同样需要遵循引用规则。一个数据库对象通过四个命名部分组成的结构来引用,引用数据库的语法格式如下所示。

<服务器名>.<数据库名>.<架构名>.<数据库对象名>

上述名称使用句号运算符“.”来连接。由此可以看出,架构是指包含表、视图、存储过程等数据库对象的容器。从包含关系上来讲,架构从属于数据库的内部,而数据库从属于 SQL Server 服务器内部。这些实体就像嵌套框放置在一起。架构下面可以包含很多安全对象,但不能包括其他架构。

在 SQL Server 2019 中,系统默认的架构是 dbo(database owner),如果在创建数据库对象时没有指定架构,那么,默认的数据库对象放在 dbo 架构中。如果是访问默认架构中的对象则可以忽略架构名称,否则在访问表、视图等对象时需要指定架构名称。例如,引用服务器 XHR 上的图书管理(books)数据库中的图书表(book)时,完整的引用为 XHR.books.dbo.book。虽然 SQL Server 对数据库对象的引用包括四个部分,但在实际运用中,在能够区分对象的前提下,前三个部分是可以根据情况省略的,往往可以将其简写。当要访问的数据库对象在当前数据库上,则可以省略服务器名和数据库名,但要指定架构名;当要访问的数据库对象与正在使用的数据库对象有相同的架构时,则可以只写数据库对象名。如 dbo.图书表,即省略了服务器和数据库名。

通过对架构安全对象进行管理,可以提高 SQL Server 的安全性。SQL Server 2019 的架构管理详见本书 8.1.4 节内容。

5.4.2 SQL Server 2019 的数据库基本结构

数据库是 SQL Server 服务器管理的基本单位,在 SQL Server 环境下,如何使用数据库表示和管理数据?数据库在磁盘上是以文件为单位存储的,SQL Server 2019 将数据库映射为一组操作系统文件。

1. 数据库的组成

SQL Server 2019 数据库主要由文件和文件组组成,数据库中的所有数据和数据库对

象都以文件的形式存储在磁盘中。

1) 数据库文件

数据库文件指数据库中用来存放数据库数据和数据库对象的文件,在 SQL Server 2019 系统中,一个数据库在磁盘上可以保存为一个或多个数据库文件,一个数据库文件只能属于一个数据库。当有多个数据库文件时,有一个文件被定义为主数据文件。数据和日志信息分别存储在不同的文件中,而且每个数据库都拥有自己的数据和日志信息文件。SQL Server 数据库文件根据其作用的不同,可以分为主数据文件、次数据文件、事务日志文件三种类型。

(1) 主数据文件(primary file):主数据文件是数据库的起点,指向数据库文件的其他部分。主数据文件是用来存放数据和数据库的初始化(启动)信息和部分或全部数据,是 SQL Server 数据库的主体,它是每个数据库不可缺少的部分,每个数据库有且仅有一个主数据文件,用户数据和对象也可以存储在此文件中,主数据文件的文件扩展名为 .mdf。

(2) 次数据文件(secondary file):次数据文件用来存储主数据文件没有存储的其他数据和对象。如果数据库中的数据量很大,主数据文件不能满足数据存储的需求,需要增加次数据文件,以保存用户数据;如果主数据文件足够大,能够容纳数据库中的所有数据,则该数据库不需要次数据文件。使用次数据文件是因为数据量太过庞大,可以将数据分散存储在多个不同磁盘上以方便进行管理、提高读取速度。每个数据库可以有多个次数据文件,次数据文件的扩展名为 .ndf。

(3) 事务日志文件(transaction log file):用来记录数据库更新情况的文件,SQL Server 2019 具有事务功能,可以保证数据库操作的一致性和完整性,用事务日志文件来记录所有事务及每个事务对数据库进行的插入、删除和更新操作。事务日志是数据库的重要组件,如果数据库遭到破坏,可以根据事务日志文件分析出错的原因;如果数据丢失,可以使用事务日志恢复数据库内容。每个数据库至少拥有一个事务日志文件,也可以拥有多个日志文件。事务日志文件的文件扩展名为 .ldf。

在建立数据库时,需要注意以下两点。

(1) SQL Server 2019 不强制使用 .mdf、.ndf、.ldf 文件扩展名,但使用这些扩展名可以帮助标识文件的用途。

(2) SQL Server 2019 的每个数据库文件都有逻辑文件名和物理文件名两种名称。物理文件名是数据库文件在操作系统中存储的文件名,是操作系统文件的实际名称,每个物理文件名都有明确的存储位置(文件所在的路径),其文件名称比较长,在 SQL Server 内部访问非常不便。因此,每个数据库又有逻辑文件名,逻辑文件名只在 T-SQL 语句中使用,是实际磁盘文件名的代号,比较简单,引用起来比较方便,一个物理文件名对应一个逻辑文件名。

2) 数据库文件组

为了方便数据库管理员管理多个数据文件,可以将多个数据库文件集合起来形成一个整体,称为文件组。文件组是 SQL Server 2019 数据文件的一种逻辑管理单位,对文件分组的目的就是便于进行管理和进行数据分配。可以将文件组中的文件存放在不同的磁盘,以便提高数据库的访问性能。

每个文件组对应一个组名,SQL Server 2019 提供了两种类型的文件组,包括主

(primary)文件组、用户自定义(user-defined)文件组,还有一个特殊的默认(default)文件组。

(1) 主文件组:当创建数据库时,如果用户没有定义文件组,系统会自动建立主文件组,当数据文件没有指定文件组时,默认都在主文件组中。主文件组包含了所有的系统表、主要数据文件和所有没有包含在其他文件组的次数据文件。主数据文件只能置于主文件组。

(2) 用户自定义文件组:用户创建的文件组也称为次文件组,包含所有在使用 create database 或 alter database 的 SQL 语句,使用 filegroup 关键字来指定文件组的文件,该组包含逻辑上一体的数据文件和相关信息。一个数据库中,用户可以根据需要创建多个自定义文件组,创建用户自定义文件组的主要目的是便于数据分配。

(3) 默认文件组:如果在创建数据文件时没有明确指定所属的文件组,则该数据文件会被放置在默认文件组中。在每个数据库中,每次只能有一个文件组是默认文件组。可以将用户自定义文件组指定为默认文件组,如果没有指定默认文件组,则系统将主文件组设置为默认文件组。

大多数数据库只需要一个文件组和一个日志文件就可以很好地运行,如果数据库中的文件很多,则需要创建用户自定义文件组。

使用数据库文件和文件组时,必须遵循以下规则。

(1) 一个数据文件只能存在于一个文件组中,不能存在于两个或两个以上的文件组中。

(2) 一个文件组也只能被一个数据库使用。

(3) 日志文件不属于任何文件组。

2. 数据库的存储

数据库是以文件的方式存储到磁盘中,其中数据文件和日志文件的结构不同,存储方式也不一样。

1) 数据文件的存储结构

SQL Server 为了兼顾存取效率,在存储空间分配中使用了较小的数据存储单元,即页和盘区。也就是说,数据库对应磁盘文件在逻辑上可以被划分为多个页,SQL Server 在执行底层的磁盘 I/O 操作时以页为单位。

页是 SQL Server 数据文件存储数据的基本(最小)单位。SQL Server 数据文件的内容在逻辑上是分成连续的页,当数据库配置文件的磁盘空间时,就是配置 $0 \sim n$ 页的连续页,数据库的数据表或索引就是使用这些分页来存放数据。

数据文件划分为不同的页,每个页的大小为 8KB,128 页等于 1MB 空间。当在数据文件新建记录时,如果在空数据文件新建第一笔记录时,不论记录大小,SQL Server 一定配置一页给数据表来存储这笔记录,其他记录则会按照顺序存入分页配置的可用空间中。

对于分页中尚未使用的空间,SQL Server 可以用来存入其他新记录,如果可用空间不足以存入一条记录时,SQL Server 就会配置一个新分页存储这笔记录。也就是说,表中每一行的数据不能跨页存储,分页中的记录一定是完整记录,不会只记录部分字段数据。

盘区是管理存储空间的基本单位,SQL Server 将 8 个连续的页组成一个盘区,即盘区是 8 个连续页的集合。因此,SQL Server 分配存储空间是以 1 盘区/次为单位进行分配的。为了提高空间的利用率,SQL Server 在为数据库中的某个数据表分配存储空间时,采用两

种不同的策略。根据实际保存数据的不同,盘区可以划分为统一盘区(单一盘区)和混合盘区。统一盘区存放的数据为一个数据库对象所有,如某个盘区 8 个连续的页,存放的都是“读者表”的数据。混合盘区可以分配给不同的对象,由多个对象共同使用,如存放“图书表”和“读者表”的数据等。为了提高数据访问的效率,SQL Server 对表或索引进行存储空间分配时,前 8 页都会分配到一个混合盘区中,直到第 9 页需要分配时,才使用统一盘区。也就是说当混合盘区的表或索引的大小增长到 8 页时,系统会将表或索引存放统一到统一盘区中。

2) 日志文件的存储结构

SQL Server 数据库中的事务日志以日志行为单位,每条日志行是由一个日志序列号(log sequence num,LSN)标识,每一日志中都包含该日志行所属的事务 ID。每条新日志行均写入日志的逻辑结尾处,并使用一个比前一行 LSN 大的 LSN。SQL Server 的日志文件中包含着一系列日志行,日志行按照顺序存储到实现事务日志的物理文件集中。

5.4.3 用户数据库的创建与修改

为了创建完善的数据库管理机制,SQL Server 设计了严格的对象命名规则。在创建数据库、数据库对象或引用数据库实例时,必须遵守 SQL Server 的命名规则。

1. 数据库的命名规则

在 SQL Server 中创建数据库时,其名称必须遵循 SQL Server 2019 的标识符的命名规则。数据库的命名规则取决于数据库的兼容级别。兼容级别可以为 80、90、100、110、120、130、140、150,一般来说,SQL Server 2000 使用的是 80 级别,SQL Server 2019 默认和支持的数据库兼容级别为 150 级别,当兼容级别为 150 时,其命名规则如下。

(1) SQL Server 数据库管理系统中的数据库对象名称长度为 1~128,不能超过 128 个字符,本地临时表(临时对象)的名称不能超过 116 个字符,不区分大小写。

(2) 名称的第一个字符不可以使用数字、下划线、@、#、\$ 等符号。

(3) 在中文版的 SQL Server 2019 中,可以直接使用中文名称。

(4) 名称不建议使用 T-SQL 关键词,因为 T-SQL 不区分大小写,因此,不建议包含任何大小写的关键词。

(5) 名称中不能有空格、特殊字符开头,否则需要使用界定标识符“`”或方括号“[]”将名称括起来。

需要注意的是:在 T-SQL 中,以“@”开头的变量表示局部变量,以“@@”开头的变量表示全局变量,以“#”开头的表示全局临时对象,所以,用户在命名数据库时最好不要以这些字符开头,以免引起混乱。

2. 创建数据库

在开发 SQL Server 2019 数据库应用程序之前,首先要设计数据库结构并创建数据库。创建数据库时需要对数据库的属性进行设置,包括数据库的名称、所有者、大小以及存储数据库的文件和文件组。

在 SQL Server 2019 中创建数据库有两种方法,一种是使用 SQL Server Management Studio 创建数据库,此方法是图形化操作界面、简单、直观,适合初学者学习;另一种是使用 T-SQL 语句创建数据库,此方法难度稍大,需要对 T-SQL 语法和语句非常熟悉,但可以将数据库的脚本保存下来,在其他计算机上运行以创建相同的数据库,对于高级用户,此方法



视频讲解

使用起来更加得心应手。

本节主要介绍使用 SQL Server Management Studio 图形管理界面创建数据库,第二种方法将在 6.2.1 节详细介绍。

1) 创建数据库的步骤

在 SQL Server Management Studio 中创建数据库的具体步骤如下。

(1) 启动 SQL Server Management Studio,并连接到 SQL Server 2019 中的数据库,在“对象资源管理器”中展开“服务器”→“数据库”,右击“数据库”选项,在弹出的快捷菜单中选择“新建数据库”命令,打开如图 5-22 所示的“新建数据库”窗口。该窗口有三个选项:常规、选项和文件组。

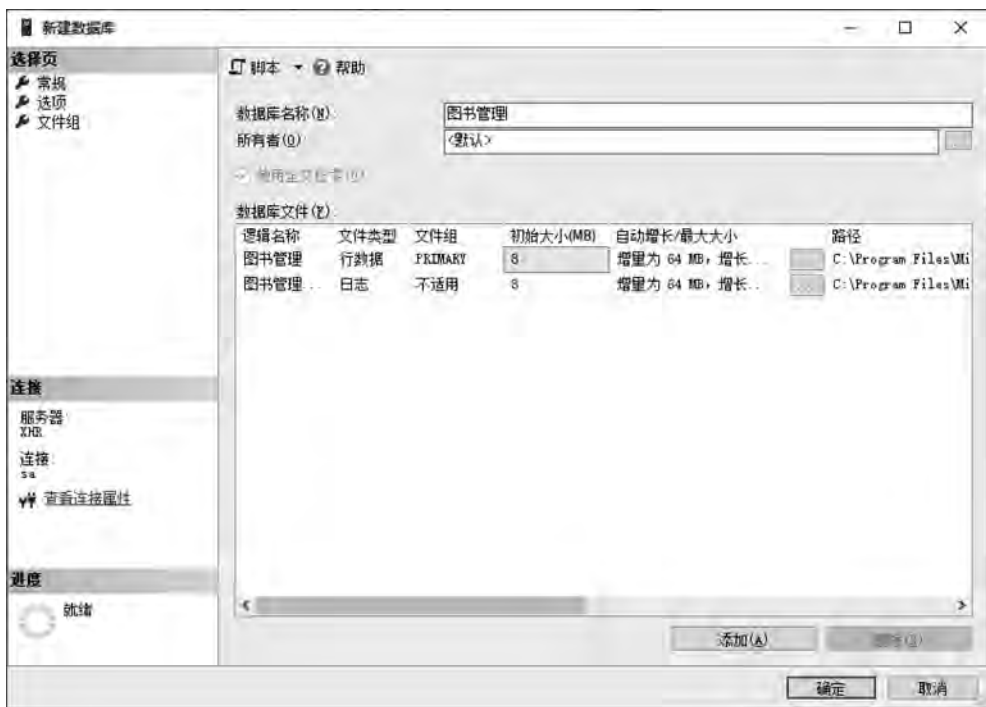


图 5-22 “新建数据库”窗口

(2) 单击“常规”选项,在“数据库名称”文本框中输入要创建的数据库名称“图书管理”。数据库名称设置完成后,SQL Server 2019 自动在数据库文件列表中产生一个主数据文件“图书管理.mdf”和一个日志文件“图书管理_log.ldf”,同时显示文件组、自动增长和路径的默认值,用户可以根据需要修改这些默认设置,也可以单击右下角的“添加”按钮添加数据文件。这里均采用默认值。

(3) 在“新建数据库”窗口中单击“选项”选项,设置数据库的排序规则、恢复模式、兼容性级别和其他选项,如图 5-23 所示,这里采用默认设置。

(4) 在“新建数据库”窗口中,单击“文件组”选项,设置或添加文件组的属性,如是否只读、是否为默认文件组,如图 5-24 所示,单击“添加”或“删除”按钮即可添加或删除用户自定义的文件组。

(5) 在“新建数据库”对话框的“常规”选项中,分别为各个文件指定文件组。



图 5-23 “选项”选项



图 5-24 “文件组”选项

(6) 设置完上面的参数,单击“确定”按钮,系统开始创建数据库。SQL Server 2019 在执行创建过程中将对数据库进行检验,如果存在一个相同名称的数据库,则创建操作失败,并提示错误信息。创建成功后,刷新“对象资源管理器”窗口的“数据库”结点的内容,再展开“数据库”结点,则会显示出新创建的数据库“图书管理”。

2) 创建数据库的基本信息说明

(1) “新建数据库”窗口中的“常规”选项中“所有者”可以是任何具有创建数据库权限的登录名,对数据库有完全操作权限的用户。在“所有者”文本框中可以输入数据库的所有者,也可以单击  按钮,打开“选择数据库所有者”对话框,选择数据库的所有者。默认值表示当前登录到 SQL Server 上的账户,如 sa。

(2) 数据库文件的逻辑名称:默认情况下,数据文件的逻辑文件名和数据库同名,创建数据库时,系统会以数据库文件名作为前缀创建主数据文件和日志文件,日志文件的逻辑名称加上“_log”,也可以为数据文件和日志文件指定其他合法的逻辑名称。

(3) 路径:数据库文件存放的物理位置,默认情况下,SQL Server 2019 将数据文件保存在安装目录下的 data 文件中,用户可以根据需要修改。

(4) 文件名:数据文件和日志文件的物理文件名,默认时与数据库同名,主数据文件名的扩展名是 .mdf,日志文件名在主数据文件名上加上“_log”,其扩展名是 .ldf。

注意: 数据文件尽量不要保存在系统盘上,并与日志文件保存在不同的磁盘区域中。

5.4.4 数据库的管理

随着数据库的增长或变化,用户需要用手动方式对数据库进行管理,数据库的管理包括修改数据库的配置、删除数据库、查看数据库的属性、更改名称、扩充或收缩数据库与日志文件等。

1. 查看和修改数据库参数

对于已有的数据库,可以查看数据库的属性,在数据库创建完成后,又可能因为种种原因,需要修改数据库的设置,修改数据库的主要内容包括扩充数据库的数据或事务日志的存储空间,增加或减少数据文件和事务日志文件,更改数据库的名称、数据库的配置,收缩数据库或事务日志空间。

查看和修改数据库参数可以在 SQL Server Management Studio 中进行,也可以用 T-SQL 语句完成。在此介绍第一种方法。

在 SQL Server Management Studio 中查看和修改数据库的步骤如下。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中展开服务器,定位到要查看的数据库“图书管理”。

(2) 右击目标“图书管理”数据库,在弹出的快捷菜单中选择“属性”命令,打开如图 5-25 所示的“数据库属性”窗口。

(3) 在该窗口的“常规”选项中,可看到该数据库的基本信息,如数据库的名称、状态、所有者、创建数据库的时间、大小、数据文件和日志文件剩余的可用空间、数据库的备份、连接数据库的用户数等信息。

(4) 在“文件”选项中,可以添加、删除数据库文件以及修改数据库文件的初始大小和自动增长属性等相关属性。



图 5-25 “数据库属性”窗口

(5) 在“文件组”选项中,可以添加、删除文件组以及修改文件组的相关属性。

(6) 在“选项”选项中,可以控制数据库是单用户使用模式还是多用户使用模式,数据库是否仅可读,设置此数据库是否自动关闭、自动收缩和数据库的兼容级别、限制用户对数据库的访问等。这里对“图书管理”数据库进行“限制访问”设置,选择“状态”→“限制访问”下拉列表框,出现三个选项,如图 5-26 所示。

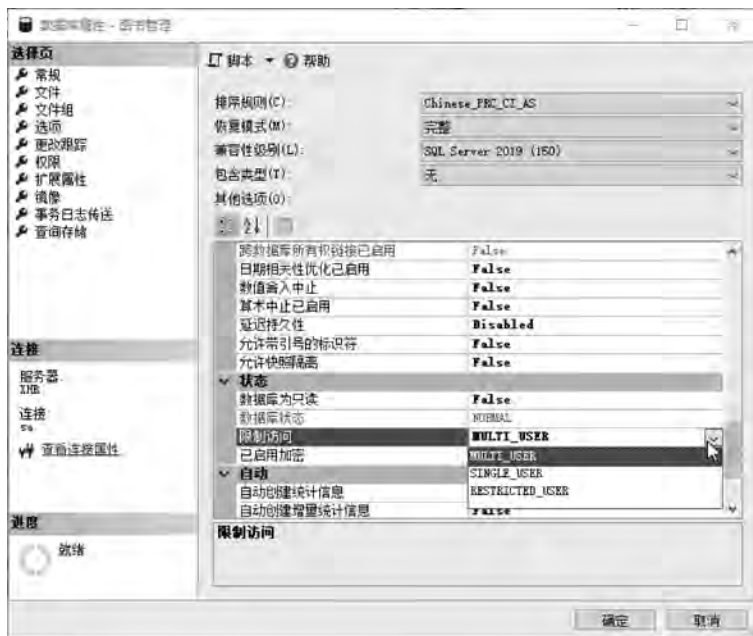


图 5-26 “限制访问”设置

其中, MULTI_USER 指数据库处于正常生产状态, 允许多个用户同时访问数据库; SINGLE_USER 指定一次只能允许一个用户访问, 其他用户的连接被中断, 拥有对数据库的独占访问权限; RESTRICTED_USER 是限制除数据库所有者、数据库创建者和系统管理员以外的角色成员访问数据库, 但对数据库的连接不加限制, 一般用于维护数据库时的设置。

(7) 设置好后, 单击“确定”按钮即可。

2. 收缩用户数据库

当为数据库分配的磁盘空间过大时, 为了节省存储空间, 可以收缩数据库。SQL Server 允许收缩数据库以及数据库文件以删除未使用的页, 该活动在后台进行, 并不影响数据库内的用户活动。在收缩数据库之前, 可以先查看数据库磁盘的使用情况。

1) 查看数据库磁盘使用情况

SQL Server 提供了丰富的数据库报表, 可以查看数据库的使用情况。具体操作: 右击要查看的数据库, 如“图书管理”数据库, 在弹出的快捷菜单中依次选择“报表 Reports”→“磁盘使用情况”, 或“报表 Reports”→“标准报表命令”, 出现磁盘使用情况报表。根据磁盘使用情况报表可以查看到数据库、数据文件以及事务文件的空间使用量, 并且以饼图的方式显示各文件的空间使用率, 根据报表, 以便决定是否需要收缩数据库。

2) 设置自动收缩数据库

设置数据库的自动收缩, 可以在数据库的属性窗口的“选项”选项卡页面中进行设置, 将选项中的“自动收缩”设置为 True 即可。

3) 手动收缩数据库

手动收缩用户数据库的步骤如下。

(1) 在 SQL Server Management Studio 中, 右击需要收缩的用户数据库, 如图书管理数据库, 从弹出的快捷菜单中依次选择“任务”→“收缩”→“数据库”命令。

(2) 打开“收缩数据库”窗口, 如图 5-27 所示, 选中“在释放未使用的空间前重新组织文件”复选框, 设置其下的“收缩后文件中的最大可用空间”为 60%, 表示将数据库的可用空间从原先收缩到 60%。

(3) 单击“确定”按钮, 即可完成数据库的收缩操作。

(4) 如果单击“脚本”按钮, 则会在“新建查询”界面中显示出收缩数据库操作的脚本。

4) 手动收缩数据库文件

手动收缩数据库文件的步骤如下。

(1) 在 SQL Server Management Studio 中, 右击需要收缩的用户数据库, 如图书管理数据库, 从弹出的快捷菜单中依次选择“任务”→“收缩”→“文件”命令。

(2) 打开收缩文件的窗口, 在此窗口中, 可以对指定的数据文件、日志文件分别进行收缩, 收缩操作可在三种选择其一。

(3) 单击“确定”按钮, 即可完成操作。

(4) 如果单击“脚本”按钮, 则会在“新建查询”界面中显示出收缩文件操作的脚本。

3. 重命名数据库

在 SQL Server 2019 中, 用户可以根据需要修改数据库的名称, 其名称可以包含任何符合标识符规则的字符。

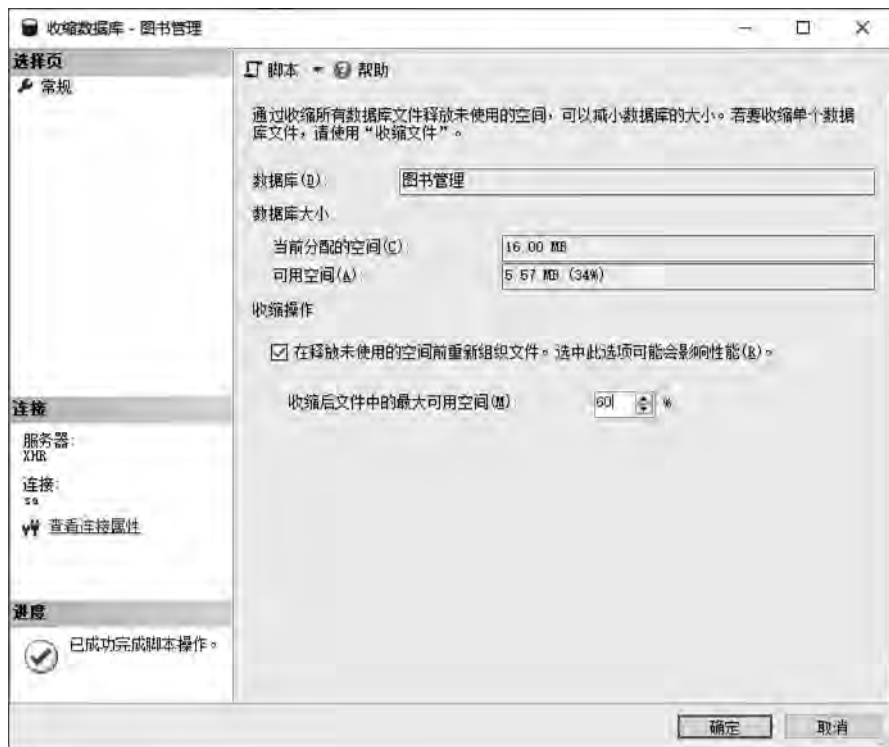


图 5-27 “收缩数据库”窗口

更改数据库的名称可以在 SQL Server Management Studio 的“对象资源管理器”窗格中修改,也可以用 T-SQL 语句。

在 SQL Server Management Studio 中修改数据库的步骤如下。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中展开数据库结点。

(2) 右击需要重命名的“图书管理”数据库,在弹出的快捷菜单中选择“重命名”命令,此时,数据库名处于可修改的状态,直接输入修改后的数据库名称,然后按 Enter 键进行确认即可。

注意: 在数据库重命名之前,应确保没有人使用该数据库,而且该数据库设置为单用户模式。

4. 删除数据库

删除数据库也是数据库管理中重要的操作之一,如果用户不再需要某一数据库就可以将其从 SQL Server 服务器上删除,以释放该数据库所占有的磁盘空间。

删除数据库一定要慎重,因为数据库的删除是彻底地将相应的数据库文件从物理磁盘上删除,是永久性的。一旦删除,与此数据库有关联的数据库文件和事务日志文件都会被删除,存储在系统数据库中有关该数据库的所有信息也被删除;如果数据库正在被用户使用,则无法将其删除,删除数据库仅限于 dbo(database owner)和 sa(super administrator)。

在 SQL Server Management Studio 中删除数据库的步骤如下。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中展开数据库

结点。

(2) 右击需要删除的数据库,在弹出的菜单中选择“删除”命令,弹出“删除对象”窗口。单击“确定”按钮,即可删除数据库。

如果在“删除对象”窗口的下方选中了“删除数据库备份和还原历史记录信息”复选框,则在删除数据库的同时,也将从 msdb 数据库中删除该数据库的备份和还原历史记录。

如果选中了“关闭现有连接”复选框,在删除数据库前,SQL Server 会自动将所有与该数据库相连的连接全部关闭后,再删除数据库。如果没有选中该复选框,而且有其他活动连接在要删除的数据库时,将会出现错误信息。

注意: 系统数据库 msdb、model、master、tempdb 无法删除,删除数据库后应立即备份 master 数据库,因为删除数据库将更新 master 数据库中的信息。

5. 分离和附加数据库

分离和附加是数据库开发过程中的重要操作。如果需要将数据库从一台服务器复制到另一台服务器,或者需要从一个硬盘或分区迁移到另一个硬盘或分区时,可以使用分离和附加数据库的方法,将用户数据库从 SQL Server 服务器的管理中分离,同时保持数据文件与日志文件的完整性和一致性,之后,使用附加数据库的方法将分离后的数据库文件附加到任何 SQL Server 实例中。

1) 分离数据库

如果不需要对数据库进行管理,又希望保留其数据,可以对其执行分离操作。

分离数据库是指将数据库从 SQL Server 实例中删除。也就是说,分离实际上只是从 SQL Server 的 master 等系统数据库中删除与被分离数据库的有关信息,并没有删除磁盘上的数据库文件,但分离之后的数据库无法在当前服务器上使用。

在 SQL Server Management Studio 中分离数据库的具体操作如下。

(1) 在对象资源管理器中,展开“服务器名称”→“数据库”结点,右击需要分离的数据库,如图书管理,从弹出的快捷菜单中选择“任务”→“分离”命令,打开“分离数据库”窗口。

(2) 在“分离数据库”窗口中,设置分离参数。如果看到“消息”栏中有“活动链接”的消息,说明已有用户连接到数据库,需要选中“删除连接”(表示分离数据库时需要先把当前的连接删除)。“更新统计信息”指在分离之前选择是否更新过时的优化统计信息,若“状态”显示为“就绪”,即可以进行分离该数据库。

(3) 单击“确定”按钮,分离数据库。

分离数据库后,在对象资源管理器中将不再出现该数据库。用户可以将该数据库的数据文件及日志文件拷贝到移动硬盘或 U 盘,以便在需要时附加该数据库所用。

说明: 分离数据库时,需要拥有对数据库的独占访问权限。如果要分离的数据库正在使用,则必须将数据库设置为 SINGLE_USER 模式,才能进行分离操作。

2) 附加数据库

分离后的数据库如果需要重新使用,可以被重新附加到当前或其他 SQL Server 实例中。附加就是将数据库的信息保存到 SQL Server 的 master 等系统数据库中,以便 SQL Server 可以再次使用和管理这个数据库。

在 SQL Server Management Studio 中附加数据库的具体操作如下。

(1) 在对象资源管理器中,展开“服务器名称”→“数据库”结点,右击“数据库”结点,从

弹出的快捷菜单中选择“附加”命令,打开“附加数据库”窗口。

(2) 在“附加数据库”窗口中,单击“添加”按钮,弹出“定位数据库文件”窗口,在该窗口中查找并选择要附加的主数据文件“图书管理.mdf”。

说明:该对话框默认只显示数据库的主数据文件,如图书管理_data.mdf,只要选择了主数据文件,数据库的日志文件会自动加载到“数据库详细信息”列表框中。

(3) 单击“确定”按钮,在“附加数据库”窗口中会根据添加的主数据文件,更新“要附加的数据库列表”和“数据库详细信息”列表,如图 5-28 所示。其中,“附加为”表示最终附加的数据库名称,可以与默认的“数据库名称”相同,也可以根据需要进行更改。

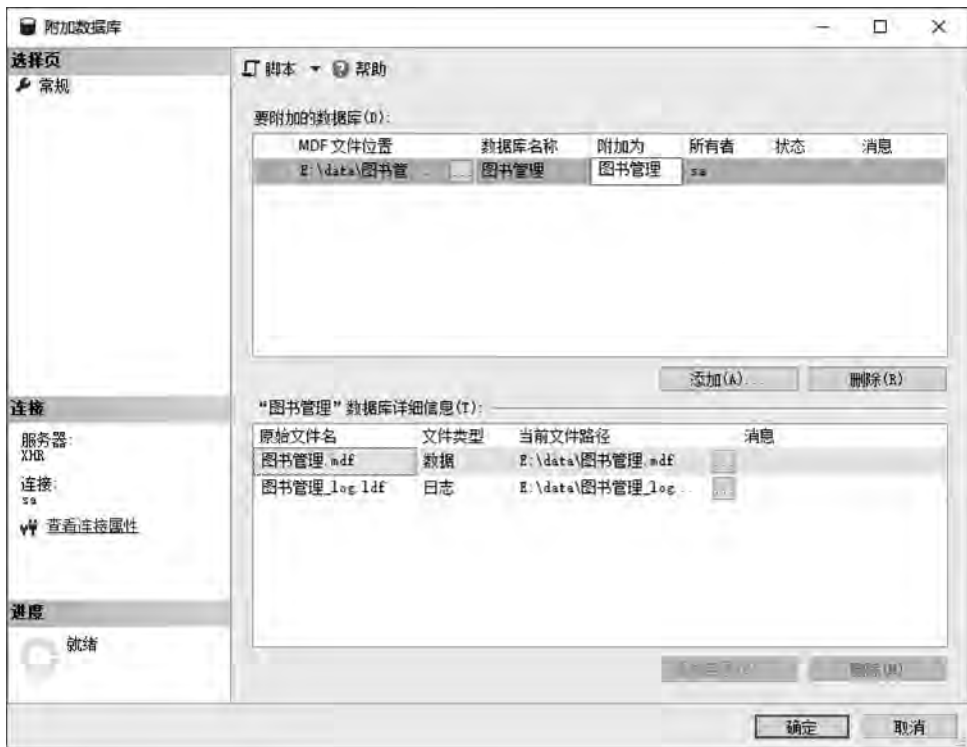


图 5-28 添加了数据文件的“附加数据库”窗口

(4) 确认信息无误后,单击“确定”按钮,执行附加数据库操作。操作完成后,在对象资源管理器中将显示所附加的数据库。

需要强调的是,数据库的主数据文件中存放了其他文件的相关信息,因此附加数据库时,只要指定主数据文件即可,但在分离数据库后,数据库的主数据文件与日志文件应该成对存放在同一文件夹下,如果移动了日志文件,在数据库附加时就会出现文件“找不到”的提示。

5.5 SQL Server 2019 数据库基本对象操作

数据库创建好后,就可投入使用,用户可以根据需要在数据库中建立相关的对象,SQL Server 2019 的数据库对象包括基本表、视图、索引、存储过程、触发器、规则、用户、角色、架

构等。本节主要介绍最常用的基本表、视图、索引等。数据库是表的集合,所以用户在建立数据库时首先要定义表。在创建表时,需要确定表的结构,也就是确定表中的各列及各列的数据类型,例如,是字符型、货币型还是数值型。只有设计好表的结构,系统才会在磁盘上开辟相应的空间,用户才能在表中填写数据。因此,在创建表之前,首先介绍 SQL Server 2019 的数据类型。

5.5.1 SQL Server 2019 的数据类型

数据类型是指用于存储、检索及解释数据值的类型的预先定义的命名方法,决定了数据在计算机中的存储格式,代表不同的信息类型。SQL Server 数据类型可以定义数据表字段能够存储哪一种数据和使用多少字节来存储数据,即数据的范围。为表中的各列选择合适的数据类型尤为重要,因为它影响着系统的空间利用、性能和是否易于管理等特性。SQL Server 2019 数据类型可分为两种:系统内置数据类型和用户自定义数据类型。

1. 系统数据类型

SQL Server 2019 提供了丰富的系统数据类型,可以支持大部分数据库应用系统。按系统数据的表现形式和存储方式分为整数型字符型、日期型、数值型、货币型和二进制等。下面介绍几种 SQL Server 比较常用的数据类型。

1) 整数数据类型

整数数据类型可以存储整数数值数据。SQL Server 提供数种数据类型来存不同范围的整数数据,实际应用时,可以按照字段取值的范围决定使用哪一种整数数据类型,如表 5-1 所示。

表 5-1 整数数据类型

类型名称	取值范围	字节
int	可存储 $-2^{31} \sim 2^{31} - 1$ 范围的整数数据	4
smallint	可存储 $-2^{15} \sim 2^{15} - 1$ 范围的整数数据	2
tinyint	可存储 0~255 之间的所有整数	1
bigint	可存储 $-2^{63} \sim 2^{63} - 1$ 范围的整数数据	8
bit	可存储取值为 1、0 或 NULL 的整型数据	

2) 精确小数数据类型

精确小数数据类型可以存储包含小数的数值数据,而且完全保留数值数据的精确度。SQL Server 提供两种精确小数数据类型,如表 5-2 所示。

表 5-2 精确小数数据类型

类型名称	取值范围	字节
decimal(p, s)	可存储 $-10^{38} + 1 \sim 10^{38} - 1$ 范围的数据	占 5~17 字节
numeric(p, s)	可存储 $-10^{38} + 1 \sim 10^{38} - 1$ 范围的数据	占 5~17 字节

其中, p 为精度,指定小数点左边和右边可以存储的十进制数字的最大个数; s 为小数位数,指定小数点右边可以存储的十进制数字的最大个数。

3) 浮点数数据类型

浮点数数据类型可以用来存储拥有小数点的数值数据,此类型也称为不精确小数数据

类型,因为当数值非常大或非常小时,其存储的数据是一个近似值,如表 5-3 所示。

表 5-3 浮点数数据类型

类 型 名 称	取 值 范 围	字 节
float(<i>n</i>)	可存储 $-1.79\text{E}+308 \sim 1.79\text{E}+308$	占 4 或 8 字节
real	可存储 $-3.40\text{E}+38 \sim 3.40\text{E}+38$	占 4 字节

其中, float 数据类型可以指定存储数值数据的位数 *n*, *n* 的值如果是 1~24, SQL Server 使用 24, 占用 4 字节; *n* 的值如果是 25~53, 则使用 53, 占用 8 字节, 是默认值。当使用 float 和 real 数据类型来定义数据表字段时, 如果数值超过精确度的位数, 就会四舍五入而产生误差的近似值。

4) 货币数据类型

在 SQL Server 中可以使用 money 和 smallmoney 两种数据类型存储货币数据或货币值, 这些数据类型可以使用常用的货币符号, 如 ¥、\$。货币数据类型如表 5-4 所示。

表 5-4 货币数据类型

类 型 名 称	取 值 范 围	字 节
money	可存储 $-2^{63} \sim 2^{63} - 1$ 范围的货币型数据, 精确到千分之十	占 8 字节
smallmoney	可存储 $-214748.3468 \sim 214748.3467$ 范围的货币型数据	占 4 字节

货币型常量可以包含小数点, 都可以精确到小数点后 4 位。实际上, 可以使用 decimal 数据类型来存储货币数据。

5) 字符串数据类型

字符串数据类型是用来存储由字母、符号和数字组成的字符串数据。在 SQL Server 中, 字符的编码方式有两种: ASCII 码和 Unicode 码。ASCII 码指的是不同国家或地区的编码长度不一样, 如英文字母编码是 1 字节(8 位), 中文汉字的编码是 2 字节(16 位)。Unicode 码存储的是统一字符编码的字符串数据, 解决字符集不兼容问题, 不管对哪个地区、哪种语言所有的字符均使用双字节(16 位)编码。常用的字符串数据类型的说明如表 5-5 所示。

表 5-5 字符串数据类型

类 型 名 称		取 值 范 围	字 节
ASCII 码	char(<i>n</i>)	存放 1~8000 固定长度的字符	<i>n</i>
	varchar(<i>n</i>)	存放 1~8000 可变长的字符	最大 <i>n</i> 个字节
	varchar(max)	存放 2G 个字符	最大 2GB
Unicode 码	nchar(<i>n</i>)	存放 1~4000 固定长度的字符	$n \times 2$
	nvarchar(<i>n</i>)	存放 1~4000 可变长的字符	$n \times 2$
	nvarchar(max)	存放 1G 个字符	最大 2GB
二进制码	binary	存放 1~8000 字节的定长二进制数据	<i>n</i>
	varbinary	存放 1~8000 字节的变长二进制数据	最大为 <i>n</i> 字节
	varbinary(max)	存放变动长度为二进制字符串 2G 字节	最大 2GB
	image	存放长度可变的二进制数据	最大 2GB

字符串数据类型常量需用定界符(单引号)括起来。

6) 日期数据类型

日期数据类型可以存储日期与时间数据,SQL Server 提供了 datetime 和 smalldatetime 两种数据类型,在 2008 后的版本中新建了 date、time、datetime2 和 datetimeoffset 四种日期数据类型。常用的日期时间数据类型如表 5-6 所示。

表 5-6 日期数据类型

类 型 名 称	取 值 范 围	字 节
datetime	日期从 1/1/1753 到 12/31/9999,时间精确到 3.33ms	8
smalldatetime	日期从 1/1/1900 到 6/6/2079,时间精确到分	4
date	日期从 1/1/0001 到 12/31/9999	3
time(n)	表示一天中的某个时间,使用 24 小时表示	3~5
datetime2(n)	日期从 1/1/0000 到 12/31/9999,时间精确到 100ns	6~8
datetimeoffset	日期从 1/1/0000 到 12/31/9999,时间精确到 100ns	10

注意: 在定义某一具体的日期时间时可以使用字符串按照日期时间的格式进行定义,系统将自动把该字符串转换为日期时间类型。

2. 自定义数据类型

SQL Server 2019 除了系统提供的数据类型外,还支持用户自定义数据类型。允许用户根据需要定义自己需要的数据类型,并且可以用自定义数据类型来声明变量或字段。但用户自定义数据类型是使用 SQL Server 2019 系统提供的原生数据类型为基础来创建,由系统已有的数据类型来派生,而不是定义一个具有新的存储及检索特性的类型。更确切地说,用户自定义数据类型是在创建别名数据类型,是一种数据类型的别名。

当多个表中的列要存储同样类型的数据,且想确保这些列具有完全相同的数据类型、长度和是否为 NULL 属性时,可以使用自定义数据类型。例如,在一个数据库中,有许多表都需要用到 varchar(10)的数据类型,那么用户就可以自己定义一个数据类型,例如,用 vc10 代表 varchar(10)的数据类型,在表需要用到 varchar(10)的列时,都可以将其设为 vc10。

创建用户自定义数据类型时必须提供名称、新数据类型所依据的系统数据类型、是否允许为空值。可以使用 SQL Server Management Studio 或 T-SQL 语句来创建,用户自定义数据类型一旦创建成功,就可以像使用系统数据类型一样使用。

下面利用 SQL Server Management Studio 为数据库图书管理创建基于 varchar 型的用户自定义数据类型 address_code,长度为 20,允许为空。其操作步骤如下。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中展开“图书管理”→“可编程性”→“类型”→“用户自定义数据类型”结点。

(2) 右击“用户自定义数据类型”,从弹出的快捷菜单中选择“新建用户自定义数据类型”命令,打开“新建用户自定义数据类型”对话框。

(3) 在“新建用户自定义数据类型”对话框中进行如下设置。

- ① 在“名称”文本框中输入类型名称 address_code。
- ② 在“数据类型”下拉列表框中选择 varchar 数据类型。
- ③ 在“长度”数值框中输入 20。

④ 选中“允许为空”复选框,表示此数据类型定义的列可以不输入数据。

(4) 设置完毕后,单击“确定”按钮,即创建了自定义的数据类型 address_code。

5.5.2 表的创建和维护

表(Table)是 SQL Server 数据库系统的基本信息存储结构。SQL Server 数据库中的表是一个非常重要的数据库对象,是数据存放的地方,也称基本表。用户所关心的数据都存储在表中,对数据的访问、维护都是通过对表的操作实现的,一个数据库管理员在数据库中打交道最多的也是表。因此,掌握数据库表的操作就显得尤为重要。本节主要介绍如何利用 SQL Server Management Studio 的对象资源管理器创建“图书管理”数据库的四个基本表:读者类别表、读者表、图书表和借阅表。各表的基本结构见 3.5.3 节。

表的创建一般要经过定义表的结构,设置约束和添加数据三个步骤,其中设置约束可以在定义表结构时进行,也可以在表结构定义完成后,在已创建好的表结构上通过修改表的方式添加约束。

1. 表结构的创建、修改和删除

表是包含数据库中所有数据的数据库对象,与平常所说的表类似,但必须遵循关系的性质,也是由行和列组成,用于组织和存储数据。表的创建首先要设计和定义表的结构,即录入数据的表的框架,有多少列组成,每列的数据类型,有无约束等。如果表结构创建好后,还可以根据实际需要进行表结构的修改或删除,如新增某些列或约束。

1) 创建表结构

表定义为列的集合,创建表结构也就是定义表中各列的过程,如添加字段、设置字段的主键等属性。

使用 SQL Server Management Studio 可以非常方便地创建表结构,下面以读者信息表为例,介绍创建表结构的具体步骤。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中依次展开需要创建表的数据库,如“数据库”→“图书管理”→“表”结点。右击“表”结点,从弹出的快捷菜单中选择“新建”→“表”命令,打开如图 5-29 所示的表设计器窗口。通过表设计器可以进行表结构的创建、修改。

表设计器主要分为上下两部分。上部分用来定义表的列(字段),包括列名、数据类型、允许空属性;下部分用来设置列的其他属性,如是否使用默认值、是否为“标识列”等。

用户可用鼠标、Tab 键或方向键在单元格间移动和选择,完成“列名”“数据类型”“长度”“允许空”栏中相关数据的输入。需要注意的是,有些数据类型的长度是固定的,不能修改。

(2) 各字段设置完成后,右击字段“读者卡号”,在弹出的快捷菜单中选择“设置主键”命令或使用工具栏上的“设置主键/取消主键”按钮 ,将“读者卡号”设置为读者表的主键,设置好后读者卡号左侧显示有一个小钥匙图标,如图 5-30 所示。

(3) 执行主菜单中“视图”→“属性窗口”命令或使用 F4 键,打开属性面板。在该面板中可以设置新建表的名称、架构等。在表名称中输入“读者”,架构使用默认的 dbo。

(4) 执行“文件”→“保存”命令或者单击工具栏的“保存”按钮 ,打开“选择名称”对话框,如图 5-31 所示。



视频讲解

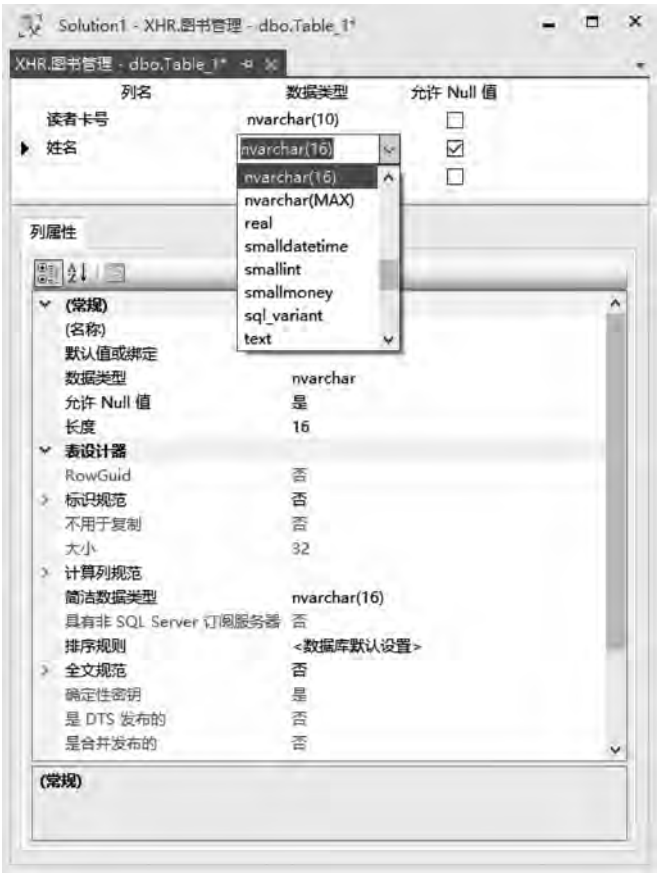


图 5-29 表设计器窗口



图 5-30 设置表的主键

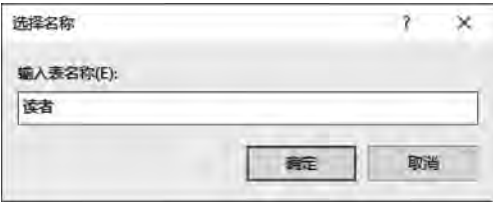


图 5-31 “选择名称”对话框



图 5-32 “图书管理”数据库中的四张表

(5) 在文本框中输入要保存的数据表名称“读者”，单击“确定”按钮即可完成表的创建。关闭表设计器之后，展开图书管理数据库，在表结点下可以看到创建的表。

同样的方法，可以创建“图书管理”数据库中的其他表，图书管理数据库各表的创建结果如图 5-32 所示。

2) 修改表结构

在使用数据表的过程中可以根据需要随时对表的列、约束等属性进行修改。修改操作包括增加或删除列，修改列的名称、数据类型、长度，修改表的名称等。在 SQL Server Management Studio 中修改表结构的操作步骤如下。

(1) 打开 SQL Server Management Studio 窗口，在“对象资源管理器”中依次展开“数据库”→“图书管理”→“表”结点。鼠标右击需要修改的表，从弹出快捷菜单选择“修

改”命令，打开“表设计器”。

(2) 如果要增加一列，首先选择新增加列的位置，然后右击，在弹出的快捷菜单中选择“插入列”命令，这时在选定的列的上面会出现一个空行，在空行中输入新增加列的信息。

(3) 如果要删除一列，右击需要删除的列，在弹出的快捷菜单中选择“删除列”命令。

(4) 如果要修改某一列的名称、数据类型、数据长度以及是否为空值，可以直接在表设计器中修改。

(5) 修改完后，单击工具栏上的“保存”按钮即可。

如果在保存修改时出现如图 5-33 所示的对话框，是因为 SQL Server 默认设置启动了“阻止保存要求重新创建表的更改”选项，该选项会阻止保存对表的修改操作。要更改此项设置，可以在 SQL Server Management Studio 中，通过选择菜单“工具”→“选项”命令，打开如图 5-34 所示的“选项”对话框，选择左侧的“设计器”→“表设计器和数据库设计器”。在右边的“表选项”中不要选中“阻止保存要求重新创建表的更改”复选框。



图 5-33 阻止保存修改内容的警告

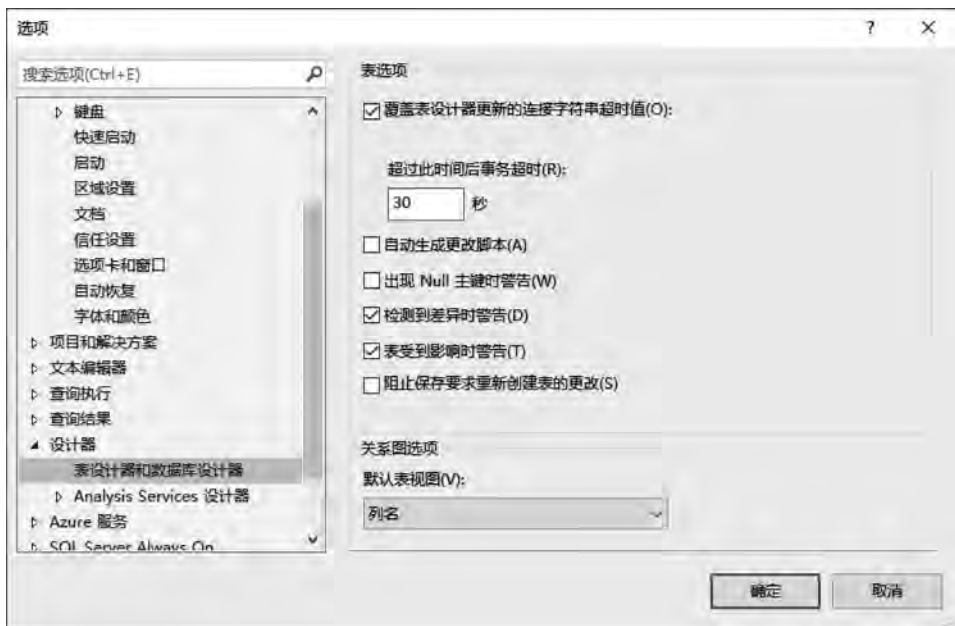


图 5-34 “选项”对话框

注意：在 SSMS 中可以对表中字段的顺序进行调整，只需要选中要调整顺序的列左边的方块，然后通过拖动的方式便可以实现列顺序的调整。

3) 删除表结构

对于数据库中不需要的数据表可以将其删除，以释放存储空间，删除表时，表的结构定义、数据、约束等都将永久地从数据库中删除。删除基本表同样可以在 SQL Server Management Studio 的对象资源管理器中删除，其操作步骤是右击需要删除的表，从弹出快捷菜单选择“删除”命令，打开“删除对象”对话框，单击“确定”按钮，即可删除。

2. SQL Server 的表约束

为了减少输入错误、防止出现非法数据，在创建表结构时可以在列字段上设置约束，以保证数据库中数据的一致性和完整性。SQL Server 2019 中的约束包括主键约束、唯一性约束、空值约束、默认值约束、检查约束、外键约束六种。

约束可以实施在列级和表级两个层次上：列级约束只涉及所约束的一个列，表级约束是指对表中各元组之间、若干关系之间的约束，如果约束条件表达式需要涉及多列属性，应当作为表级完整性约束来定义。

1) 主键约束

主键约束是能够唯一地标识表中的每一条记录的列或列的组合，即主键的列不能为空值，也不能为重复的值。例如，读者表的读者卡号列上可以设置主键约束，保证该列没有空值和重复值。主键约束实现了实体完整性规则，每一个表只能定义一个主键，并且系统自动为其创建主索引（主索引是建立在有序数据表的基于主键的排序字段上，即主索引的索引字段与主文件数据表的主键有对应关系）。主键约束设置完成后，可以在数据表结点下的“键”结点中查看主键约束情况。

2) 唯一性约束

唯一性约束用于保证属性列中不会出现重复的属性值。在一个表中可以定义多个唯一性约束,定义了唯一性约束的列可以取空值。

唯一性约束分为 UNIQUE 约束和主键约束。

唯一性约束与主键约束的区别有两点:一是一个表可以定义多个唯一性约束,但只能定义一个主键约束;二是定义了唯一性约束的列可以输入空值,而定义了主键约束的列不能取空值。

使用 SQL Server Management Studio 的对象资源管理器为读者表添加唯一性约束的操作步骤如下。

(1) 右击“读者”表,在弹出的快捷菜单中选择“设计”选项,进入读者表的设计窗口。

(2) 在“表设计器”窗口中,右击表设计器空白区,选择快捷菜单的“索引/键”命令(或者在“表设计器”工具栏中单击“管理索引/键”按钮),系统弹出“索引/键”对话框,如图 5-35 所示。此时,读者表只是将读者卡号作为主键,还没有创建唯一性约束。



图 5-35 “索引/键”对话框

(3) 在“索引/键”对话框中单击“添加”按钮,在左边列表中将会新建一行默认名称为 IX_读者的索引键,选中 IX_读者,然后在右边的属性窗口中,设置“类型”为“唯一键”,将“是否唯一的”选项改为“是”,单击列后的按钮,打开如图 5-36 所示的“索引列”对话框,指定列名为“姓名”,排序顺序为“升序”,单击“确定”按钮,返回到“索引/键”对话框。

(4) 可以通过右侧属性列表框中的“名称”项进行索引/键的名称修改,修改为 IX_name,设置结果如图 5-37 所示。

(5) 单击“关闭”按钮,保存表设计,此时唯一性约束已经创建完成。在对象资源管理器中展开“图书管理”数据库→“表”→“读者”→“索引”结点,可以看到唯一性约束。



图 5-36 索引列



图 5-37 修改索引/键的名称

3) 空值约束

空值约束限制属性值允许为空或不允许为空,其值为 NULL,表示该列允许为空;其值为 NOT NULL,表示不能为空,必须填入内容,否则输入数据时不能被数据库接收。

尽量避免使用空值,因为空值会使查询和更新变得复杂,而且空值列与非空值列不能一起使用建立主键约束。

4) 默认值约束

默认值约束即 DEFAULT 约束,一般将属性列中使用频率最高的属性值定义为 DEFAULT 约束中的默认值,以减少数据输入的工作量。在 SQL Server 2019 中,可以给列设置默认值。如果某列已设置了默认值,当用户在表中插入记录时,如果没有给该列输入数

据,系统会自动为该列输入默认值。每列只能有一个默认约束。

5) 检查约束

检查约束即 CHECK 约束,用于指定表中属性列值应满足的条件,是所有约束中最灵活的约束,不仅可以对一列的输入进行检查,也可以对多列进行约束。通过在列上设置逻辑表达式来限制列上可以输入的数据值,以此判断输入数据的合法性。检查约束实现域完整性规则。

可以在 SQL Server Management Studio 的对象资源管理器中设置表的 CHECK 约束。创建的操作步骤如下。

(1) 打开表设计器对话框,右击表设计器的空白处,在弹出的快捷菜单选择“CHECK 约束”命令,或工具栏中单击“管理 CHECK 约束”按钮,打开“CHECK 约束”对话框。单击“添加”按钮,设置表的 CHECK 约束对话框如图 5-38 所示。



图 5-38 添加 CHECK 约束

(2) 在表达式文本框中输入 CHECK 表达式,或单击“表达式”右边的按钮,打开“CHECK 约束表达式”对话框,如图 5-39 所示。

(3) 在图 5-39 的表达式文本框中进行 CHECK 约束设置,如对性别进行 CHECK 约束,则约束表达式为性别 in ('男','女'),表示性别只能取“男”或“女”两个值。



图 5-39 CHECK 约束表达式



视频讲解

(4) 约束表达式确定后,单击图 5-39 的“确定”按钮,返回到 CHECK 约束对话框,单击“关闭”按钮返回到表设计器窗口,保存对基本表的修改,完成 CHECK 约束的创建。

6) 外键约束

外键约束是基于表间的约束关系,实现关系数据库中表与表之间的关联关系,如果某列是表的外键,则需要指定该外键要关联到哪一个表的主键字段上。外键约束实现了参照完整性规则。

使用 SQL Server Management Studio 的对象资源管理器创建外键约束,以借阅表建立外键约束为例,创建的操作步骤如下。

(1) 右击借阅表,在弹出的快捷菜单中选择“设计”选项,进入借阅表的设计窗口。

(2) 选择菜单“表设计器”→“关系”命令,弹出“外键关系”对话框。

也可以在对象资源管理器中展开图书管理数据库中的表“借阅”→“键”,右击快捷菜单的“新建外键”命令,打开“外键关系”对话框。

(3) 单击“添加”按钮,系统将在左边“选定的关系”区域中新建一个关系 FK_借阅_借阅,如图 5-40 所示。在右侧的属性窗口中选“表和列规范”选项,并单击旁边的按钮 ,系统弹出“表和列”对话框。

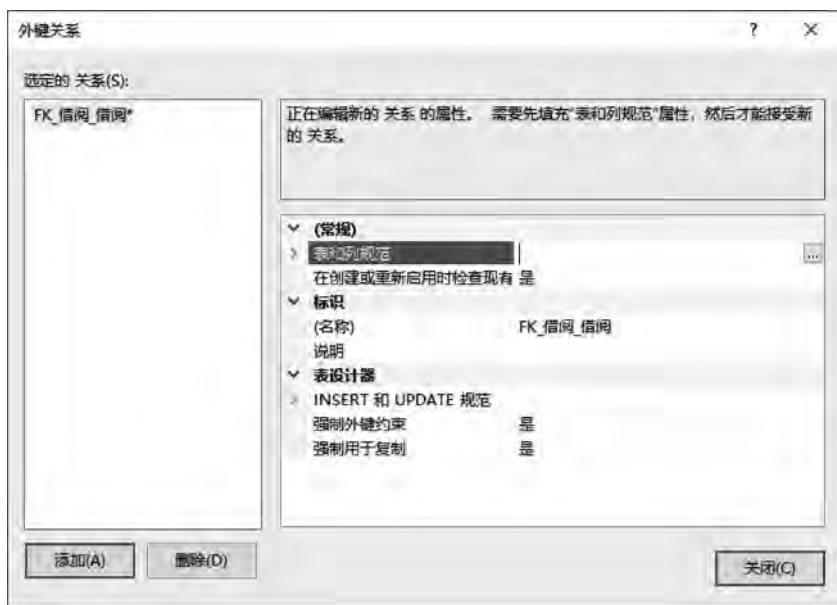


图 5-40 “外键关系”对话框

(4) 选择读者表为主键表,读者卡号作为参照列,选择借阅表为外键表,读者卡号为引用列,如图 5-41 所示。

(5) 单击“确定”按钮,返回到外键关系窗口。

(6) 展开“INSERT 和 UPDATE 规范”,可设置更新和删除规则。如将“删除操作”设置为“级联”,即级联删除,如图 5-42 所示。所谓级联删除,表示当删除主键表数据时,引用主键表数据的外键表数据也同步删除。

(7) 单击“关闭”按钮,返回到“表设计器”窗口,单击工具栏“保存”按钮。



视频讲解

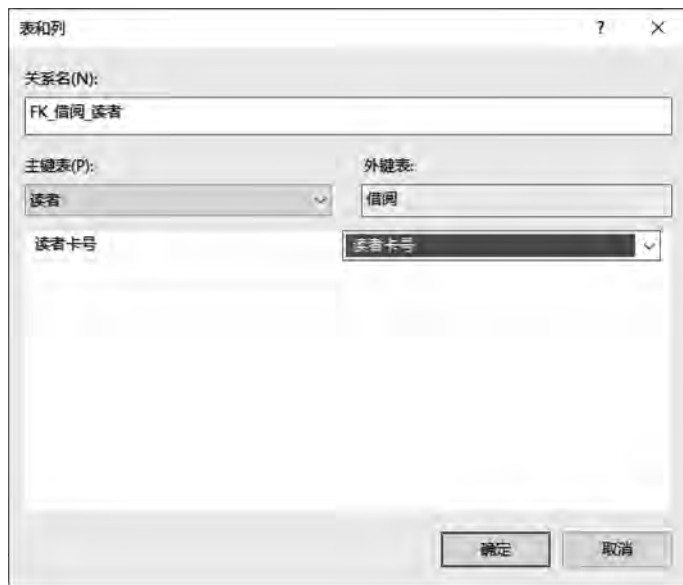


图 5-41 “表和列”对话框



图 5-42 设计外键删除规则

3. 管理表中的数据

创建表结构的目的是利用表来存储和管理数据,没有数据的表只是一个空的表结构,没有实际意义。因此,完成数据表结构的创建后,就可以在表中添加数据,也可以根据用户的需要进行记录的更新和删除操作。

1) 插入数据

使用 SQL Server Management Studio 进行数据的插入,具体步骤如下。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中依次展开数据库“图书管理”→“表”结点,定位到“读者类别”表结点上。

(2) 右键单击“读者类别”表,在弹出的快捷菜单中选择“编辑前 200 行”命令,右窗格显示如图 5-43 所示的数据表编辑窗口,该窗口与电子表格类似,可以看到每一行为一记录的编辑窗口。



类别编号	类别名称	可借阅天数	可借阅数量	逾期罚款额
01	教师	90	6	0.2000
02	博士研究生	80	6	0.4000
03	硕士研究生	60	5	0.6000
04	本科生	30	3	0.8000
NULL	NULL	NULL	NULL	NULL

图 5-43 数据表编辑窗口

图 5-43 的编辑窗口下方是工具栏,可以显示目前数据表的记录数和当前指向的记录,使用工具栏按钮就可以移动和添加记录。在最后“*”行的字段直接输入记录的值,或单击下方工具栏“移动到新行”按钮 ,就可以增加数据表记录。

(3) 输入各记录的字段值后,只要将光标定位到其他记录上,或关闭“结果窗格”窗口,新记录就会自动保存。

在输入新记录时,需要注意以下几点。

- ① 输入字段的数据类型要和字段定义的数据类型一致,否则会出现警告提示框。
- ② 不能为空的字段,必须输入内容。
- ③ 有约束的字段,输入的内容必须满足这些约束;输入的值如果是 NULL 或拥有默认值,可以不用输入任何数据,因为在保存记录时,系统会自动填入 NULL 或默认值。
- ④ 如果不想新增记录,按 X 键放弃新增记录。

2) 更新数据

打开数据表后,找到要修改的记录,然后可以在记录上直接修改字段内容,修改完毕后,只需将光标从该记录上移开,定位到其他记录上,SQL Server 就会自动保存修改的记录。

同样,需要注意修改后的数据记录的合法性,否则系统也会报错。

3) 删除数据

删除数据时,选中要删除的一行或多行连续的记录,然后右击鼠标,在弹出的快捷菜单中选择“删除”命令,弹出删除确认对话框,单击“是”按钮,确认删除,单击“否”按钮则取消删除。

需要注意的是,记录删除后不能再恢复,所以删除前一定要确认。也可以按 Shift 键或 Ctrl 键一次删除多条记录。

5.5.3 数据库关系图的创建和维护

数据库关系图是 SQL Server 的一种数据库对象。关系图工具提供了可视化定义数据

库中表间的约束方法和形象地表示数据库中基本表的逻辑关系。SQL Server Management Studio 提供了数据库关系图功能,可以使用符号图形显示数据库的数据表内容与其关系。不仅如此,数据库关系图也提供了编辑功能,可以直接在数据库关系图的编辑画面创建数据表、创建关系和约束。

1. 创建数据库关系图

以“图书管理”数据库为例,说明建立数据库关系图的方法,建立关系图的步骤如下。

(1) 在 SSMS 的对象资源管理器窗口下,展开数据库“图书管理”→“数据库关系图”结点,右击“数据库关系图”命令。

(2) 如果第一次执行,可以看到需要创建支持对象的信息窗口,如图 5-44 所示。

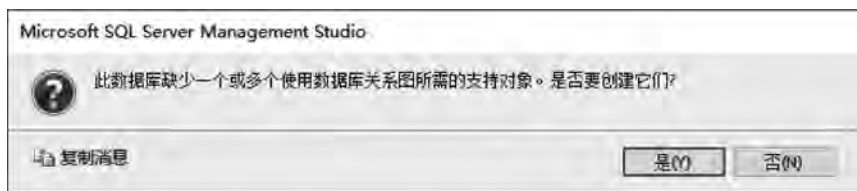


图 5-44 信息窗口

(3) 单击“是”按钮,创建支持对象,同时可以看到“添加表”对话框,如图 5-45 所示。



图 5-45 “添加表”对话框

(4) 在“添加表”对话框中,选中关系图中的基本表后,单击“添加”按钮,完成后,关闭“添加表”对话框,可以看到创建的数据库关系图如图 5-46 所示。

(5) 单击工具栏的“保存”按钮 , 存储数据库关系图,可以看到“选择名称”对话框。输入数据库关系图的名称为“图书管理系统的数据库关系图”,如图 5-47 所示。

(6) 单击“确定”按钮,存储数据库关系图。在对象资源管理器中展开数据库“图书管理”→“数据库关系图”结点,可以看到新建的数据库关系图。

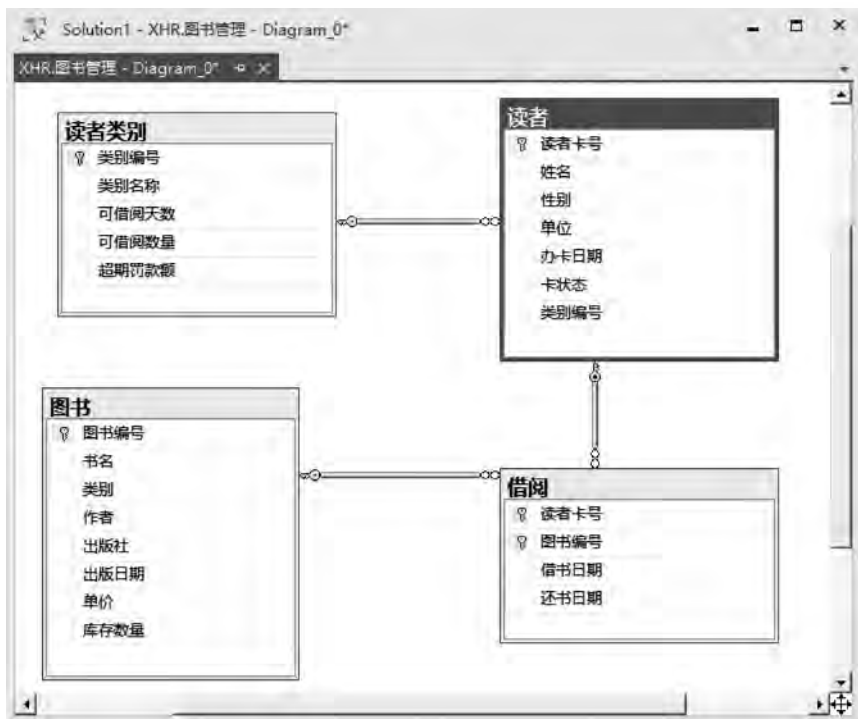


图 5-46 添加的数据库关系图

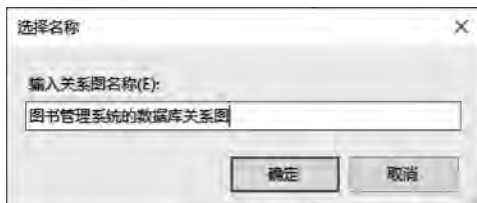


图 5-47 “选择名称”对话框

需要说明的是,如果各表之间没有设置外码约束,则图 5-46 的关系图窗口只包含图表中的表结构,各表之间没有连线,此时,需要通过在图表间拖拉字段和填写外键关系对话框的方法解决建立表之间的关联。

2. 修改和删除数据库关系图

要修改和删除一个数据库关系图,可以在 SSMS 中选中数据库关系图,右击要编辑的关系图,弹出快捷菜单。如果要修改关系图,选择快捷菜单的“修改”项,则会弹出关系图编辑框,在数据库关系图的空白部分右击打开快捷菜单命令,可以新建表或添加表。选数据表图标,可以编辑定义数据、创建约束和关系。如果要删除关系图,则选择“删除”项。

5.5.4 索引的创建和维护

数据库中的索引与书籍中的目录类似,一本书中,利用目录可以快速查找到需要的信息,无须阅读整本书;在数据库中,索引使数据库程序无须对整个表进行扫描,就可以在其

中找到需要的数据。当创建数据库并优化其性能时,应该为数据查询使用表的“列”创建索引。

索引(index)是一个单独的、存储在磁盘上的物理数据结构,是影响数据库性能的一个重要因素。在实际的数据库应用中,在表上创建和维护索引是一项重要的工作。通过建立表的索引,可以帮助数据库引擎在磁盘中定位记录数据,以便在数据表的庞大数据中快速找到数据。换句话说,创建数据表的索引可以提升 SQL 查询效率,让用户更快速地得到查询结果。一个数据表中可以针对不同的属性或属性组合建立不同的索引。

1. 索引的类型

从功能逻辑上讲,索引主要有四种,分别是普通索引、唯一索引、主键索引和全文索引。

普通索引是基础的索引,没有任何约束,主要用于提高查询效率。唯一索引就是在普通索引的基础上增加了数据唯一性约束,对一个数据表可以建立多个唯一索引。主键索引是在唯一性索引的基础上增加了不为空的约束,也就是 NOT NULL+UNIQUE。全文索引可以用专门的全文搜索引擎,如 ES(Elasticsearch)和 Solr。显然,普通索引、唯一索引、主键索引都是一类索引,只不过对数据的约束逐渐提升。

由于数据存储的文件中只能按照一种顺序进行存储,所以一张数据表最多只有一个主键索引,但可以有多个普通索引或多个唯一索引。

按照物理实现方式,索引可以分为聚集索引和非聚集索引。

聚集索引(clustered index)是指数据表的记录按索引字段的排序方式进行存储,也就是说,数据表中记录的物理顺序和索引文件中行的顺序是相同的,或者说索引文件中邻近的记录在数据表中也是临近存储的。由于数据行本身只能按一种顺序存储,因此,每个表只能创建一个聚集索引。

非聚集索引(nonclustered index)是指数据表中记录的物理顺序和索引文件中行的顺序不相同。由于非聚集索引不涉及数据表中数据按索引字段进行重新排序,即不决定和影响数据存储,只能用于查询,指出已存储记录的位置,因此可以对数据表创建多个非聚集索引。

创建索引时,可以先创建一个聚集索引,再创建非聚集索引。

2. 创建索引

不同的数据库管理系统提供了不同的索引类型,SQL Server 2019 中的索引主要有聚集索引和非聚集索引两种,两者的主要区别是在物理数据的存储方式上。

除了聚集索引和非聚集索引外,SQL Server 2019 中还提供了其他的索引类型,如唯一索引、全文索引、空间索引等。

下面简单介绍使用 SQL Server Management Studio 图形化界面方式创建索引的步骤。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中展开要创建索引的“图书”表结点。

(2) 右击“图书”表结点下面的“索引”结点,在弹出的快捷菜单中选择“新建索引”→“非聚集索引”命令,打开如图 5-48 所示的“新建索引”窗口。

(3) 在“新建索引”窗口的“常规”选项卡中,可以配置索引名称、索引类型、是否唯一、添加索引键列和包含性列、字段的排列顺序等,单击“添加”按钮,打开选择添加索引的列窗口,从中选择要添加索引的表中的列,如图 5-49 所示。这里选择图书表的“书名”列,表示在此列上添加索引。如果是创建复合索引,可再加选其他字段。



视频讲解

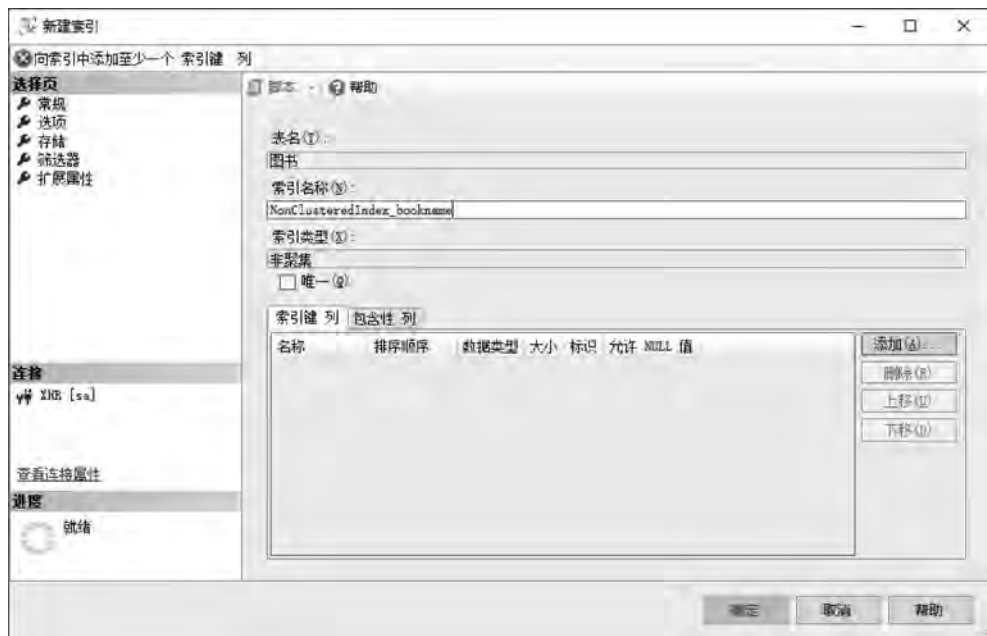


图 5-48 “新建索引”窗口

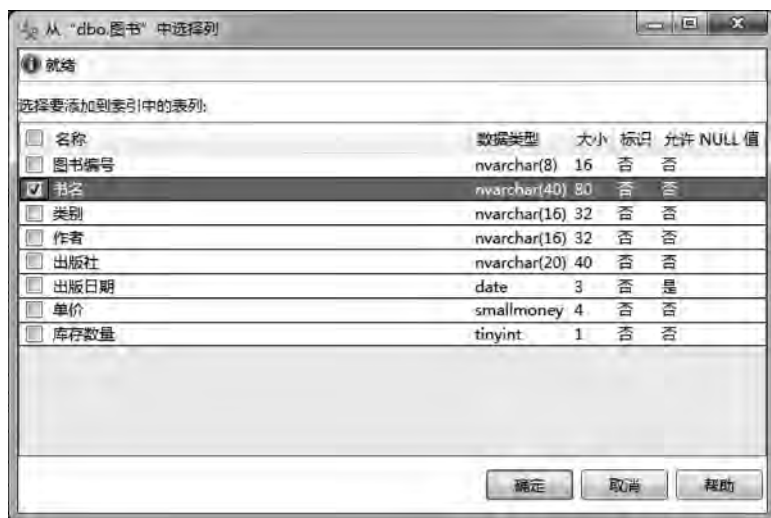


图 5-49 选择索引列

(4) 选择索引列后,单击“确定”按钮,返回“新建索引”窗口,如图 5-50 所示。单击该窗口中的“确定”按钮,返回对象资源管理器,可以在索引结点下面看到新建的索引,如图 5-51 所示,说明该索引创建成功。

3. 管理和维护索引

由于在数据表中进行录入、删除或更新操作时,会使索引页出现碎块,因此为了提高系统的性能,索引创建之后可以根据需要对数据库中的索引进行管理,包括修改索引、删除索引、显示索引信息、索引的性能分析和维护等。

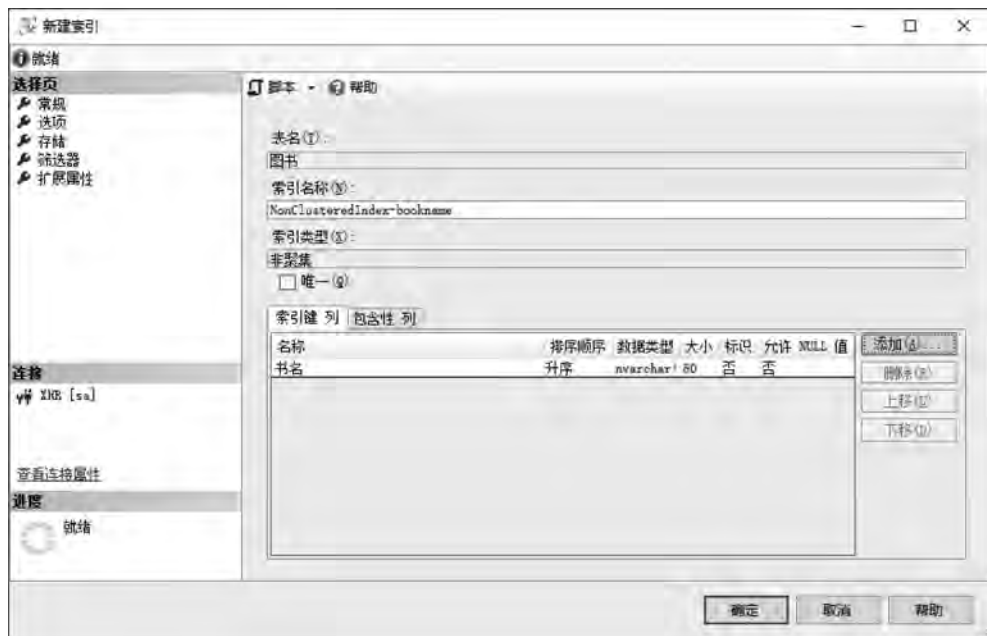


图 5-50 新建索引



图 5-51 创建的非聚集索引

修改索引就是更改索引属性,在 SSMS 的“对象资源管理器”窗口依次展开数据库“图书管理”→“图书”表→“索引”,右击已创建的非聚集索引 NonClusteredIndex_bookname,打开快捷菜单的“属性”命令,可以看到“索引属性”窗口,如图 5-52 所示。在窗口左边选择选项卡,在窗口右边字段修改数据表索引的相关属性,如更改“升序”“降序”排序顺序。

修改索引也可以在修改数据表字段定义数据时,执行“表设计器”→“索引/键”命令,打开“索引/键”对话框进行修改。

由于创建索引需要额外的磁盘空间和维护成本,当不再需要某个索引时,可以将其删除。

4. 数据表的索引规划

索引的建立由 DBA 或表的创建者负责,不管是主索引还是用户创建的索引,DBMS 都将自动维护索引,使其和基本表保持一致。DBMS 在存取数据时,会自动选择合适的索引作为存取路径。虽然建立索引可以提高查询速度,但过多建立索引会占用很多的磁盘空间,

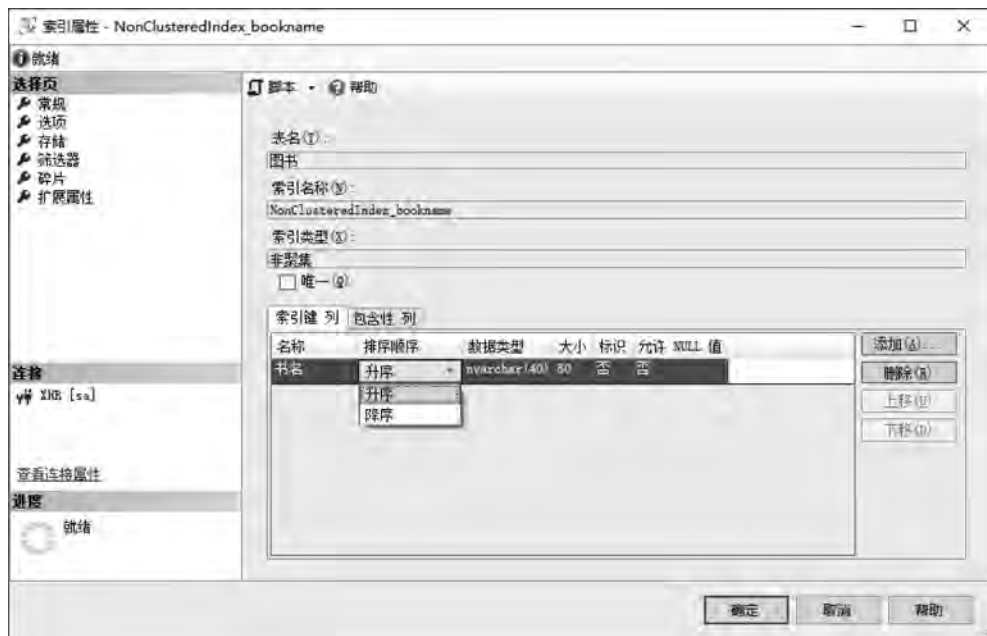


图 5-52 “索引属性”窗口

给维护带来麻烦。所以,需要权衡利弊,合理规划和使用索引。

1) SQL Server 建立索引的原则

建立索引是加快数据查询的有效手段。基本上,SQL Server 数据表的所有字段都可以选择创建索引(除了已经建立约束的字段),或作为索引的组成字段创建联合索引。在数据表中选择索引字段,就是判断指定字段是否应该创建索引加速查询。

(1) 根据查询要求建立索引。对于数据表中查询频率高、实时要求高、更新不频繁的字段,应该为这些字段创建索引,如主键、外键、经常需要连接查询的字段、排序、分组的字段和需要查询指定范围的字段。一般来说,数据表的主键建议创建聚集索引,SQL Server 默认将主键自动创建为聚集索引,但主键并不等同于聚集索引。

(2) 索引不是万能的,不合理的索引设计可能会阻碍数据库和业务处理的性能。对于数据表查询时很少引用到的字段、数据大量重复的字段和取值较少的字段,不应该为其建立索引。

(3) 大表应当建索引,小表则不必建索引。一般来说,如果数据表的数据量太小,索引能够改进数据访问的效率十分有限,可以不必创建索引。

(4) 对于一个基本表,不要建立过多的索引。使用索引可以快速定位需要查找的数据,但也存在一些不足,索引文件需要占用文件目录和存储空间,索引过多会使系统负担过重。索引需要自身维护,当基本表的数据增加、删除或修改时,索引文件随之变化,数据库引擎需要花费额外时间和资源来更新索引数据,因此索引需要额外的磁盘空间和维护成本。显然,索引过多会影响数据的增加、删除和修改的速度。

2) 创建索引的注意事项

SQL Server 创建索引前需要了解一些注意事项,即索引的约束,具体如下。

(1) 一个数据表最多只能有一个聚集索引和 999 个非聚集索引。

(2) 由于数据表的记录数据和聚集索引的顺序一致,因此每一个数据表只能创建一个聚集索引,但是可以在数据表的多个字段创建多个非聚集索引。

(3) 可以选择数据表的多个字段集合来创建联合索引(联合索引是指索引字段超过一个的索引),但联合索引最多只能有 16 个字段,按照最左优先的方式进行索引的匹配(最左匹配原则)。由于联合索引的索引字段尺寸通常比较大,需要更多的磁盘读取,因此会影响整体的执行性能。

3) 索引的作用

索引可以加快数据的访问,数据库引擎可以通过索引结构快速找到指定记录,能够让 SQL 的连接查询、排序操作更加有效。索引的作用主要表现在以下三个方面。

(1) 使用索引可以明显地加快数据查询的速度。对于数据文件大的基本表,如果没有建立索引,在进行数据查询时则需要将数据文件分块,逐个读到内存,进行查询的比较操作。而使用索引后,先将索引文件读入内存,根据索引项找到记录的地址,然后根据地址将记录直接读入计算机。由于索引文件小,只包含索引项和记录地址,一般可一次读入内存,并且索引文件的索引字段是经过排序的,可以快速找到索引字段值和记录地址。因此,使用索引大大减少了磁盘的 I/O 次数,从而加快查询速度。

(2) 使用索引可以加快数据库中基本表的连接速度。在进行关系的连接操作时,系统需要在连接关系中对每一个被连接字段进行查询操作,如果没有在连接字段上建立索引,则数据的连接操作速度非常慢,而在连接字段上建立了索引,则可大大提高连接操作。因此,许多系统要求连接文件必须有相应索引,才能执行连接操作。例如,要实现图书和借阅关系的连接操作,就要求在借阅表中必须在图书编号上建立索引,加快连接速度。

(3) 使用索引可保证数据的唯一性。索引的定义包含定义数据唯一性的内容。当定义了数据唯一的功能后,在对相关的索引项进行数据输入或更改时,系统需要进行检查,确保满足数据唯一性要求。

5.5.5 视图的创建和维护

视图(view)是从一张或多张表中导出的表,是用户查看数据库中数据的一种方式,其结构和数据内容建立在对表的查询基础上,由查询语句执行后得到的查询结果构成。与表一样,视图包含一系列带有名称的字段和记录数据,只是这些字段和记录来源于其他被引用的表或视图,通过视图看到的数据只是存放在基本表中的数据。所以,视图在数据库中并不是真实存在的,数据库中存在的只是视图的定义,视图是一个虚拟的表。

如果通过视图需要对看到的数据进行修改时,相应地,基本表的数据也要发生变化,如果基本表的数据发生变化时,从视图中查询出的数据也随之发生变化。

视图一经定义,就可以和基本表一样被查询和删除,并且可以在视图之上再定义新的视图。

1. 视图的创建、删除和修改

1) 创建视图

对于其他基本表来说,视图的作用类似于筛选。用户创建的视图可以基于表,也可以基于其他视图。在 SQL Server 2019 系统中,只能在当前数据库中创建视图,但定义视图的筛



选可以来自当前数据库或其他数据库中的一个或多个表,或者其他视图。

通过图形化工具 SQL Server Management Studio 创建视图的步骤如下。

(1) 启动 SQL Server Management Studio 窗口,在“对象资源管理器”中展开目标数据库“图书管理”→“视图”结点,右击“视图”结点,在弹出的快捷菜单中单击“新建视图”命令,打开如图 5-53 所示的“添加表”对话框,可从对话框中的四个选项卡中选择在新视图中包含的元素,在“表”选项卡中列出了用来创建视图的基本表,在列表框中选中读者表和读者类别表,单击“添加”按钮,然后关闭该对话框。



图 5-53 “添加表”对话框

(2) 打开“视图设计器”窗口,如图 5-54 所示。



图 5-54 “视图设计器”窗口

“视图设计器”窗口默认包含了三个区域,上方区域是“关系图”窗格,“关系”图窗格使用数据库关系图方式显示数据表,可以在此窗格中添加或删除表,选择视图中包含的列;中间区域是“条件”窗格,用于显示视图需要输出的列、排序类型、筛选条件等;下方区域是“SQL 语句”窗格,根据在“关系图”窗格和“条件”窗格中的设置,自动在“SQL 语句”窗格生成相应的 T-SQL 代码,用户也可以在此区域直接编写或修改视图定义中的查询语句。

(3) 在“关系图”窗格区域中,单击表中字段左边的复选框选择需要的字段,“条件”窗格自动显示输出,“SQL 语句”窗格生成相应的 T-SQL 代码,如图 5-55 所示。

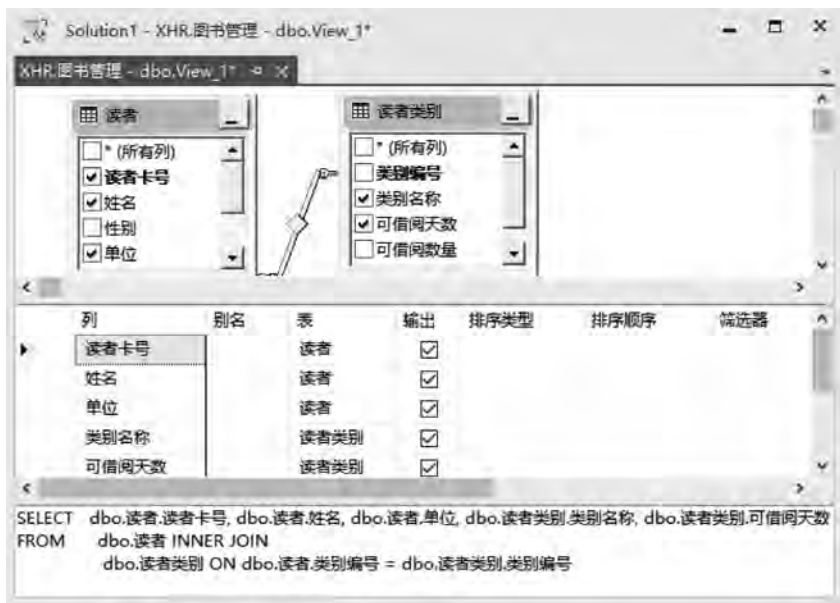


图 5-55 视图设计器中选择所需的列

(4) 单击工具栏中的“保存”按钮 , 打开“选择名称”对话框, 在输入视图名称下的文本框中输入视图名为 View_ddxx, 并单击“确定”按钮, 完成视图的创建。之后, 在数据库“图书管理”→“视图”结点下可以看到新创建的视图 View_ddxx。

(5) 右击创建的视图 View_ddxx, 单击快捷菜单“设计”, 打开视图设计器, 右击视图设计器的空白区域, 在弹出的快捷菜单中选择“执行 SQL”命令, 运行视图, 在视图设计器的底部区域的“结果”窗格中显示执行结果, 如图 5-56 所示。也可以在新建查询窗口中执行代码 `select * from View_ddxx`, 显示相同的结果。

可以看到, 不管是视图执行结果显示的列, 通过查询结果显示的列, 还是视图设计器选中的列都是相同的, 因此, 可以通过视图而不是基本表查询数据, 以满足不同用户对数据的不同需求。

2) 修改视图

创建好的视图也可以根据实际需要进行修改, 以满足新的需求。如果基表发生变化, 或者要通过视图查询更多的信息, 就可以根据需要修改视图的定义。

修改视图的方法和创建视图相同, 右击需要修改的视图, 在弹出的快捷菜单中选择“设计”命令, 打开视图设计器窗口, 可以按照创建视图的方法修改视图。



图 5-56 视图的执行结果

注意：由于视图可以被另外的视图作为数据源使用，所以修改视图时要小心。如果删除了某列输出，而该列正好在其他视图中使用，那么，在修改该视图后，其他关联的视图将无法再使用。

3) 删除视图

创建好的视图，如果不再需要，或想清除视图定义及与之相关联的权限，则可以删除该视图。删除视图后，表和视图基于的数据并不受影响，删除的是一个对象，因此任何基于已删除视图的查询将会失败。

删除视图同创建视图一样，也可以在 SQL Server Management Studio 的“对象资源管理器”窗口中，打开视图所在数据库结点，右击需要删除的视图，在弹出的快捷菜单中选择“删除”命令，在弹出的“删除对象”对话框中单击“确定”按钮即完成视图的删除。另外，还可以在该对话框中单击“显示依赖关系”按钮，查看删除该视图对数据库的影响。

2. 通过视图修改数据

视图除了可以用来查看数据外，还可以利用视图对创建它的数据源进行一定的修改，如插入新的数据记录、删除记录和更新记录，但通过视图添加、更新和删除数据时，与表相比有一些限制，要求所建的视图必须满足源数据表的完整性约束条件，关于视图更新的限制详见 6.2.4 节。

修改视图仍可以使用 SQL Server Management Studio 图形化界面方式和 T-SQL 语

句。通过图形化方式修改视图和创建视图类似,右击要修改的视图,在弹出的快捷菜单中选择“编辑前 200 行”命令,在打开的视图中直接添加、删除、修改数据。

总之,视图作为一种数据库对象,提供了数据的访问控制能力,即允许用户通过视图访问数据,但用户没有得到访问视图的基本表的权限,以防止用户非法存取数据,在一定程度上保护了数据的安全性。

3. 视图的用途

视图实现了数据库管理系统三级模式中的外模式,其主要作用表现在如下四个方面。

1) 视图能够简化用户的操作

视图是对 SQL 查询的封装,简化了用户对数据库的操作,因为定义视图的本身就是一个复杂的查询结果集,这样在每一次执行相同查询时,不必重写这些复杂的查询语句,只用一条简单的查询视图语句即可,隐藏了表与表之间的复杂连接操作。

2) 视图可以使用户以不同的方式(多个角度)看待同一数据

当多个不同用户共享数据库的数据时,视图能够实现让不同用户以不同的方式看到不同的或相同的数据集合。通过定义视图,使用户着重其感兴趣的数据,可以提高数据的查询效率。

3) 视图在一定程度上保证数据的逻辑独立性

对于视图的操作,例如,查询只依赖于视图的定义,当创建视图的基本表需要修改时,只需要修改视图定义中的子查询部分,而基于视图的查询不用改变,从而使外模式保持不变,原有的应用程序通过视图重载数据,保证数据的逻辑独立性。

4) 视图可以对机密的数据提供安全保护

视图可以作为一种安全机制。在设计数据库应用系统时,针对不同的用户开放不同的数据查询权限,则可通过为不同用户定义不同的视图,使用户只能看到与自己有关的数据,其他数据库或表不可见也不可访问,视图引用表的访问权限与视图权限的设置互不影响,这在一定程度上也保证了数据表的安全性。

5.5.6 存储过程的创建和维护

存储过程(stored procedure, SP)是数据库中运用十分广泛的一种数据库对象,已成为数据库管理人员的重要工具。存储过程实际上就是一组预先编译好的、为了实现某种特定功能的 T-SQL 语句集合,包括查询、插入、删除和更新等操作,经编译后以名称的形式存储在 SQL Server 服务器端的数据库中,而不是单独的文件中。用户可以使用应用程序或其他 SQL 脚本通过指定存储过程的名称调用并执行存储过程中的语句,类似于应用程序中的程序调用子程序。对于需要经常执行的操作,可以将所需的 SQL 语句都编写在一个存储过程中,然后由前台应用程序执行存储过程即可。

1. 存储过程的优点

一般来说,客户端程序有两种方式执行 T-SQL 语句。其一是在客户端创建应用程序后,使用 ADO.NET 等组件发送 T-SQL 语句到 SQL Server,然后在 SQL Server 数据库引擎执行 T-SQL 语句。其二是 SQL Server 先将欲执行的 T-SQL 语句创建为存储过程,客户端程序可以直接执行位于 SQL Server 的存储过程。

在客户端执行存储过程与存储在客户机的本地 T-SQL 语句相比,具有以下四个优点。

1) 改善系统性能,提高执行效率

存储过程只在创建时编译一次,就可多次执行,减少了编译花费的时间,当重复执行时,因为不需要重新编译,提高了执行 T-SQL 语句的效率,从而改善系统的性能。系统在创建存储过程时会对其进行分析和优化,并将编译生成的执行计划驻留于系统缓存中,用户使用时,会直接从缓存中读取(除非用户有重新编译的需求),以加速存储过程的后续执行。

2) 减少了网络通信的数据流量

创建存储过程后,网络中传送的只是调用存储过程的语句,而不是从客户端发送数百行的 SQL 语句,客户端应用程序只需一条语句就可执行位于 SQL Server 服务器端的存储过程,实现相同的功能,减少了客户端与服务器网络传送过程中的数据流量。很多情况下,数据库应用系统的设计者用存储过程实现复杂的查询和统计,只将最终的处理结果返回给客户端,从而免去客户端的大量编程。

3) 提供了一种安全机制

存储过程作为 SQL Server 的数据库对象,存放在数据库中且在服务器端运行;对于不允许用户直接操作的表或视图,可通过授予存储过程的执行权限间接地访问这些表或视图,避免非授权用户访问在定义存储过程中被引用对象的数据,达到一定程度的安全性。拥有参数的存储过程还可以保护客户端应用程序的安全性,相比单纯使用 T-SQL 代码,能够有效防止 SQL 注入式攻击,降低黑客攻击 SQL Server 服务器的可能性。

4) 模块化的程序设计

存储过程可实现模块化的程序设计,一个存储过程就是一个模块,可用来封装并实现特定的功能,实现业务逻辑较为复杂的应用。也就是说,可将经常执行的 T-SQL 语句或复杂的业务操作编写为多个存储过程的模块,放在数据库中,供用户多次调用,为用户提供统一的数据库访问接口,进而改进应用程序的部署和可维护性。而业务操作对用户是不可见的,达到业务操作封装的效果,提高程序的重用和可移植性,并且存储过程与调用该存储过程的应用程序相分离,减少了应用程序和数据库之间的耦合。

2. 创建存储过程

在 SQL Server 2019 系统中,存储过程主要分为系统存储过程和用户定义存储过程。

系统存储过程由 SQL Server 2019 自身提供,用户可以直接使用。使用系统存储过程完成数据库服务器的管理工作,为系统管理员提供帮助,为用户了解系统信息、查看数据库对象提供方便。系统存储过程位于数据库服务器中,并且以 sp_ 开头,其定义在系统数据库和用户定义的数据库中。SQL Server 2019 服务器中的许多管理工作都是通过执行系统存储过程完成的,许多系统信息也可以通过执行系统存储过程获得,由于系统存储过程在服务器启动后被加载到系统缓存中,因此执行效率高。

启动 SQL Server Management Studio,并连接到 SQL Server 2019 中的数据库,在“对象资源管理器”中展开“服务器”→“数据库”→“图书管理”→“可编程性”→“系统存储过程”,可以看到系统提供的存储过程。

用户定义存储过程是用户根据数据管理和某一特定业务需求需要创建的存储过程。即在用户数据库中,用户使用 T-SQL 语句编写的实现某一特定业务需求 SQL 语句集合。用户自定义存储过程可以接受输入参数、向客户端返回结果和信息、返回输出参数等。本节主要介绍用户定义存储过程。

使用 SQL Server Management Studio 窗口创建存储过程的步骤如下。

(1) 打开 SQL Server Management Studio 窗口,在“对象资源管理器”中依次展开要创建存储过程的“图书管理”数据库中的“可编程性”→“存储过程”结点。右击“存储过程”结点,从弹出的快捷菜单中选择“新建存储过程”命令。

(2) 选择“新建存储过程”命令后,打开“创建存储过程的查询编辑器”窗口。在该窗口中,系统自动生成有提示的 CREATE PROCEDURE 语句的存储过程模板,用户可以根据模板提示补充输入存储过程包含的相应语句。当然,也可以直接输入 T-SQL 语句。

说明:因为 SSMS 创建存储过程就是编译和执行 SQL 脚本文件,换句话说,可以直接单击“新建查询”按钮,打开查询编辑窗口后自行输入 CREATE PROCEDURE 语句创建存储过程。本质上,使用 SSMS 创建存储过程与直接使用 CREATE PROCEDURE 语句创建存储过程是一样的,只是有些参数可以用模板添加而已。

(3) 在此根据需要创建名称为 pro_dzjyxx 的无参数存储过程,实现读者借阅信息的功能。在“创建存储过程的查询编辑器”窗口中,输入创建存储过程的 SQL 语句(相关 SQL 语句在第 6 章详细介绍)。

(4) 单击“SQL 编辑器”工具栏上的“分析”按钮,检查输入的 T-SQL 语句是否有语法错误,确认无错误后,单击“执行”按钮完成存储过程的创建。在数据库“图书管理”→“可编程性”→“存储过程”下可以看到已创建的 pro_dzjyxx 存储过程。

3. 执行存储过程

在 SQL Server 中,打开 SQL Server Management Studio 窗口,在“对象资源管理器”中依次展开要创建存储过程的“图书管理”数据库中的“可编程性”→“存储过程”结点。在 pro_dzjyxx 存储过程上,执行右键快捷菜单的“执行存储过程”命令后,可以看到“执行过程”对话框。在“执行存储过程”窗口中(如果是带参的存储过程,输入执行所需的参数),单击“确定”按钮即可执行存储过程,可以看到运行结果。

4. 管理存储过程

对于现有的存储过程,可以使用 SSMS 管理存储过程,管理存储过程包括修改存储过程、删除存储过程、查看存储过程和重命名存储过程。其操作步骤是在“对象资源管理器”窗口的存储过程中,执行右键快捷菜单的对应命令,完成相应的操作。管理存储过程类似于创建触发器的步骤,在此不再阐述。

5.5.7 触发器的创建和维护

触发器(trigger)是针对数据表的特殊存储过程,主要是通过事件触发而被自动执行的程序。它是一个功能强大的工具,数据库管理员可以用它在修改数据时自动执行所需操作。

触发器和存储过程都是由 SQL 语句和流程控制语句组成的。触发器也是一种特殊的存储过程,一般的存储过程通过过程名直接调用,而触发器的特殊性表现在:它是在执行某些特定的 T-SQL 语句时通过事件触发而被执行,它与表紧密相连,当用户对表中的数据进行插入、删除和修改时,触发器就会自动激活执行定义的 SQL 语句,以保证数据的完整性和一致性。所以,不能像存储过程一样,由用户自行执行触发器。

1. 触发器的作用

SQL Server 2019 中,可以用约束和触发器两种方法强制执行业务规则和保证数据的

完整性。但触发器可以实现比约束更为复杂的数据完整性约束,实现由主键和外键不能保证的复杂的参照完整性和数据的一致性,能够对数据库中的相关表实现级联更改,提供比 CHECK 约束更为复杂的数据完整性,并自定义错误信息。其主要作用如下。

1) 强制数据完整性

触发器可以实现比约束更为复杂的数据约束。在数据库中要实现数据的完整性约束,可以使用 CHECK 约束或触发器实现。触发器包含了使用 T-SQL 语句的复杂处理逻辑,不仅支持约束的所有功能,还可以实现更为复杂的数据完整性约束。例如,在 CHECK 约束中不允许引用其他表中的列完成检查工作,而触发器可以引用其他表中的列完成数据的完整性约束。

2) 自动执行

触发器不需要用户调用,是通过事件触发而自动执行,只要对表中的数据进行了修改,触发器就会立即被激活。

3) 实现数据库中多张表的级联更改

触发器虽然是基于一个表建立的,但可以对多个表进行操作,用户可以通过触发器对数据库中的相关表进行级联修改,而且可以评估数据修改前后表的状态,并根据其差异采取对策。例如,对读者表创建一个级联删除的触发器,要求删除读者表中记录时要把借阅表中相应的借阅记录也删除。

4) 维护非规范化数据

用户可以使用触发器来保证非规范数据库中的低级数据的完整性。维护非规范化数据与表的级联是不同的。表的级联是指不同表之间的主外键关系,维护表的级联可以通过设置表的主键与外键的关系来实现。而非规范数据通常是指在表中派生的、冗余的数据值,维护非规范化数据应该通过使用触发器来实现。

5) 返回自定义的错误信息

约束是不能返回信息的,而触发器可以。例如,当插入一个违背完整性约束的数据值时,可以返回一个具体的友好的错误信息给前台应用程序。

2. 触发器的分类

在 SQL Server 2000 支持 DML 触发器,在 2005 版增加了 DDL 触发器。目前,SQL Server 支持的触发器有三种: DML 触发器、DDL 触发器和 LOGON 触发器,说明如下。

1) DML 触发器

DML 触发器是当数据库服务器中发生数据操纵语言(data manipulation language, DML)事件时执行的存储过程,DML 事件包括在指定表或视图中修改数据的 INSERT 语句、UPDATE 语句或 DELETE 语句。DML 触发器依据触发时机不同又分为 AFTER 触发器和 INSTEAD OF 触发器。

(1) AFTER 触发器。AFTER 触发器又称后触发器,当执行 INSERT、UPDATE 和 DELETE 命令且记录已经改变完后(AFTER),才被激活执行,主要用于记录变更后的处理或检查,如果有错误,更改的数据可以恢复至更改前的值。需要注意的是,AFTER 触发器只能定义在表上,不能在视图上定义 AFTER 触发器。

(2) INSTEAD OF 触发器。INSTEAD OF 触发器又称前触发器,在记录变更之前被触发,并取代原来 SQL 语句里变动数据的操作(INSERT、UPDATA 和 DELETE 语句),而

去执行触发器本身定义的操作。这类触发器一般用来验证数据或替换原本需要执行的操作,也就是说,用触发器程序取代相应的操作。

2) DDL 触发器

DDL 触发器是在响应数据定义语言(data definition language,DDL)事件时执行的存储过程,主要包括 CREATE、ALTER、DROP 语句,用于执行数据库的管理工作,如防止数据库表结构的修改、审核和规范数据库操作。

3) LOGON 触发器

LOGON 触发器可以响应 LOGIN 事件跟踪登录活动、限制登录 SQL Server 或特定登录的会话数,也就是说,LOGON 触发器可以防止非法用户成功连接数据库引擎。

3. 创建触发器

触发器的创建和存储过程的创建类似,使用 SQL Server Management Studio 对象资源管理器,创建触发器的步骤如下:

在对象资源管理器中依次展开“数据库”→“表”→“要创建触发器的表结点”→“触发器”,右击“触发器”,在弹出的快速菜单中选择“新建触发器”命令,打开“创建触发器的查询编辑器”窗口,在查询编辑器窗口中显示“触发器”模板,用户根据模板输入触发器的 T-SQL 语句,当语句执行成功后,触发器创建成功。

因为在 SSMS 中新建触发器和存储过程的步骤类似,这里不再说明其创建步骤。

4. 管理触发器

对应现存的触发器,可以使用 SSMS 管理触发器,管理触发器包括修改触发器、删除触发器、查看触发器、启用和禁用触发器。

当触发器不满足需求时,可以修改触发器的定义和属性。当触发器不再需要使用时,可以将其删除,删除触发器不会影响其操作的数据表。如果触发器创建之后暂时不需要使用,可以将其禁用。禁用后,当用户执行触发操作时,触发器不会被调用,但它仍作为对象存储在当前数据库中。

习题 5

1. 简答题

- (1) 简述组成 SQL Server 2019 物理数据库的文件类型及其作用。
- (2) SQL Server 2019 有哪些系统数据库? 它们的作用是什么?
- (3) 什么是基本表? 什么是视图? 二者有什么区别与联系?
- (4) 简述索引和视图的主要作用。
- (5) 客户端程序通常使用哪两种方式执行 T-SQL 语句? 存储过程的优点是什么?
- (6) 什么是触发器? 目前 SQL Server 支持哪几种触发器? 触发器有什么作用?

2. 上机操作题

- (1) 创建数据库。

使用 SQL Server Management Studio 创建一个名为 XSXK 的数据库(学生选课数据库系统),要求包含三个数据文件,其中,主数据文件初始大小为 20MB,最大限制为 50MB,每次增长 2MB;次要数据文件为 10MB,最大大小不受限制,每次增长 20%;事务日志文件为

20MB,最大大小为 50MB,每次增长 1MB。

(2) 创建表。

使用 Server Management Studio 的对象资源管理器为 XSXK 数据库创建三张表,分别为学生表、课程表、选课表,各表的结构要求如表 5-7~表 5-9 所示。

表 5-7 学生表的结构

字段名称	数据类型	大 小	备 注
学号	char	10	主键
姓名	char	8	非空
性别	char	2	
出生日期	date		
所在院系	char	30	
联系电话	char	13	

表 5-8 课程表的结构

字段名称	数据类型	大 小	备 注
课程编号	char	10	主键
课程名称	char	16	非空
学分	smallint		
任课教师	char	8	
先行课	char	10	

表 5-9 选课表的结构

字段名称	数据类型	大 小	备 注
学号	char	10	主属性,外键
课程编号	char	10	主属性,外键
成绩	smallint		

(3) 通过图形化的对象资源管理器为学生表的姓名建立名称为 Uni_xm 的唯一性索引,在姓名和所在院系字段上建立名称为 Ind_xmbj 的非聚集组合索引。

(4) 建立性别只能为“男”和“女”的约束,成绩取值范围为 0~100 的约束。

(5) 在 SQL Server Management Studio 管理工具下,分别为 XSXK 数据库中的学生表、课程表、选课表中输入若干条记录,然后删除和修改特定记录。注意实体完整性和参照完整性约束对数据更新的影响。

(6) 创建视图。

① 创建一个名为 view_xs 的视图,用来查看数计学院所有学生的信息。

② 创建一个名为 view_xk 的视图,用来查看每名学生的选课信息,只显示学生学号、课程名和成绩。