

基础知识篇

- ◆ 模块 1 数据库概述
- ◆ 模块 2 表和表中数据的操作
- ◆ 模块 3 数据完整性
- ◆ 模块 4 索引和视图
- ◆ 模块 5 T-SQL 程序设计
- ◆ 模块 6 存储过程和触发器

1

模块 1

数据库概述

▶ 学习要点

- 数据库的相关概念。
- 关系数据库的相关概念。
- 数据库的创建。

1.1 数据库基础知识

随着计算机技术和通信技术的日益发展,人们使用的大量数据信息不断增长,而对于某些重要的数据信息,为便于以后再次使用,就需要对它进行存储,数据库正是用于存储数据信息的。可以把数据库比作衣柜,把数据比作衣服,衣柜正是用于存放衣服的;为便于在不同的季节查找衣服,就需要对衣服进行分类存放和管理,于是就产生了管理系统,在数据库中称为数据库管理系统;而衣柜中的衣服找出来后用于穿着,这就属于衣服的应用范畴,在数据库中称为数据库应用系统。

1.1.1 数据库系统中的相关概念

1. 数据

数据(data)是用于描述事物的符号记录,是数据库中存储的基本对象,也是最终用户操作的基本对象。例如,要将一个人的属性特征存放在计算机中,可以将这些特征整理为姓名、年龄、身高、体重、性别等,这些信息就是用于描述这个人的符号记录,将其存放在计算机

中,把它们称为数据。又如,将拍摄的照片、录制的小视频、制作的音乐等存放在计算机中,它们也都称为数据。

2. 数据库

数据库(database,DB)是长期存储在计算机内、有组织的和可共享的数据集。数据库为数据创造了一个存放的场所。例如,在登录 QQ 时,需要输入账号和密码,只有账号和密码正确才能登录成功,但是 QQ 系统怎么才能验证账号和密码的正确性呢? 其实,QQ 系统本身有个数据库用于存放账号和密码等用户信息,只有输入的账号和密码与数据库中存放的账号和密码一致时,才能成功登录。

3. 数据库管理系统

数据库管理系统(database management system,DBMS)用于管理数据库的一切操作,包括定义、查询、更新等。例如,在登录 QQ 时,要进行账号和密码的匹配,这时就需要读取数据库中的账号和密码信息,而该操作就用到了数据库中的查询操作;又如,用户将密码进行了更改,需要将改后的密码存放到数据库中,这时就需要使用数据库的更新操作。

4. 数据库应用系统

数据库应用系统(database application system,DBAS)是在数据库管理系统(DBMS)支持下建立的计算机应用系统。例如,使用存放在数据库中的账号和密码登录 QQ 系统。

5. 数据库系统

数据库系统(database system,DBS)由数据库(DB)、数据库管理系统(DBMS)、数据库应用系统(DBAS)组成,如图 1-1 所示。

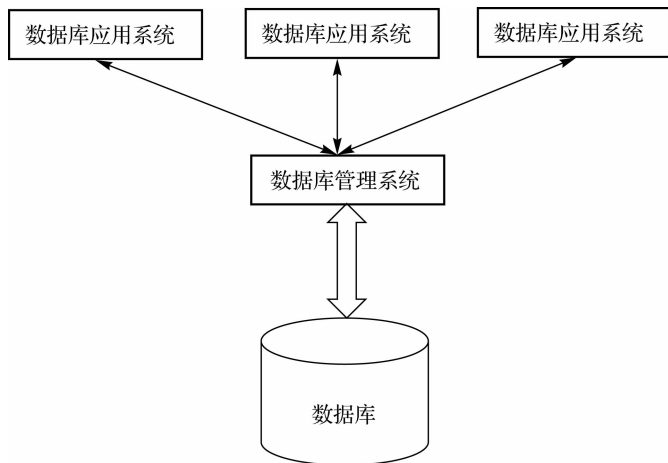


图 1-1 数据库系统

1.1.2 数据库模型

数据库模型是一种用于描述数据库是如何被建立和使用的理论与规范,主要包括层次、网状和关系。关系模型是最为常用的数据库模型,本书将以关系模型为主要介绍对象。

1. 关系

关系是一种特殊的二维表,是用二维表表示实体与实体之间的联系,见表 1-1。在关系模型中,数据的逻辑结构是一张二维表,它由行和列组成。一种关系对应一张二维表。

表 1-1 学生表

学 号	姓 名	性 别	班 级	年 龄
1703110101	李明	男	1	19
1703110102	张晓晓	女	1	19
1703110103	陈爱林	男	1	20

2. 记录

表中的一行(除了表头那一行)称为一条记录,表 1-1 中共有 3 条记录。

3. 字段

表中的一列称为一个字段,每一列的列名称为字段名。表 1-1 中共有 5 个字段,每个字段表明了其存储数据的类型及范围。

4. 主键

主键是唯一标识每一行记录的字段。主键确保了每一行记录的唯一性,使每一行记录都不相同。表 1-1 中的学号字段即该表的主键。主键具有两个特性:主键列值唯一和主键列值不为空。

1.2 SQL Server 2016 数据库

SQL Server 是一种广泛应用于网络业务数据处理的关系型数据库管理系统。SQL Server 2016 是微软公司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。其可提供对企业基础架构最高级别的支持,专门针对关键业务应用的多种功能与解决方案提供最高级别的可用性,并提供更多、更全面的功能以满足不同人群对数据的需求,包括支持来自不同网络环境的数据交互、全面的自助分析等创新功能。

1.2.1 SQL Server 2016 的版本及其对应功能

SQL Server 2016 可提供更全面、更安全可靠的网络云数据处理、存储与传输和动态迁移等功能,可以构建和管理用于业务数据处理的高可用性和高性能的应用与服务。

SQL Server 2016 共有 4 个版本,包括企业版、标准版、精简版和开发者版,类似于 SQL Server 2014,其中开发者版和精简版是免费的。新版数据库系统对云数据管理等多方面进行了优化。其中,开发者版包含了企业版的全部功能,可免费下载使用,但只限于开发和测试,不可用于产品环境或使用产品数据。

SQL Server 2016 的主要版本及其对应功能如下。

1. 企业版

企业版提供了全面的高端数据中心功能,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。

2. 标准版

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。

3. 精简版

精简版是入门级的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,可升级到更高级的数据库功能,具备所有可编程性功能,但需在用户模式下运行,并具有快速零配置安装和必备组件要求较少的特点。

4. 开发者版

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,包含企业版的所有功能,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,是构建和测试应用程序人员的理想之选。

1.2.2 SQL Server 2016 的组成结构

1. SQL Server 的总体结构和组件

SQL Server 2016 的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(reporting services, RS)、报表服务(integration services, IS)及主数据服务(master data services, MDS)等。

数据库引擎是整个 SQL Server 的主要核心,其他所有组件都与其有着密不可分的联系。SQL Server 数据库引擎有四大组件:协议(protocol)、查询引擎(query compilation and

execution engine)、存储引擎(storage engine)和 SQLOS(user mode operating system)。各客户端提交的操作指令都与这 4 个组件交互。

在 SQL Server 2016 中,常用的五大服务器组件及其对应的主要功能如下。

(1)数据库引擎。数据库引擎用于业务数据的存储、处理、查询和安全管理等操作,是系统最核心的组件,包括创建数据库和表,执行各种数据查询,访问数据库等,常用于调用数据库系统及有关操作。

(2)分析服务。分析服务的主要功能是提供商务智能解决方案,以及多维分析(联机分析处理 OLAP)和数据挖掘,支持用户建立数据仓库和商业智能分析。由数据库引擎负责多维分析,利用 AS,设计、创建和管理包含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,可更全面地理解业务数据。还可完成数据挖掘模型的构造和应用,实现知识发现、表示、管理和共享。

(3)报表服务。报表服务通过提供基于服务器的报表平台,为各种数据源提供支持 Web 的企业级报表功能。用户可方便地定义和发布满足需求的报表,可轻松实现报表的布局格式及数据源,可极大地满足企业管理人员的高效规范管理需求。例如,在航空公司的机票销售信息系统中,使用 SQL Server 提供的服务可方便地生成 Word、PDF、Excel、XML 等格式的报表。

(4)集成服务。集成服务主要用于生成企业级数据集成和数据转换解决方案的平台,是从原来的数据转换服务派生并重新以 .NET 改写而成的,可实现有关数据的提取、转换、加载等。例如,对于分析服务,数据库引擎是一个重要的数据源,将其中的数据进行适当处理并加载到分析服务中,可进行各种分析处理。IS 可高效地处理各种类型的数据源,包括处理 Oracle、Excel、XML 文档、文本文件等数据源中的数据。

(5)主数据服务。主数据服务针对主数据管理解决方案,可配置其管理任何领域(产品、客户和账户),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,也可用于管理数据的处理 Excel 的外接程序,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,其共同构成完整的服务架构。

2. SQL Server 2016 主要管理工具

在实际应用中,经常使用的 SQL Server 2016 主要管理工具如下。

1)SQL Server Management Studio

SQL Server Management Studio 用于访问、配置、管理和开发 SQL Server 组件的集成环境,使各种技术水平的开发人员和管理人员都能使用 SQL Server。

2)SQL Server 配置管理器

SQL Server 配置管理器为 SQL Server 服务、服务器协议、客户端协议和客户端别名提供基本配置管理。

3) SQL Server Profiler 事件探查器

SQL Server Profiler 事件探查器提供了一个图形用户界面,用于监视数据库引擎实例或 Analysis Services 实例。

4) 数据库引擎优化顾问

数据库引擎优化顾问可以用于协助创建索引、索引视图和分区的最佳组合。

5) 数据质量客户端

数据质量客户端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。其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。

6) SQL Server Data Tools 数据工具

SQL Server Data Tools(SSDT)数据工具提供 IDE,以便为 AS、RS 和 IS 商业智能组件生成解决方案。其包含数据库项目,为数据库开发者提供集成环境,以便在 Visual Studio (VS)中为 SQL Server 平台(包括本地和外部)执行其所有数据库设计工作,开发者可用 VS 中功能增强的服务器资源管理器,创建或编辑数据库对象和数据或执行查询操作。

7) 连接组件

连接组件安装用于客户端和服务端之间通信的组件,并用于 DB-Library、ODBC 和 OLE DB 的网络库。

1.2.3 安装 SQL Server 2016

常用 SQL Server 2016 安装向导进行安装。SQL Server 2016 安装向导提供了一个用于安装所有组件的功能树,便于根据需要分别安装这些组件。

SQL Server 2016 常用的安装步骤如下。

(1)插入 SQL Server 2016 安装光盘,然后双击根文件夹中的 Setup. exe 文件。若要从网络共享进行安装,则找到共享中的根文件夹,然后双击 Setup. exe 文件。

(2)安装向导将运行 SQL Server 安装中心。若创建新的 SQL Server 安装,选择左侧导航区域中的“安装”选项,然后选择“全新 SQL Server 独立安装或向现有安装添加功能”选项,如图 1-2 所示。

(3)对于初次安装的用户,选择第一个选项“全新 SQL Server 独立安装或向现有安装添加功能”,进入“产品密钥”界面,在该界面中可以输入购买的产品密钥。如果使用的是体验版本,可以在“指定可用版本”下拉列表框中选择 Evaluation 选项,然后单击“下一步”按钮,如图 1-3 所示。

(4)打开“许可条款”界面,选中“我接受许可条款”复选框,然后单击“下一步”按钮,如图 1-4 所示。



图 1-2 “SQL Server 安装中心”窗口

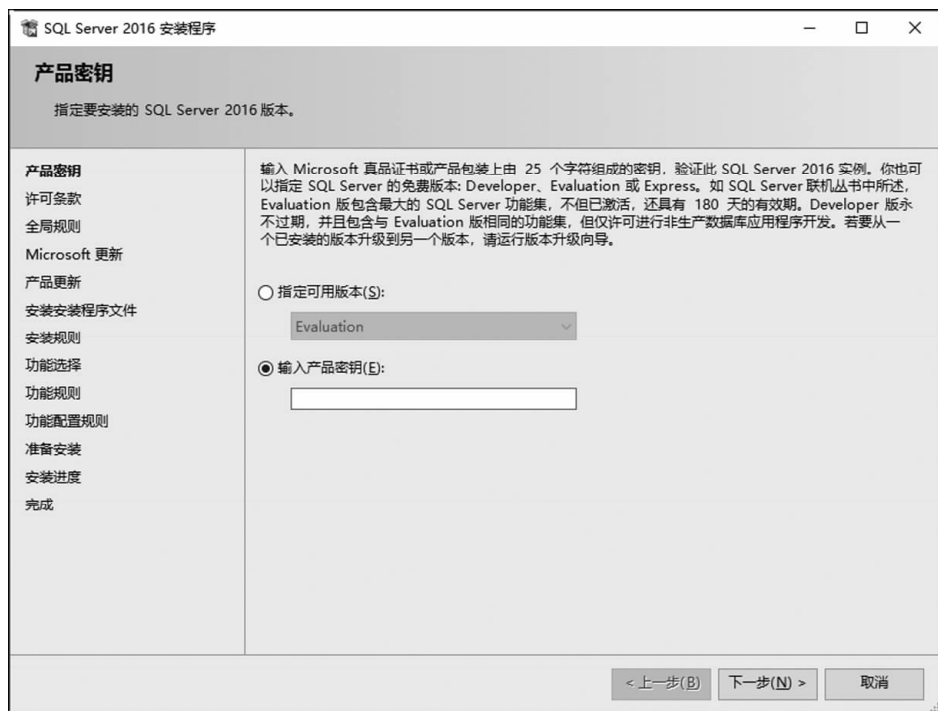


图 1-3 “产品密钥”界面



图 1-4 “许可条款”界面

(5) 安装程序将对系统进行常规检测, 如图 1-5 所示。如果系统缺少组件, 直接在官方网站下载后安装即可。如图 1-6 所示, 选中“使用 Microsoft Update 检查更新”复选框, 然后单击“下一步”按钮。



图 1-5 安装程序检测系统



图 1-6 使用 Microsoft Update 检查更新

(6)操作自动完成后,如图 1-7 所示,单击“下一步”按钮,打开“功能选择”界面。如果需要安装某项功能,则选中对应的功能名称前面的复选框,也可以通过单击下面的“全选”或“取消全选”按钮来选择。为了以后学习方便,这里单击“全选”按钮,如图 1-8 所示。

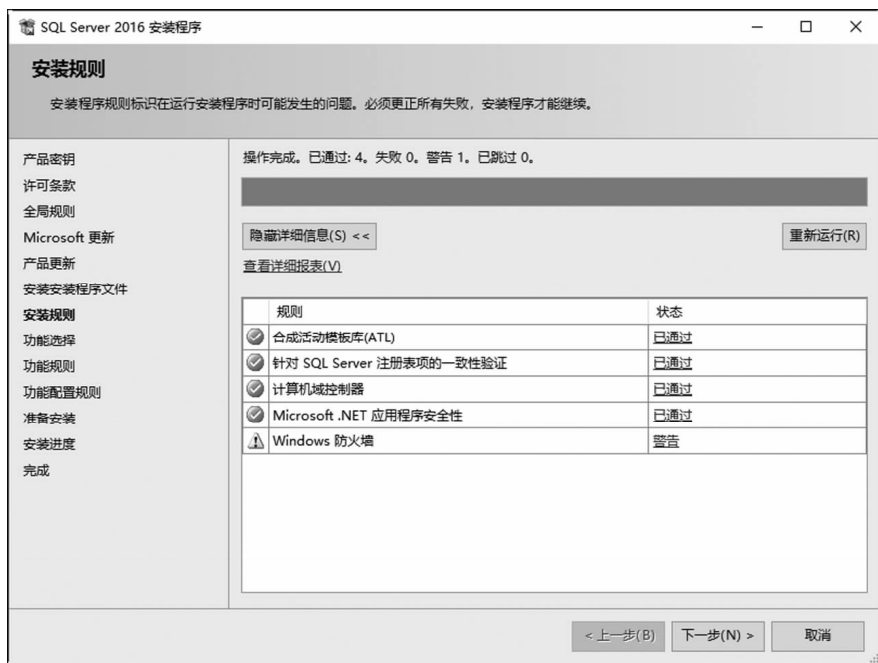


图 1-7 “安装规则”界面

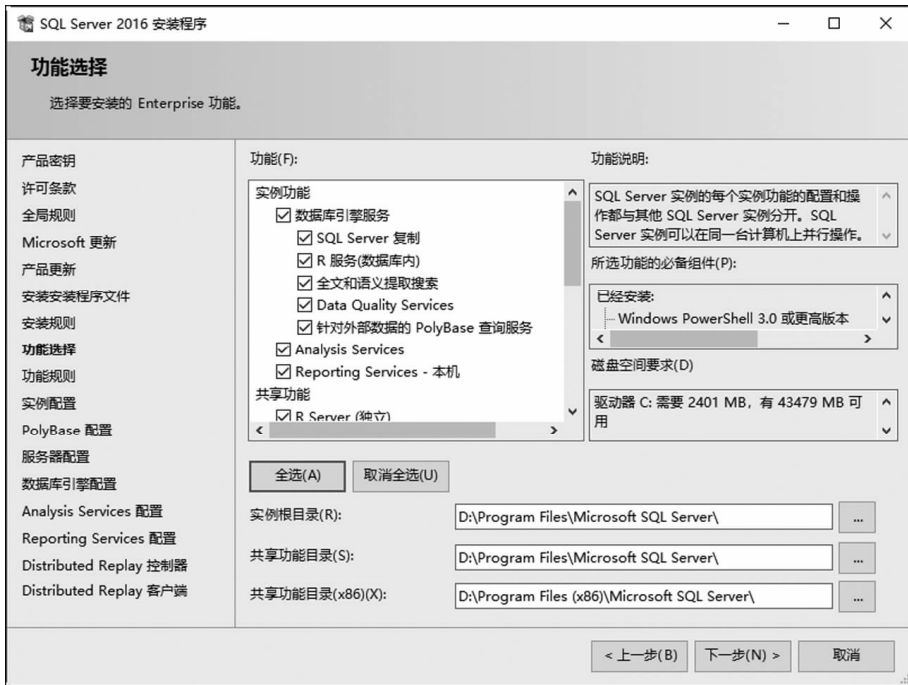


图 1-8 “功能选择”界面

(7)单击“下一步”按钮,打开“实例配置”界面,在安装 SQL Server 的系统中可以配置多个实例,每个实例必须有唯一的名称,这里选中“默认实例”单选按钮,如图 1-9 所示。

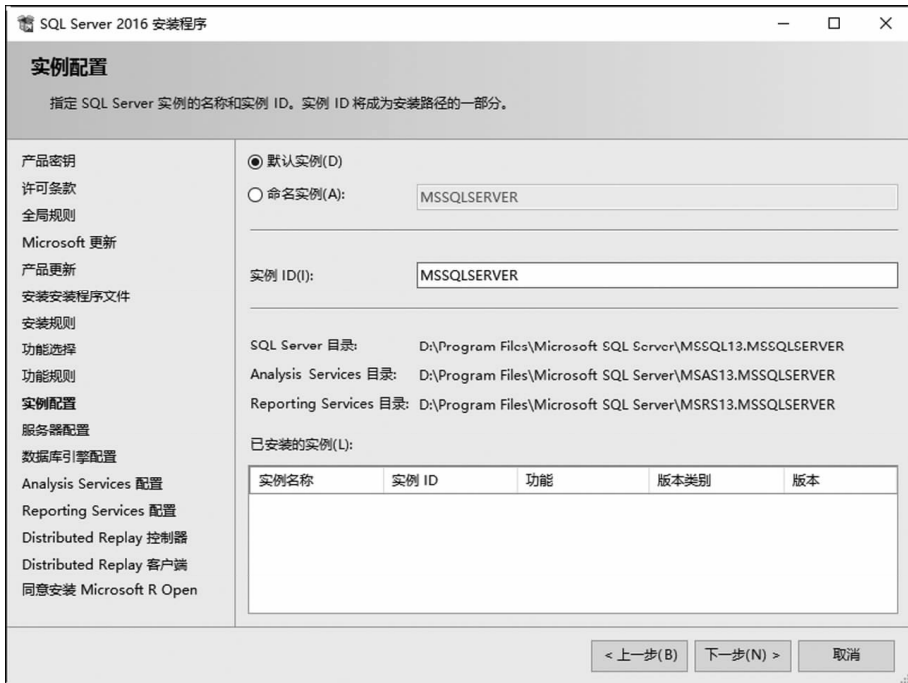


图 1-9 “实例配置”界面

(8)单击“下一步”按钮,打开“服务器配置”界面,该步骤设置使用 SQL Server 各种服务的用户,如图 1-10 所示。

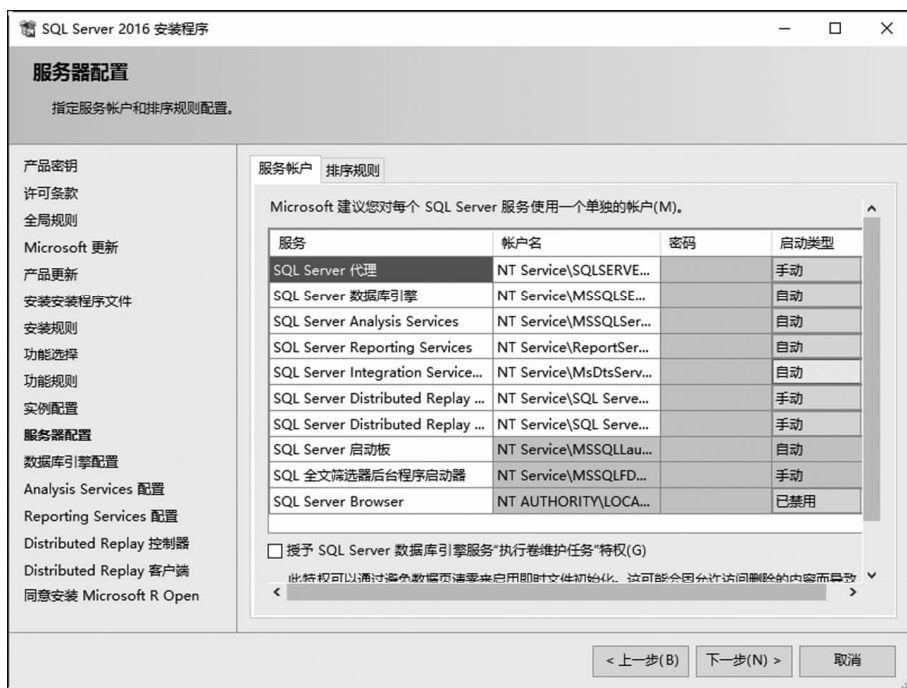


图 1-10 “服务器配置”界面

(9)单击“下一步”按钮,打开“数据库引擎配置”界面,在“服务器配置”选项卡中设置 SQL Server 的身份验证模式。

这里有两种身份验证模式,如果选择“混合模式”,需要为 SQL Server 的系统管理员设置登录密码,之后可以使用两种方式登录 SQL Server。

这里选择“Windows 身份验证模式”,然后单击“添加当前用户”按钮,将当前用户添加为 SQL Server 管理员,如图 1-11 所示。

(10)单击“下一步”按钮,打开“Analysis Services 配置”界面,在该界面中单击“添加当前用户”按钮,将当前用户添加为 SQL Server 管理员,如图 1-12 所示。

(11)单击“下一步”按钮,打开“Reporting Services 配置”界面,选中“安装和配置”单选按钮,如图 1-13 所示。

(12)单击“下一步”按钮,打开“Distributed Replay 控制器”界面,指定向其授予针对分布式重播控制器服务的管理权限的用户。具有管理权限的用户将可以不受限制地访问分布式重播控制器服务。单击“添加当前用户”按钮,将当前用户添加为具有上述权限的用户,如图 1-14 所示。



图 1-11 “数据库引擎配置”界面



图 1-12 “Analysis Services 配置”界面



图 1-13 “Reporting Services 配置”界面

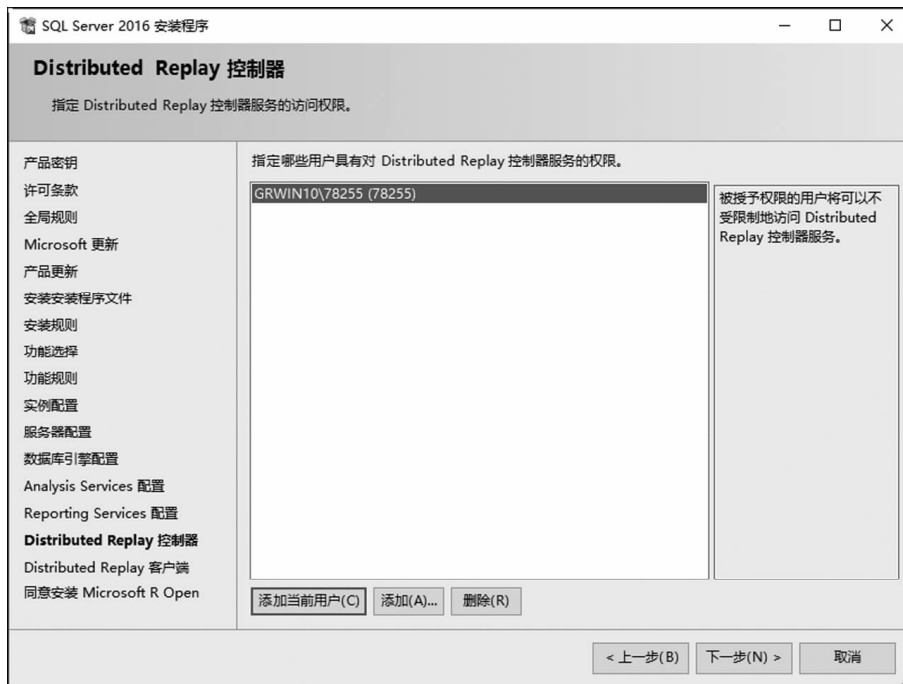


图 1-14 “Distributed Replay 控制器”界面

(13)单击“下一步”按钮,打开“Distributed Replay 客户端”界面,在“控制器名称”文本

框中输入名字,然后设置工作目录和结果目录,如图 1-15 所示。

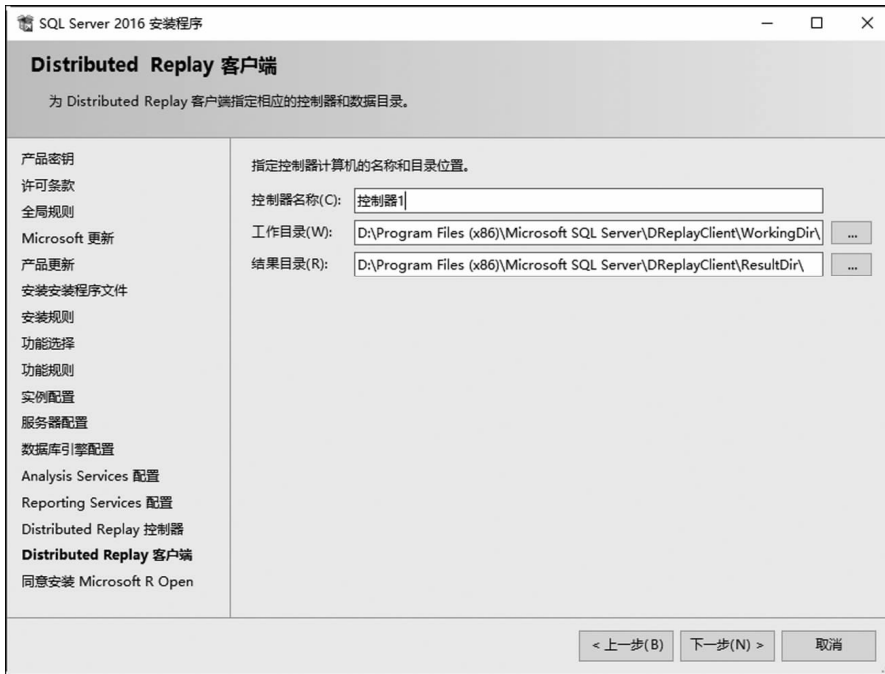


图 1-15 “Distributed Replay 客户端”界面

(14)单击“下一步”按钮,打开“同意安装 Microsoft R Open”界面,如图 1-16 所示,单击“接受”按钮。

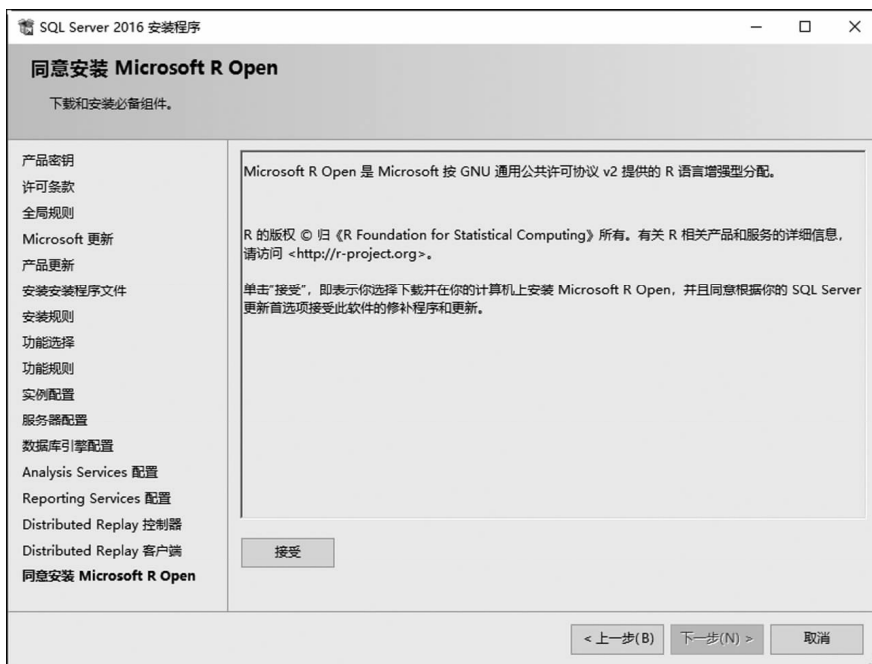


图 1-16 “同意安装 Microsoft R Open”界面

(15)单击“下一步”按钮,打开“准备安装”界面,如图 1-17 所示。

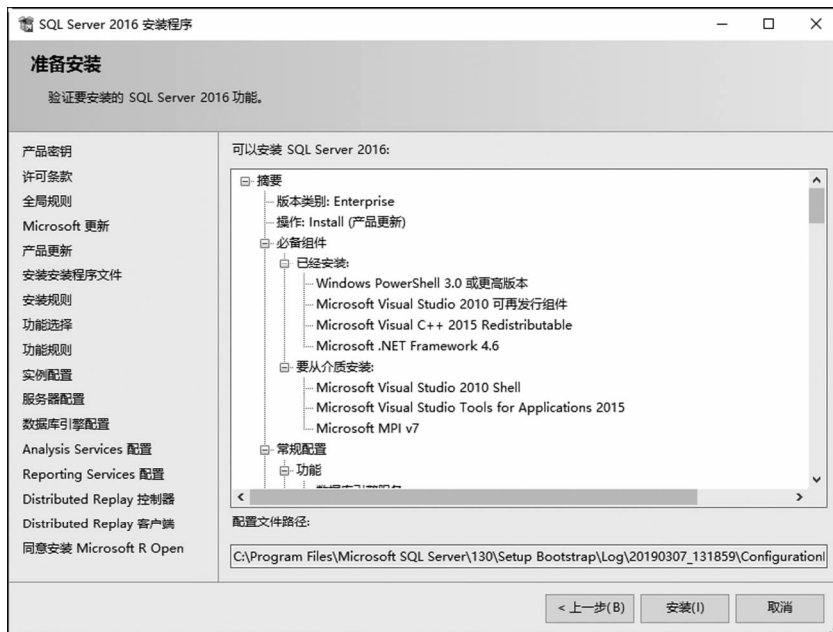


图 1-17 “准备安装”界面

(16)单击“安装”按钮开始安装,图 1-18 所示为提示安装完成的界面,单击“关闭”按钮即可。

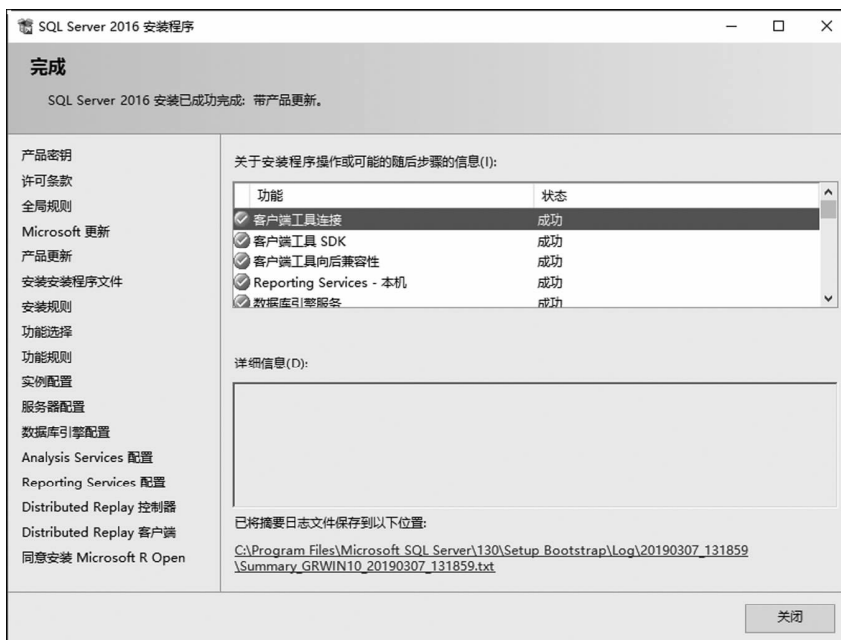


图 1-18 安装完成界面

1.2.4 安装 SQL Server Management Studio

SQL Server Management Studio 简易直观,可以使用该工具访问、配置、管理和开发 SQL Server 的所有组件。默认情况下,SQL Server Management Studio 没有被安装,下面介绍其安装步骤。

(1)运行 SQL Server 安装中心,选择左侧导航区域中的“安装”选项,然后选择“安装 SQL Server 管理工具”选项。

(2)在打开的网页中单击“下载 SQL Server Management Studio 17.9.1”超链接,如图 1-19 所示。



图 1-19 SQL Server Management Studio 下载页面

(3)下载完成后,双击 SSMS-Setup-CHS.exe,打开安装界面,如图 1-20 所示。



图 1-20 安装界面

(4)单击“安装”按钮,系统开始自动安装并显示安装进度,如图 1-21 所示。



图 1-21 正在安装

(5)安装完成后,单击“关闭”按钮即可,如图 1-22 所示。



图 1-22 安装完成

1.3 数据库的操作

1.3.1 数据库文件的构成

数据库与其他应用程序一样,需要存储在计算机中,其依然是以文件的形式被存储的。

例如,新建一个 Word 文档,计算机中就存储一个扩展名为“. docx”的文件。数据库与此类似,只是其存储在计算机中的文件不止一个。

数据库文件由数据文件和事务日志文件构成,如图 1-23 所示。其中,数据文件用于存放数据信息,分为主数据文件(扩展名为“. MDF”)和次要数据文件(扩展名为“. NDF”),主数据文件在数据库中只有一个,次要数据文件可有可无。事务日志文件用来存储对数据库的操作记录信息,在数据库中至少有一个,其扩展名为“. LDF”。

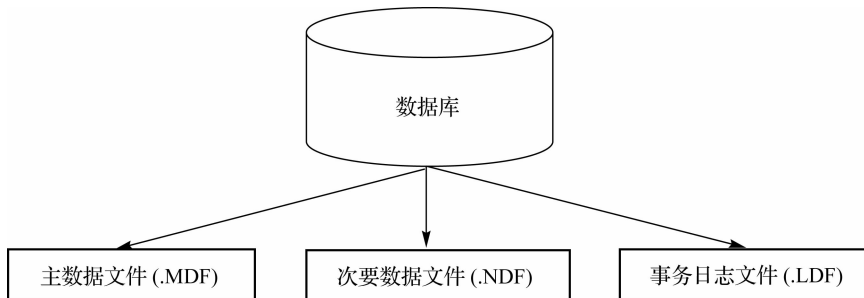


图 1-23 数据库文件

注意: 在默认状态下,数据库文件被存放在\MSSQL\DATA\目录下,主数据文件名为“数据库名_DATA.MDF”,事务日志文件名为“数据库名_LOG.LDF”。用户可以在创建数据库时指定其他路径和文件名,也可以添加次要数据文件和更多的事务日志文件。

1.3.2 数据库类型

在 SQL Server 中数据库分为系统数据库、用户数据库和示例数据库三种。

1. 系统数据库

系统数据库是在安装完数据库程序后系统自动生成的数据库,有 MASTER 数据库、MSDB 数据库、MODEL 数据库和 TEMPDB 数据库。

(1) MASTER 数据库。MASTER 数据库记录了所有系统信息,包括所有的其他数据库、登录账号和系统配置信息,以及所有用户定义数据库的存储位置和初始化信息。MASTER 数据库是 SQL Server 中最主要的系统数据库。

(2) MSDB 数据库。SQL Server、SQL Server Management Studio 和 SQL Server Agent 使用 MSDB 数据库来存储计划信息,以及与备份和恢复相关的信息。

(3) MODEL 数据库。MODEL 数据库为样板数据库,为用户数据库提供样板。它包含了将要复制到每个新建数据库中的系统表。

(4) TEMPDB 数据库。TEMPDB 数据库也是从 MODEL 数据库中复制而来的,它存储了 SQL Server 实例运行期间 SQL Server 需要的所有临时数据。

2. 用户数据库

用户数据库是用户根据实际需求,为处理某一方面事务而自行创建的数据库,用户在使用数据库时最常用的方式是自行创建数据库。

3. 示例数据库

示例数据库是 SQL Server 自带的,是用于例子、演示和说明的数据库。

1.3.3 创建数据库

创建数据库可以使用 SQL Server Management Studio 窗口方式和 T-SQL(transact structured query language)代码方式,使用窗口方式创建数据库比较直观、简单易用。

启动 SQL Server Management Studio,弹出“连接到服务器”对话框,在“身份验证”下拉列表框中选择“Windows 身份验证”选项,如图 1-24 所示。



图 1-24 “连接到服务器”对话框

1. 使用窗口方式创建数据库

进入数据库系统后,展开左侧窗格,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(见图 1-25),开始创建数据库。

在打开的“新建数据库”窗口中输入数据库的名称,这时数据库文件中的逻辑名称将自动生成,也可以自行命名。此时数据库文件为主数据文件和事务日志文件,可以分别对其逻辑名称、大小、最大大小、自动增长、路径等进行修改,如图 1-26 所示。在已经创建好的数据库中不能对路径进行修改。

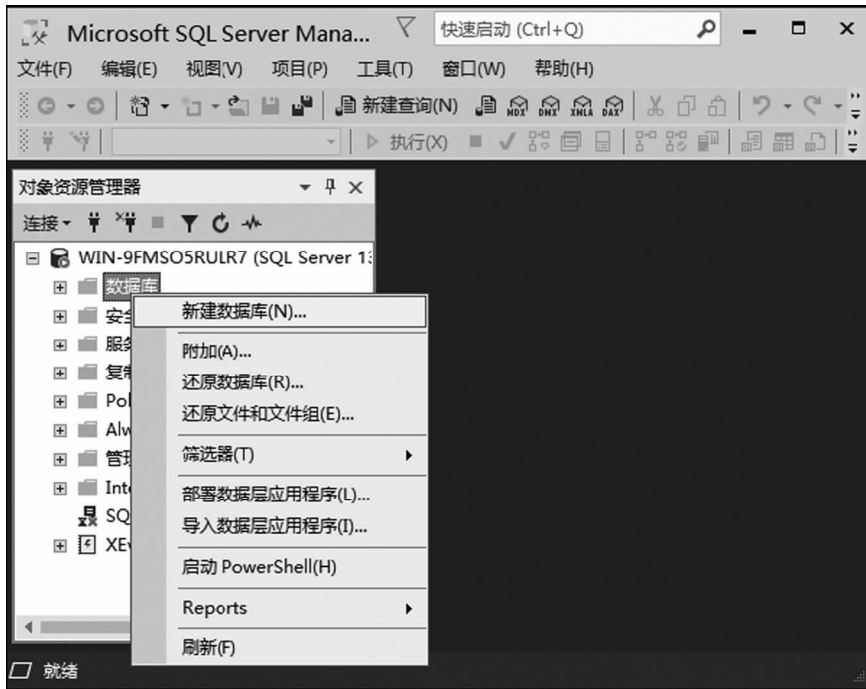


图 1-25 选择“新建数据库”命令



图 1-26 “新建数据库”窗口

单击“自动增长/最大大小”右边的按钮,打开“更改 IG 的自动增长设置”对话框,对文件增长方式和最大文件大小进行设置,如图 1-27 所示。

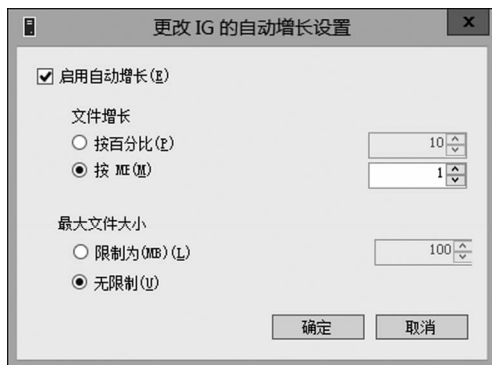


图 1-27 自动增长设置

在图 1-26 中拖动窗口下方的滚动条,单击右侧按钮可以选择文件的存放位置,如图 1-28 所示。



图 1-28 选择文件的存放位置

在图 1-26 中,单击“添加”按钮可以添加次要数据文件和事务日志文件,在“文件类型”中选择“行数据”添加次要数据文件,选择“日志”添加事务日志文件,如图 1-29 所示。设置好后单击“确定”按钮,完成数据库创建,此时在左侧窗格中出现已创建好的数据库“IG”。对已创建好的数据库,可以右击“数据库”并选择“属性”选项,进行数据文件的修改。



图 1-29 添加文件

2. 使用代码方式创建数据库

语法格式如下。

```
CREATE DATABASE 数据库名
ON [PRIMARY]
(NAME=逻辑数据名,
FILENAME='路径\文件名.MDF'
[,SIZE=初始大小]
[,MAXSIZE=最大值]
[,FILEGROWTH=增量]
)
[
(NAME=逻辑数据名,
FILENAME='路径\文件名.NDF'
[,SIZE=初始大小]
```



```

    [,MAXSIZE=最大值]
    [,FILEGROWTH=增量]
]
[,...N]
LOG ON
    (NAME=逻辑日志名,
    FILENAME='路径\文件名.LDF'
    [,SIZE=初始大小]
    [,MAXSIZE=最大值]
    [,FILEGROWTH=增量]
)
[,...N]

```

说明：在语法格式中，中括号表示可选项，若需要中括号中的内容，则去除中括号再写括号中的内容；小括号表示必选项，即必须要写的部分，带上小括号一起写在代码中；[, ... N] 表示还可以创建同样的文件。其中，主数据文件初始大小至少为 3 MB。

【例 1.1】 创建数据库，其中数据库名为 XG1，主数据文件名为 PXG1，存放位置为 D 盘根目录，初始大小为 3 MB，最大值为 100 MB，增长为 10 MB，事物日志文件名为 LXG1，其他设置与主数据文件相同。

```

CREATE DATABASE XG1
ON
    (NAME=PXG1,
    FILENAME="D:\PXG1.MDF",
    SIZE=3,
    MAXSIZE=100,
    FILEGROWTH=10
    )
LOG ON
    (NAME=LXG1,
    FILENAME="D:\LXG1.LDF",
    SIZE=3,
    MAXSIZE=100,
    FILEGROWTH=10
    )

```

1.3.4 修改数据库

用户可以对已经创建好的数据库进行修改,操作方法是:在“对象资源管理器”窗格中选择已创建的数据库,右击并选择“属性”选项,打开“数据库属性-IG”窗口,选择“文件”选项,如图 1-30 所示。在“数据库文件”选项区中可以对数据库文件的逻辑名称、大小、自动增长/最大大小等进行修改,但“路径”不能修改,因此在创建数据库时应注意路径的选择。

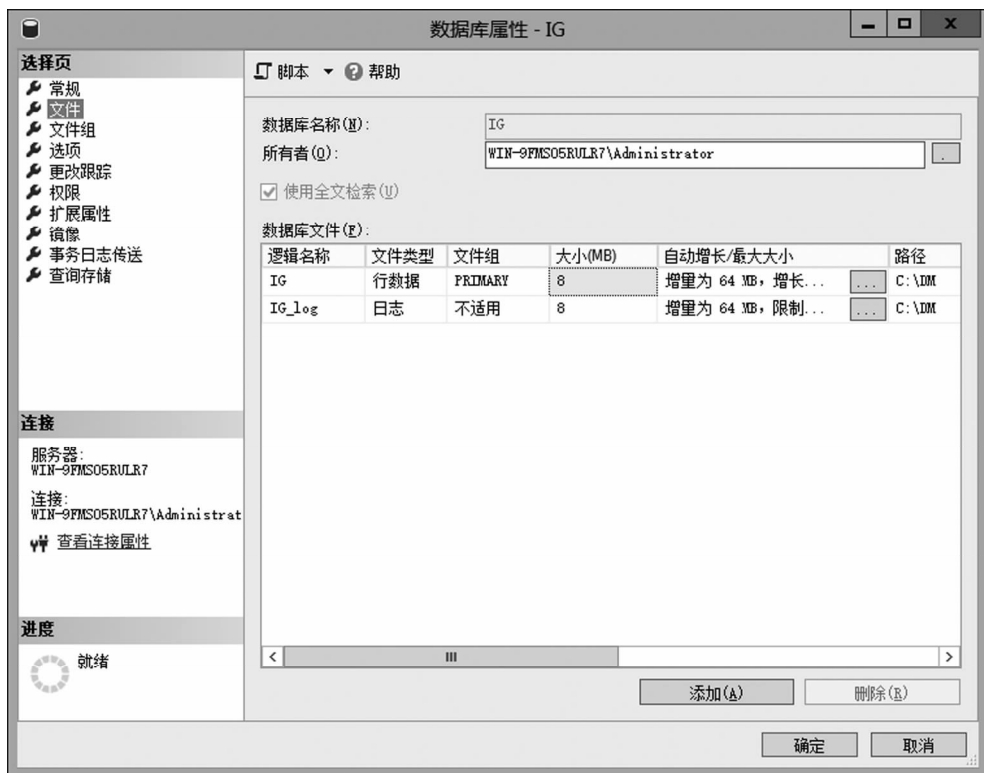


图 1-30 “数据库属性-IG”窗口

1.3.5 打开数据库

在使用数据库进行操作时,可以使用窗口方式和命令代码方式打开数据库,特别是在代码操作时,要对数据库的对象进行操作。例如,查询表中的数据,如果数据库处于关闭状态,就无法完成相应的操作。

1. 使用窗口方式打开数据库

启动 SQL Server Management Studio,在“对象资源管理器”窗格中展开“数据库”节点,单击要打开的数据库。例如,打开 XG1 数据库,单击 XG1 数据库即可。

2. 使用命令方式打开数据库

语法格式如下。

```
USE 数据库名
```

1.3.6 数据库的更名与删除

1. 数据库的更名

数据库创建完成后,可以根据实际需要更改数据库名,但是当数据库中有被打开的对象(如数据表处于打开状态)时,数据库名是不能被更改的。因此,应首先将数据库对象关闭,再进行修改操作。

【例 1.2】 使用 SQL Server Management Studio 窗口方式将“学生”数据库的名字改为“学生数据库”。

在“对象资源管理器”窗格中,展开数据库,选中要更名的数据库,这里选中“学生”数据库,右击并选择“重命名”命令,将数据库名称改成新的名字“学生数据库”即可,如图 1-31 所示。

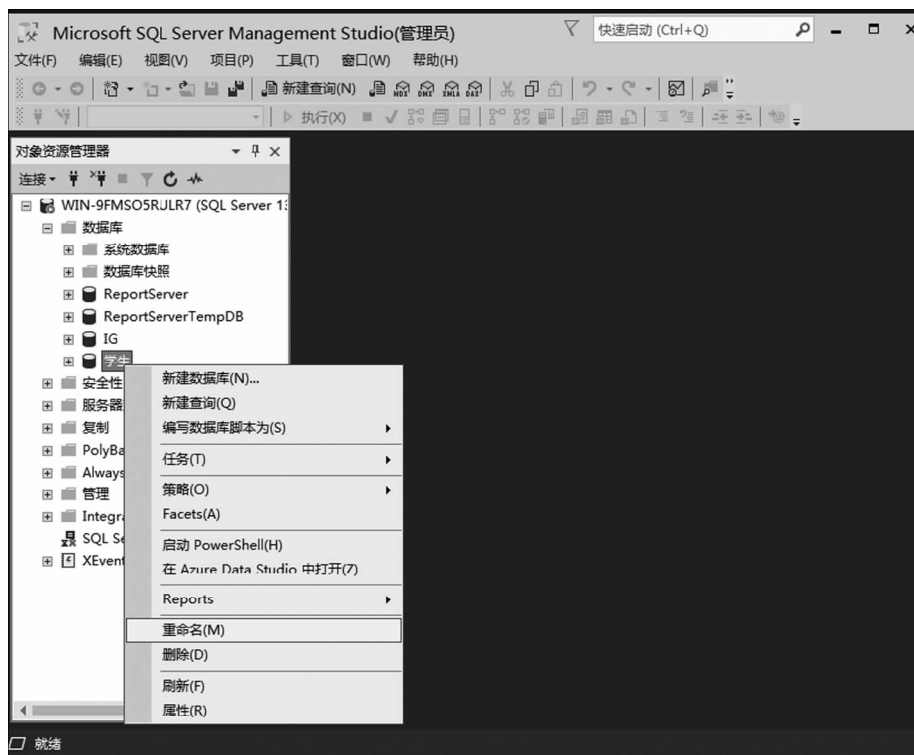


图 1-31 重命名数据库

2. 数据库的删除

可以将不需要的数据库删除,删除的数据库不仅在“对象资源管理器”中没有了,而且数据库文件也一同被删除了,因此,在删除数据库前要谨慎操作。其操作步骤为:右击数据库并选择“删除”命令,打开“删除对象”窗口,选中“删除数据库备份和还原历史记录信息”及“关闭现有连接”复选框,然后单击“确定”按钮。

1.3.7 数据库的分离与附加

分离数据库可以将数据库中暂时不用的数据文件和日志文件分离出来,从而减轻 SQL Server 服务器的负担。分离数据库虽然是将数据库从 SQL Server 数据库引擎实例中去除,但保留完整的数据库及其数据文件和事务日志文件,要区别于删除数据库,后者是将数据库文件及其数据都从服务器上的磁盘中彻底删除。

数据库的分离与附加也是数据库备份的一种常用方法,当数据库有了更新后,需要对数据库进行备份操作,可以使用数据库分离来完成。数据库的分离实际上是使用 SQL Server 软件与数据库文件分离,分离后在 SQL Server 软件“对象资源管理器”窗格的数据库中不再显示被分离的数据库,这时可以将此数据库的文件(主数据文件、次要数据文件和事物日志文件)复制到其他存储位置或 U 盘等介质中,以便下次使用,在下次使用时可以通过附加数据库完成数据库的还原操作,这时将会在 SQL Server 软件“对象资源管理器”窗格的数据库中显示被附加的数据库。

1. 分离数据库

(1)在“对象资源管理器”窗格中选择要分离的数据库“学生”,右击,在弹出的快捷菜单中选择“任务”→“分离”命令,如图 1-32 所示。

(2)在打开的“分离数据库”窗口中可见要分离的数据库名称,选中“删除连接”“更新统计信息”选项。然后单击“确定”按钮,完成分离操作,如图 1-33 所示。此时再回到“对象资源管理器”窗格中,可以发现“学生”数据库已不在数据库中。

2. 附加数据库

(1)在“对象资源管理器”窗格中右击“数据库”,在弹出的快捷菜单中选择“附加”命令,如图 1-34 所示,打开“附加数据库”窗口。

(2)在“附加数据库”窗口的“要附加数据库”中单击“添加”按钮,弹出“定位数据库文件”对话框,如图 1-35 所示。

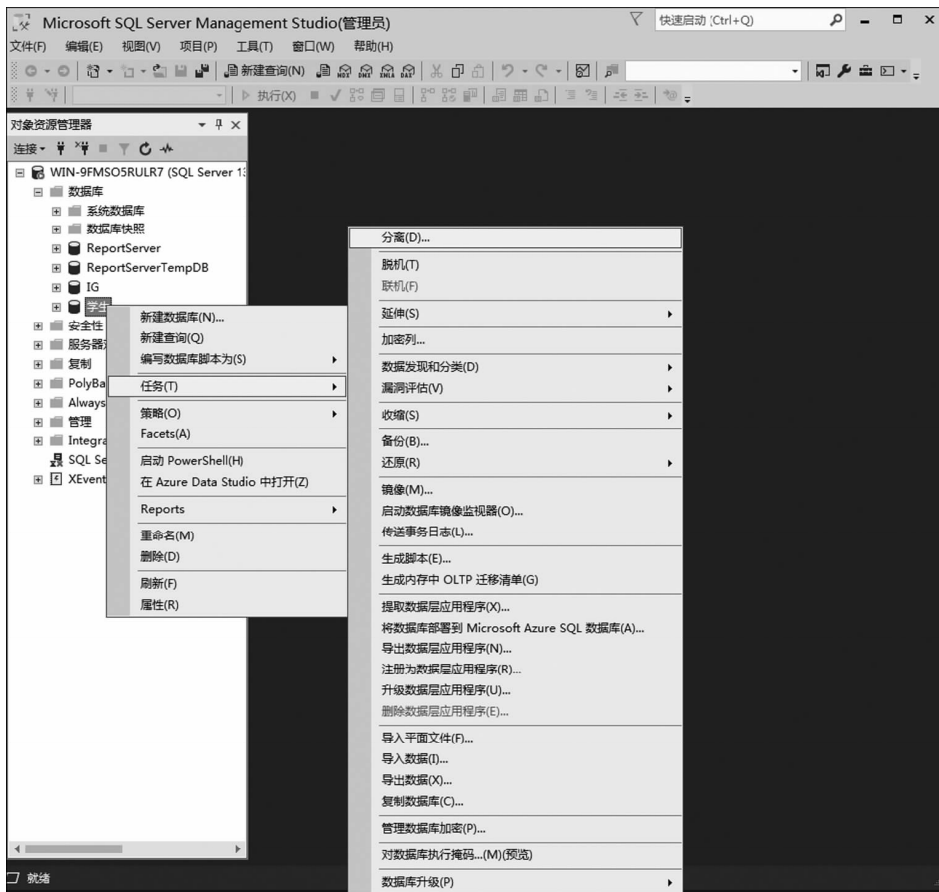


图 1-32 选择分离数据库

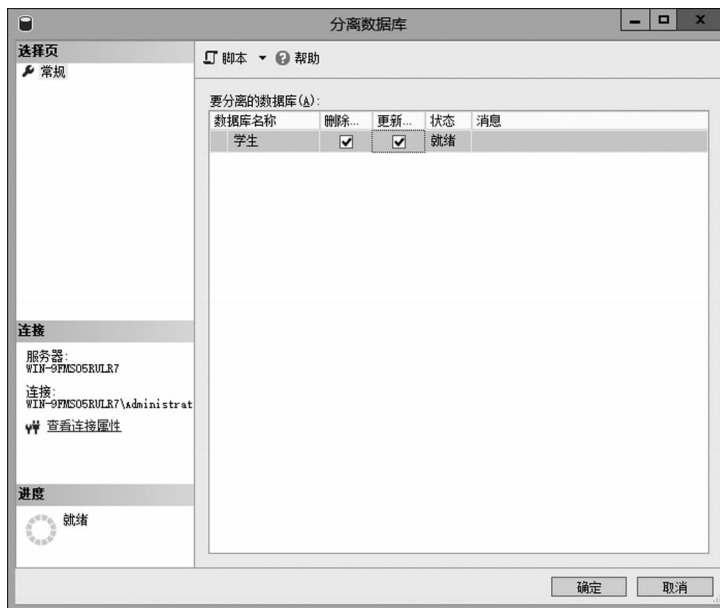


图 1-33 分离数据库

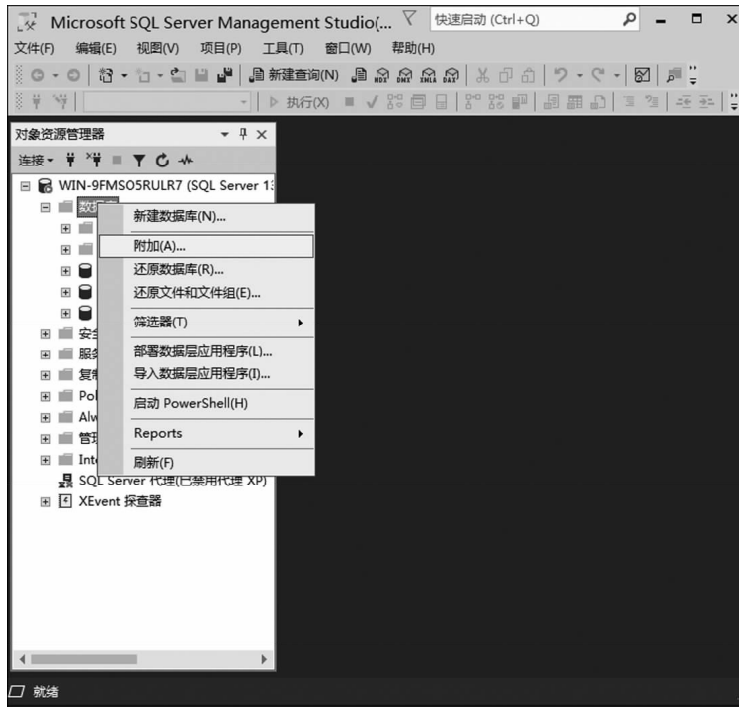


图 1-34 选择“附加”命令

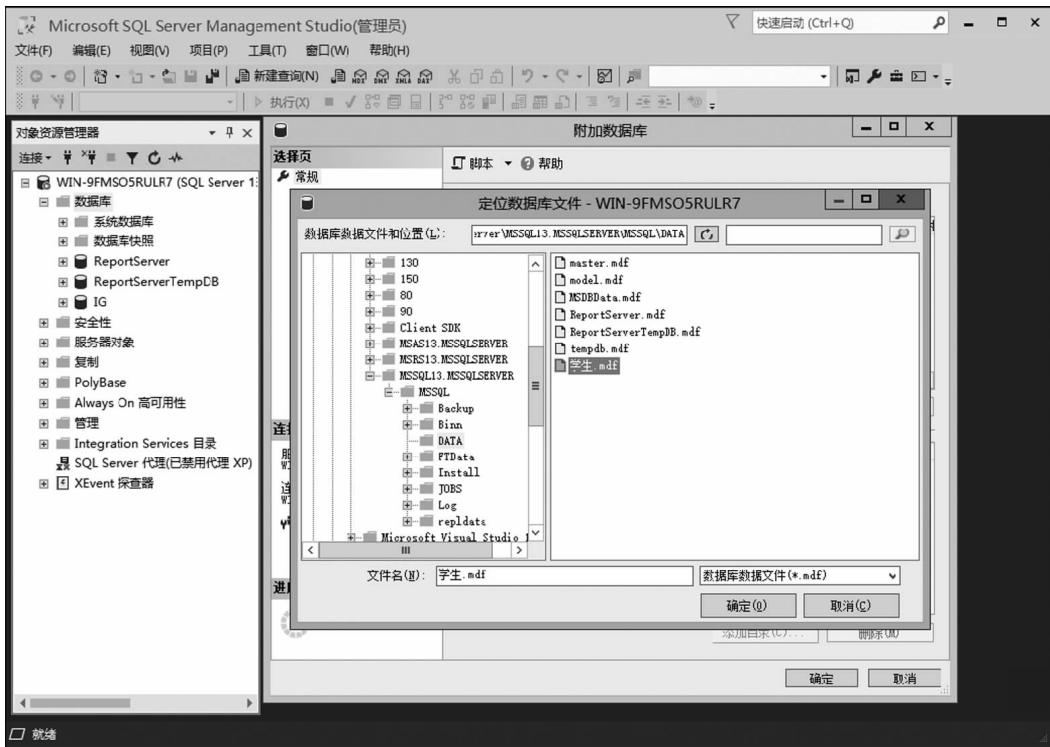


图 1-35 “定位数据库文件”对话框

(3)在“定位数据库文件”对话框中选择要附加的数据库,单击“确定”按钮回到“附加数据库”窗口,此时要附加的数据库文件信息显示在窗口中,如图 1-36 所示。

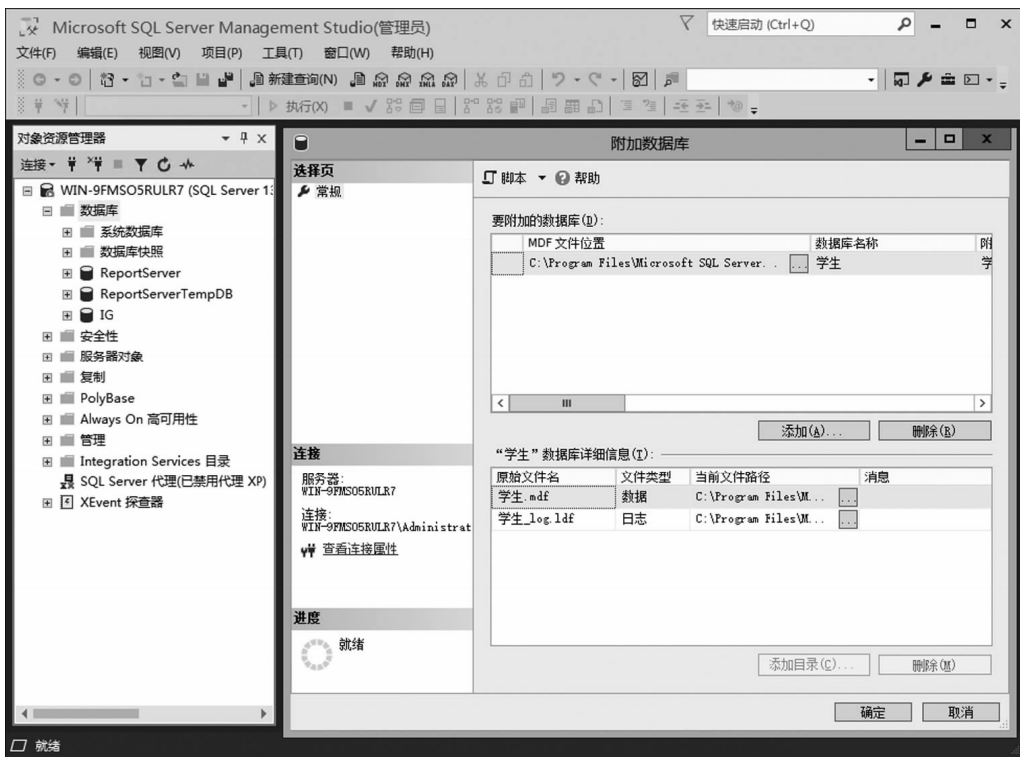


图 1-36 附加数据库

(4)在“附加数据库”窗口中单击“确定”按钮完成数据库的附加操作。此时,在“对象资源管理器”窗格中可见到已还原的数据库。

1.3.8 数据库的备份与还原

备份数据库是为了数据的安全使用,防止数据因意外情况而丢失,特别是在完成了大量数据的操作后,计算机出现故障造成数据库无法使用,如果事先没有对数据库进行备份,那么将造成重大损失。

而在某些环境中,如学校的机房,机房中的计算机中都装有还原卡,每次重启计算机,计算机系统都将还原,而对于操作的数据库也不例外。每次上机操作,都要重新新建数据库,重新录入数据,这就显得相当麻烦。为避免这种情况的出现,就需要对数据库进行备份,在下次上机使用时,通过还原操作即可恢复原有数据。

如果数据库受到损坏导致数据不可读,则用户可以从备份的文件中进行数据库的重建,从而恢复数据库。

此外,数据库的备份还可以用作其他用途。例如,要在不同服务器之间复制数据,只要

将某个服务器上的数据备份后恢复到另一台服务器上,就可以又快又方便地完成数据库的复制。

1. 数据库的备份操作

(1)在 C 盘下建立一个目录 BF,作为存储备份文件的文件夹。在“对象资源管理器”窗格中展开数据库,选中要备份的“学生”数据库,右击,在弹出的快捷菜单中选择“任务”→“备份”命令,如图 1-37 所示。

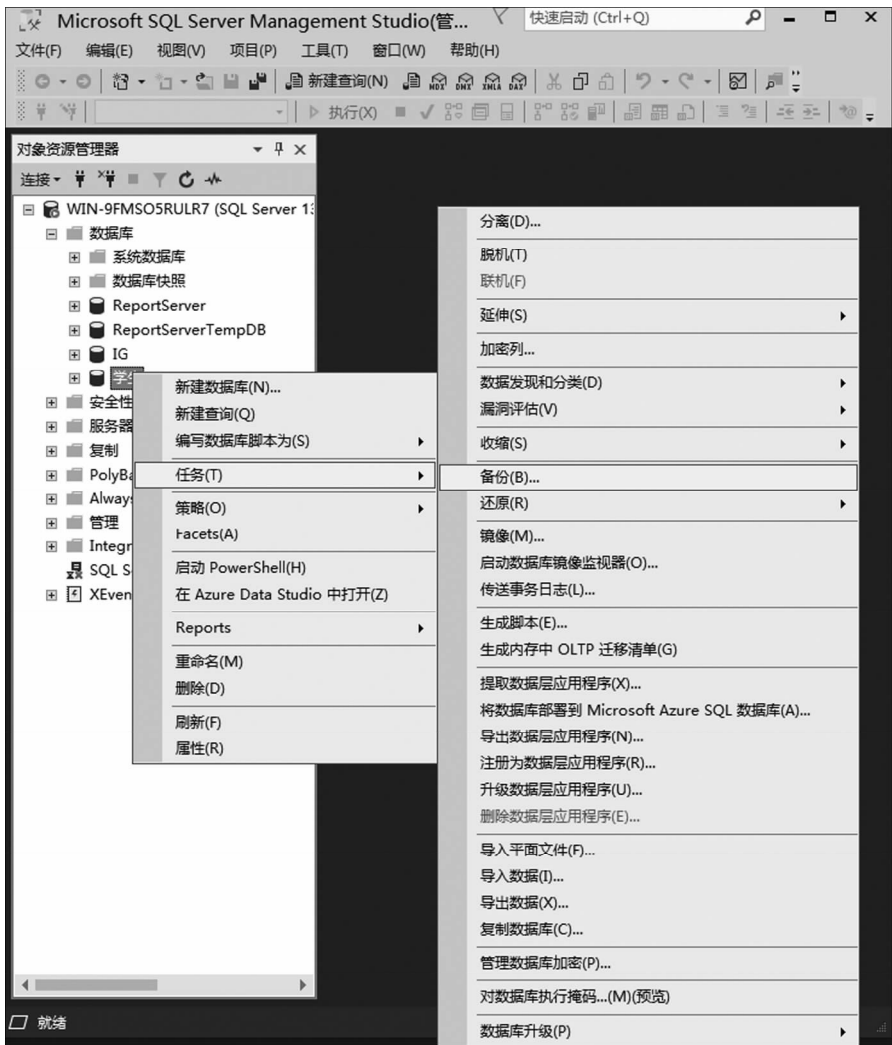


图 1-37 选择备份数据库

(2)打开“备份数据库-学生”窗口,在“源”选项组中的“数据库”下拉列表框中选择要备份的数据库;在“备份类型”下拉列表框中选择“完整”选项,选中“数据库”单选按钮,如图 1-38 所示。

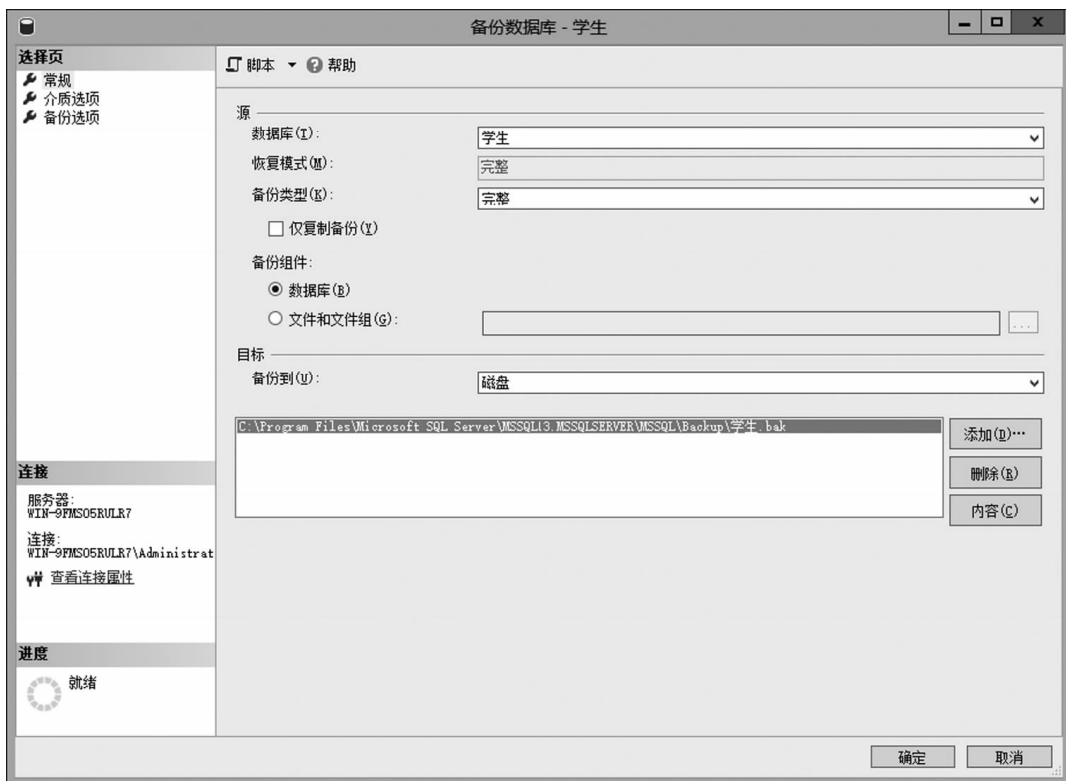


图 1-38 “备份数据库-学生”窗口

(3)在“目标”选项区中的“备份到”下拉列表框中选择“磁盘”选项,在下面的文本框中显示备份的文件位置,可以单击右边的“删除”按钮删除,也可以通过单击“添加”按钮选择备份的文件位置。如果要指定备份的文件位置,则单击“添加”按钮,这时将弹出“选择备份目标”对话框,在“磁盘上的目标”选项区中选中“文件名”单选按钮,然后单击右侧的 [...] 按钮,如图 1-39 所示,此时弹出“定位数据库文件”窗口,如图 1-40 所示。



图 1-39 选择备份目标 1

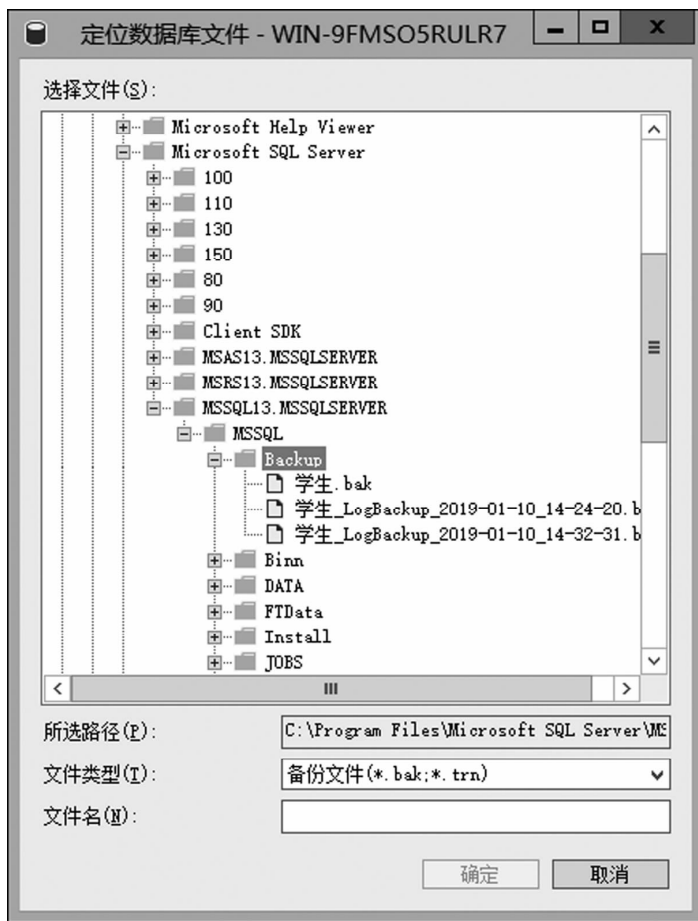


图 1-40 “定位数据库文件”窗口 1

(4)在“定位数据库文件”窗口中选择备份路径,这里选择 C 盘根目录,再选择目录“BF”,此时窗口中的所选路径变为“C:\BF”,在“文件名”文本框中输入“学生 6_6. bak”,如图 1-41 所示。

(5)单击“确定”按钮返回“选择备份目标”对话框,单击“确定”按钮返回“备份数据库-学生”窗口,完成数据库的备份操作,此时弹出数据库备份完成提示框,单击“确定”按钮完成备份操作,如图 1-42 所示。此时,在 C 盘 BF 目录下可以看到备份的数据库文件“学生 6_6. bak”,可以将备份的文件复制到 U 盘或其他存储位置以备用。

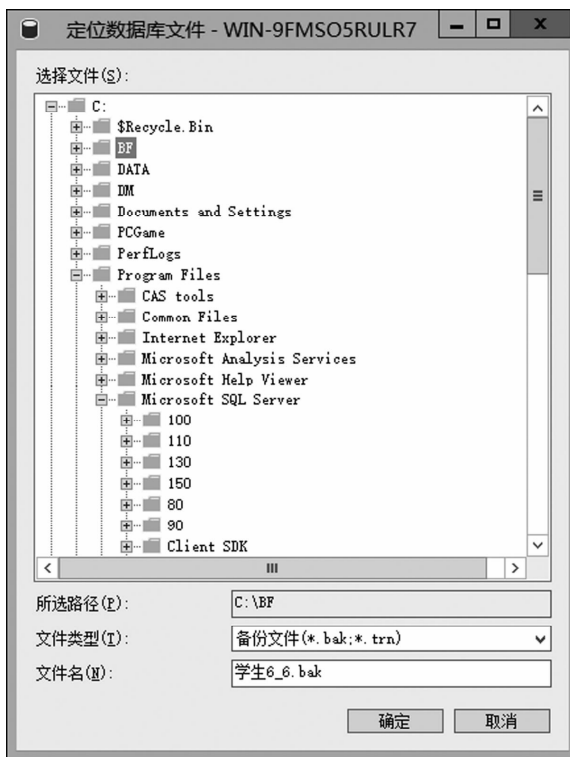


图 1-41 “定位数据库文件”窗口 2

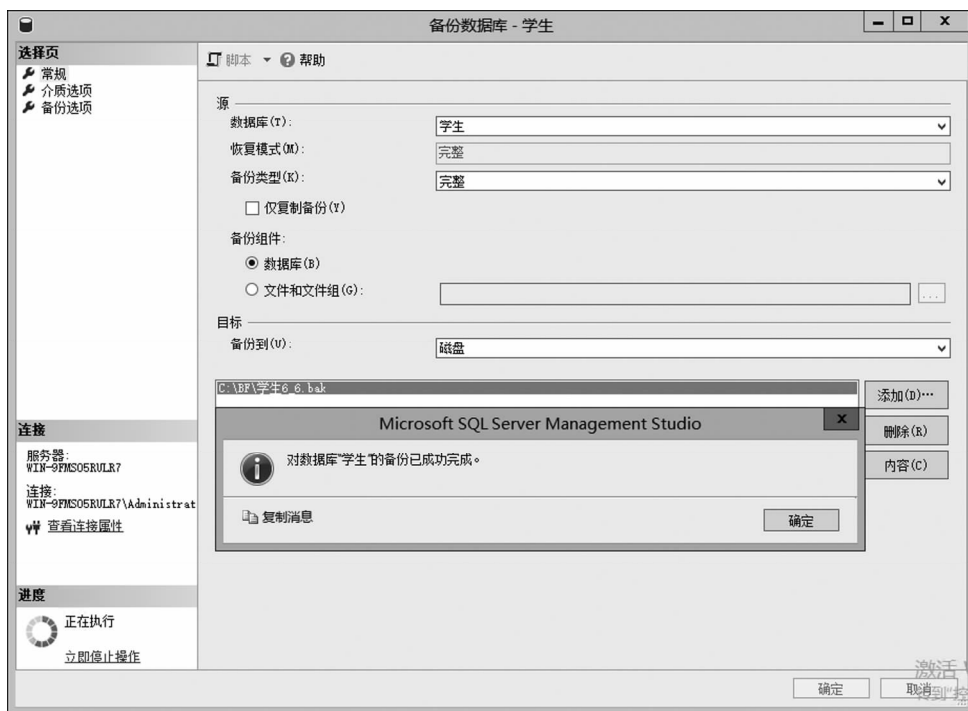


图 1-42 备份完成

2. 数据库的还原操作

当从备份上重载一个数据库时,需要检查备份文件或备份设备的有效性,还要确保此数据库不处于使用状态,限制其他用户对数据库的访问,指定数据库中的任何数据都由重载的数据替换;否则,无法还原数据库。

(1)如果要还原的数据库已存在,则先将其分离,然后在“对象资源管理器”窗格中右击“数据库”并选择“还原数据库”命令,如图 1-43 所示。

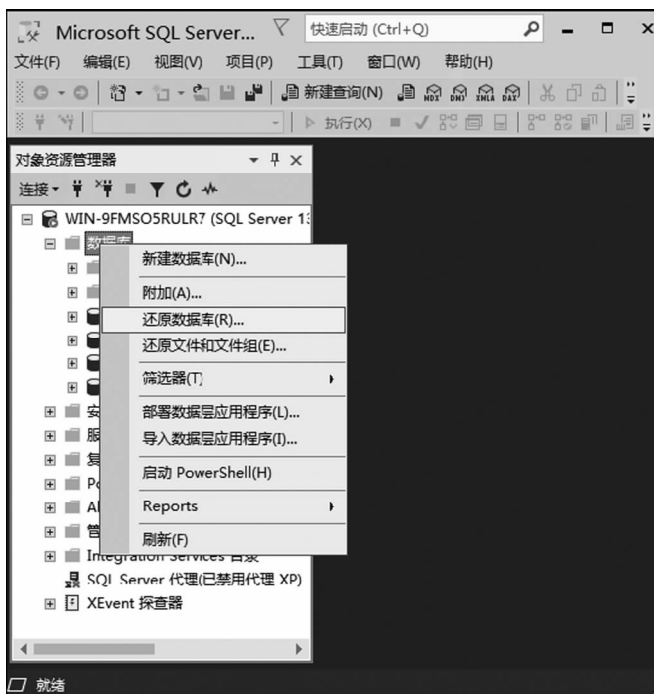



图 1-43 选择“还原数据库”命令

(2)在“还原数据库-学生数据库”窗口的“目标”选项区的“数据库”文本框中输入“学生数据库”;在“源”选项区中选中“设备”单选按钮,如图 1-44 所示。

(3)单击“设备”右边的  按钮,弹出“选择备份设备”窗口,单击“添加”按钮,在弹出的“选择备份设备”窗口中选择备份文件的位置及备份文件,如图 1-45 所示。

(4)选择好备份文件后,单击“确定”按钮返回到“指定备份”窗口,然后单击“确定”按钮返回到“还原数据库”窗口,这时在“要还原的备份集”列表框中显示要进行还原的数据库信息,在“还原”列下勾选,最后单击“确定”按钮,完成数据库的还原操作,如图 1-46 所示。

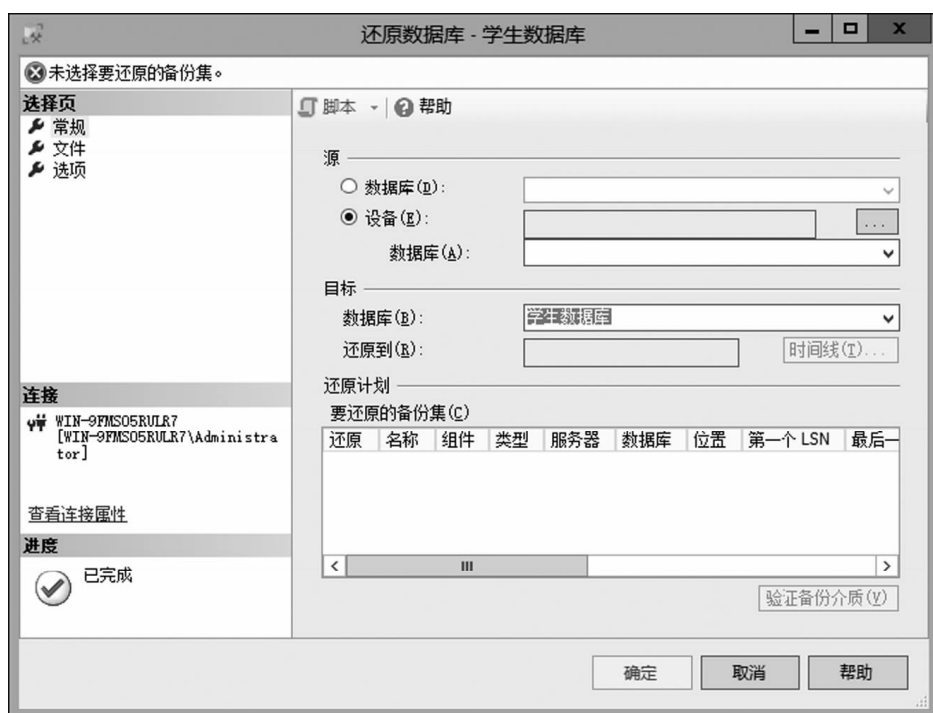


图 1-44 “还原数据库-学生数据库”窗口



图 1-45 指定备份

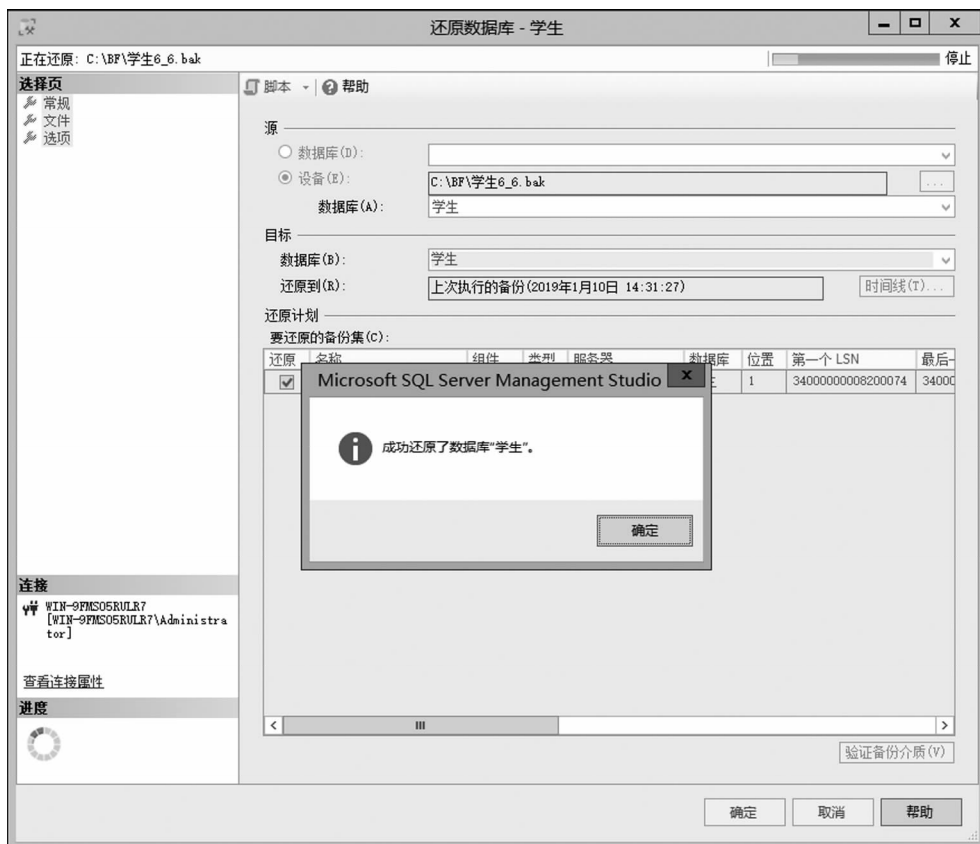


图 1-46 完成数据库的还原

1.3.9 数据库的导入与导出

数据库的导入与导出可以实现在不同的数据源和目标之间的数据复制与转换。利用 SQL Server 的导入与导出向导可以在 SQL Server 之间,在 SQL Server 与 ODBC 数据源、OLE DB、文本文件之间进行数据的导入与导出操作。下面以将 Excel 工作簿中的工作表导入 SQL Server 数据库中为例进行讲解。

1. 数据库的导入

(1) 启动 SQL Server Management Studio, 展开“对象资源管理器”窗格中的数据库, 选择“学生”数据库, 右击并选择“任务”→“导入数据”命令, 如图 1-47 所示。



图 1-47 选择“导入数据”命令

(2) 打开“SQL Server 导入和导出向导”窗口,单击 Next 按钮,进入“选择数据源”界面。在“数据源”下拉列表框中选择“Microsoft Excel”,在“Excel 连接设置”选项区中选择要导入数据的 Excel 工作簿的位置及版本,如图 1-48 所示。



图 1-48 选择数据源

(3)单击 Next 按钮进入“选择目标”界面,在“数据库”下拉列表框中选择要将数据导入哪个数据库中,这里采用默认值“学生”数据库,如图 1-49 所示。

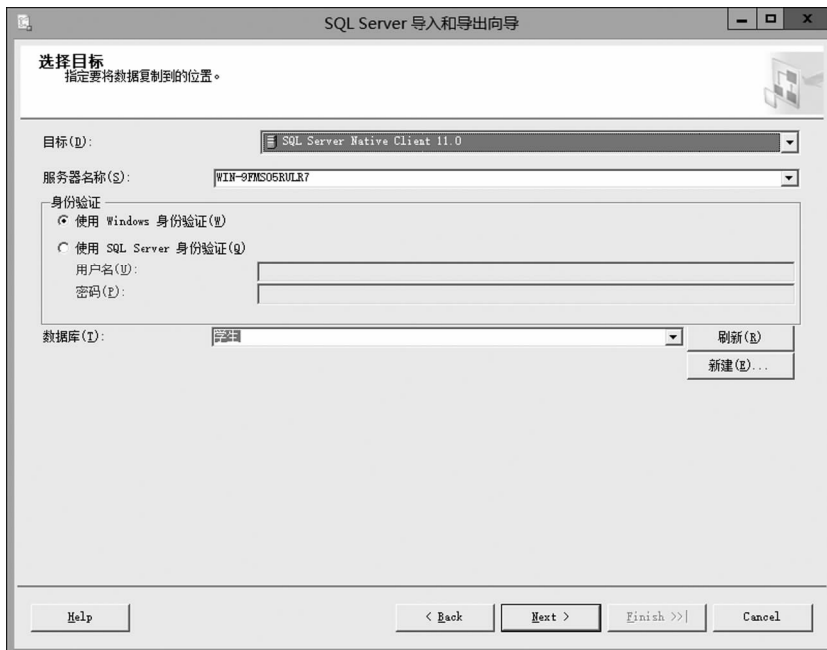


图 1-49 选择目标数据库

(4)单击 Next 按钮进入“指定表复制或查询”界面,采用默认设置。单击 Next 按钮进入“选择源表和源视图”界面,选择要导入的工作表,如图 1-50 所示。



图 1-50 “选择源表和源视图”界面

(5)单击 Next 按钮进入“保存并运行包”界面,选择“立即运行”,然后单击 Next 按钮进入“完成该向导”界面,单击“完成”按钮,完成数据导入。

2. 数据库的导出

(1)选择要导出的数据库,右击并选择“任务”→“导出数据”命令,打开导出向导。在数据源位置选择要导出的数据库,这里采用默认值。进入“选择目标”界面,在“目标”下拉列表框中选择“Microsoft Excel”,在“Excel 文件路径”中选择要导出的位置及给导出的文件命名(根据使用的计算机情况选择合适的位置进行导出),然后选择版本,如图 1-51 所示。

(2)进入下一步,在“指定表复制或查询”界面中采用默认值;进入下一步,在“选择源表和源视图”界面的“表和视图”选项区中选择要导出的表“成绩表”,如图 1-52 所示。

(3)进入下一步,在“查看数据类型映射”界面中单击 Next 按钮,立即运行,完成导出操作。这时可以在导出文件的位置(这里是 C 盘根目标)下查看已被导出的工作表。

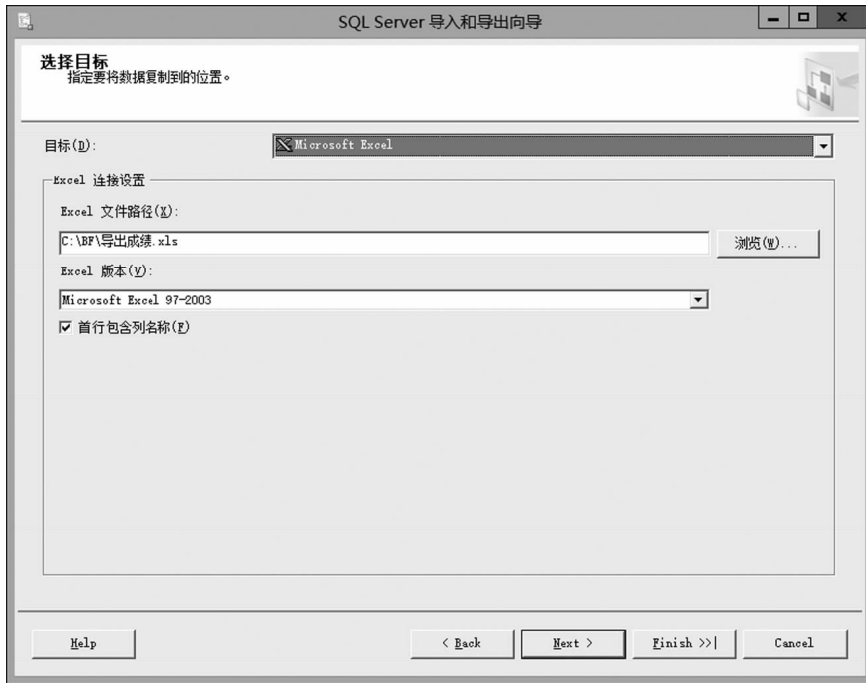


图 1-51 指定导出数据库



图 1-52 选择导出的表

1.4 小 结

本模块首先介绍了数据库的相关概念,详细介绍了数据库、数据库管理系统、数据库应用系统的关系;然后对数据库文件的类型进行了详细的分析,着重介绍了必要的主数据文件和事务日志文件;最后对数据库的创建进行了详细的介绍,分别利用 SQL Server Management Studio 和 SQL 代码的方式进行阐述。通过本模块的学习,学生应该对数据库的用途、作用和创建方法有了总体的认识。

1.5 习 题

1. 简述数据库的作用及数据库系统的构成。
2. 简述数据库文件的构成。
3. 简述关系表中字段、记录和主键的概念。
4. 使用 SQL Server Management Studio 窗口方式创建名为 XS1 的数据库,将其存放在 D 盘 DATA 文件夹中。其初始大小为 3 MB,最大值为 20 MB,增量为 5 MB。
5. 使用代码方式创建名为 XS2 的数据库,将其存放在 D 盘根目录下,主数据文件名为 XS2M,事务日志文件名为 XS2L,初始大小为 5 MB,最大值为 20 MB,增量为 5 MB。
6. 将数据库 XS1 备份到 D 盘 DATA 文件夹中。
7. 对数据库 XS1 进行还原操作。
8. 对数据库 XS2 进行分离操作,然后将数据文件复制到 U 盘。
9. 将 XS2 的数据文件从 U 盘复制到 D 盘根目录,然后通过数据库的附加操作将 XS2 数据库还原。

2

模块 2

表和表中数据的操作

▶ 学习要点

- SQL Server 的数据类型。
- 表的创建、修改和删除操作。
- 表中数据的添加、修改和删除操作。

2.1 表的构成

数据库设计、创建完成后,接下来的工作就是创建表。表是数据库中最主要的数据对象,许多操作都是基于表进行的。表是用于存储和操作数据的一种逻辑结构,是一列列的集合。例如,学生信息被登记在学生表中,见表 2-1。

表 2-1 学生表

学号	姓名	专业	年龄	出生日期	身份证号	特长
2017123501	李明	计算机	19	1998.5.1	510111199805015830	唱歌
2017123502	张强	会计	18	1999.6.3	510112199906037636	
2017123503	李小爱	电商	17	2000.9.12	510110200009125837	舞蹈
2017123504	王琴	会计	18	1999.5.26	510113199905265871	国画

通过表 2-1,能看出表是由表名、表头和表中数据构成的。

(1)表名。表名是二维表的标识。每张二维表都有一个表名,而且在同一个数据库中表

名不能相同。

(2)表头。表头是二维表的字段,二维表的每一列对应数据表中的一个字段,其属性决定了字段名、字段类型和字段长度,见表 2-2。

表 2-2 类型

学 号	姓 名	专 业	年 龄	出生日期	身份证号	特 长
CHAR(10)	VARCHAR(10)	VARCHAR(6)	TINYINT	SMALLDATETIME	CHAR(18)	TEXT

(3)表中数据。表中数据是二维表中的记录,每一行称为一条记录,每一条记录表明一条信息。

2.2 SQL Server 的数据类型

在创建表之前,必须定义表中数据的类型。数据类型指定了某列能够保存的信息的类型及如何存储这些数据。例如,若某一列要用于保存姓名,则应定义为字符型;又如,某列要用于保存成绩,则应定义为数值型。Microsoft SQL Server 提供了多种系统数据类型,各数据类型及其符号标识如表 2-3 所示。另外,SQL Server 也允许基于系统数据类型的用户定义数据类型。

表 2-3 数据类型及符号标识

数据类型	符号标识
字符型	CHAR[(N)],VARCHAR[(N)],TEXT
UNICODE 字符型	NCHAR(N),NVARCHAR[(N)],NTEXT
整型	INT,BIGINT,SMALLINT,TINYINT
精确数值型	DECIMAL[(P[,S])],NUMERIC[(P[,S])]
浮点型	FLOAT[(N)],REAL
日期和时间型	DATETIME,SMALLDATETIME
货币型	MONEY,SMALLMONEY
二进制型	BINARY[(N)],VARBINARY[(N)],IMAGE
特殊类型	UNIQUEIDENTIFIER,BIT,TIMESTAMP,TABLE,SQL_VARIANT

在确定数据类型时,会使用到精度、小数位数和长度 3 个概念,其中前两个概念是针对数值型数据的。

(1)精度:数值型数据中存储的十进制数据的总位数。例如,123.56 的精度为 5。

(2)小数位数:数值型数据中小数点右边可以有的数字位数的最大值。例如,1.23 的小数位数为 2。

(3)长度:存储数据所占用的字节数。例如,TINYINT 类型数据占用的存储空间为 1 个字节。

2.2.1 系统数据类型

1. 字符型

字符型数据用于存放字符串,而所有用单引号引起的数据都称为字符型数据。例如,'张明' 'AB@C'。字符型分为定长字符型、可变长度字符型和 TEXT 类型。

(1)定长字符型 CHAR[(N)]:N 定义字符型数据的长度,N 为 1~8 000,默认为 1。在存储数据时,若实际要存储的串长度不足 N,则在串的尾部添加空格补足。例如,定义姓名列的数据类型为 CHAR(6),而实际输入的数据为'张明',则存储的数据为'张明 ',其中末尾添加了两个空格,以达到 6 个字节。

(2)可变长度字符型 VARCHAR[(N)]:N 的规定与定长型相同,但此类型的 N 为可表示的最大长度。VARCHAR[(N)]的长度为输入的字符串的实际字符个数,而不一定是 N。例如,对部门名称定义为 VARCHAR(20),而实际输入的字符串为'市场部',此时不在后面加空格,其长度为 6。这样大大节省了存储空间。

两者使用的区别:当列中数据长度比较接近时,使用 CHAR 类型;当列中数据长度明显不同时,使用 VARCHAR 类型。

(3)TEXT 类型:当需要存储大量字符串(特别是长度超过 8 000)时,需要使用 TEXT 类型。例如,备注、说明等列可以定义为此类型。TEXT 类型最大长度为 $2^{31}-1$ 个字符。

2. UNICODE 字符型

UNICODE 是统一字符编码标准,用于支持国际上非英语语种的字符数据的存储和处理。使用 UNICODE 数据类型,列可存储由 UNICODE 标准定义的任何字符,包含由不同字符集定义的所有字符。但是 UNICODE 数据类型需要相当于非 UNICODE 数据类型两倍的存储空间。因为它采用的是双字节编码方式存储每个字符,当要使用的字符不能全部存储在单字节字符数据中时(与存储英文不同),就要使用 UNICODE 字符型进行存储,即每个字符占用两个存储空间。因此,对于 NCHAR、NVARCHAR 最多能存储 4 000 个字符,占用 8 000 字节的空间。

UNICODE 数据使用 SQL Server 中的 NCHAR、NVARCHAR 和 NTEXT 数据类型进行存储。对于存储来源于多种字符集的字符的列,可采用这些数据类型。当列中各项所包含的 UNICODE 字符数不同(最多为 4 000)时,使用 NVARCHAR 类型。当列中各项为

同一固定长度(最多为 4 000 个 UNICODE 字符)时,使用 NCHAR 类型。当列中任意项超过 4 000 个 UNICODE 字符时,使用 NTEXT 类型。

UNICODE 字符型的用法与字符型相同,只不过 N 的范围为 1~4 000,则相应的存储空间为 2~8 000 字节,即长度为 2N 字节。

3. 整型

整型数据由负整数或正整数组成,如-15、0、5、2 509 等。在 SQL Server 2016 中,整型数据使用 BIGINT、INT、SMALLINT 和 TINYINT 数据类型存储。BIGINT 数据类型可存储的数字范围比 INT 数据类型广。INT 数据类型比 SMALLINT 数据类型的存储范围大,而 SMALLINT 的数值范围又比 TINYINT 类型大。

(1)BIGINT 数据类型存储从 -2^{63} 到 $2^{63}-1$ 范围内的数字。其精度为 19,长度为 8 个字节。

(2)INT 数据类型的存储范围是 $-2\ 147\ 483\ 648 \sim 2\ 147\ 483\ 647$ 。其精度为 10,长度为 4 个字节。

(3)SMALLINT 数据类型的存储范围只有 $-32\ 768 \sim 32\ 767$,其精度为 5,长度为 2 个字节。

(4)TINYINT 数据类型只能存储 $0 \sim 255$ 范围内的数字,其精度为 3,长度为 1 个字节。

4. 精确数值型

精确数值型由整数和小数两部分组成,要求准确地指定使用的小数位数和精度,包括 DECIMAL 和 NUMERIC 两类。两者格式相同,但用法不同,即前者不能用于含有 IDENTITY 关键字的列。

精确数值型的格式如下。

DECIMAL(P,S)和 NUMERIC(P,S)

其中,P 为精度,S 为小数位数。例如,定义成绩列时,要求保留一位小数,则应定义为 DECIMAL(4,1)。

精确数值型的存储范围是 $-10^{38}+1 \sim 10^{38}-1$,长度为 5~17 字节。

5. 浮点型

浮点型也称近似数值型,不能精确表示数据,有可能会损失一些精度,因此被用于对精度要求不高的数值型数据中。例如,分数 $1/3$ 用小数表示为 $0.33333\dots$,该值只能用近似的小数表示,而不可能用精确值表示。

浮点型分为 FLOAT[(N)]和 REAL 两种。通常使用科学计数法表示数据。例如, $1.23E+3$ 、 $-1.256E-3$ 等,其中 E 表示阶数。FLOAT 类型中 N 的取值范围是 1~53,N

默认时,精度为 15,长度为 8。其数据范围为 $-1.79E+308\sim 1.79E+308$ 。REAL 类型精度为 7,数据范围为 $-3.1E+38\sim 3.4E+38$ 。

6. 日期和时间型

日期和时间型数据用于存储日期和时间信息,分为 DATETIME 和 SMALLDATETIME 两类。

(1)DATETIME 型:可表示从 1753 年 1 月 1 日到 9999 年 12 月 31 日的日期和时间数据。其长度为 8 字节,日期和时间分别使用 4 字节存储。

(2)SMALLDATETIME 型:可表示从 1900 年 1 月 1 日至 2079 年 12 月 31 日的日期和时间数据。其长度为 4 字节,日期和时间分别使用 2 字节存储。

日期型数据在表示时可用“/”“-”“.”进行分隔,写法可以按年月日或月日年的顺序。例如,1982.5.9、1982-5-9、1982/5/9、5.9.1982、5-9-1982、5/9/1982 等。

时间表示格式有时:分:时:分:秒:时:分:秒:毫秒、时:分 AM | PM。例如,11:05、11:05 AM 等。

7. 货币型

货币型数据表示正的或负的货币值。货币型分为 MONEY 和 SMALLMONEY 两种类型。

(1)MONEY 型:数据范围为 $-2^{63}\sim 2^{63}-1$,精度为 19,小数位数为 4,长度为 8 字节。MONEY 型与 BIGINT 型相似,只不过 BIGINT 表示整数,没有小数。

(2)SMALLMONEY 型:数据范围为 $-2^{31}\sim 2^{31}-1$,精度为 10,小数位数为 4,长度为 4 字节。SMALLMONEY 型与 INT 型相似,同理,后者表示整数,没有小数。

如果数值超过了上述范围,则可使用 DECIMAL 数据类型代替。

货币型的表示方法:在数据前面加上货币符号“\$”。例如,\$5000000、\$-1.2 等。

8. 二进制型

二进制数据类型包括 BINARY、VARBINARY 和 IMAGE。

(1)定长二进制 BINARY[(N)]:N 位固定的二进制数据。其中,N 的取值范围是 1~8 000。其存储空间的大小是 N+4 个字节。

(2)变长二进制 VARBINARY[(N)]:N 位变长度的二进制数据。其中,N 的取值范围是 1~8 000。其存储空间的大小是 N+4 个字节。

(3)图像型 IMAGE:用于存储图片和照片。此数据类型中存储的数据是以位字符串形式存储的,不是由 SQL Server 解释的,必须由应用程序来解释。例如,应用程序可以使用 BMP、TIEF、GIF 和 JPEG 格式把数据存储在 IMAGE 数据类型中。

9. 特殊类型

(1)TIMESTAMP 型:用于表示 SQL Server 活动的先后顺序,以二进制的格式表示。

TIMESTAMP 数据与插入数据或日期和时间没有关系。若要自动记录表中数据更改的时间,则使用 DATETIME 或 SMALLDATETIME 数据类型记录事件或触发器。

(2)BIT 型:由 1 或 0 组成。当表示真或假、ON 或 OFF 时,使用 BIT 数据类型。

(3)UNIQUEIDENTIFIER 型:以一个 16 位的十六进制数表示全局唯一标识符(GUID)。当需要在多行中唯一标识某一行时可使用 GUID。例如,可使用 UNIQUEIDENTIFIER 数据类型定义一个客户标识代码列,以编辑来自多个国家/地区的总的客户名录。在使用此类型时,应将“是 ROWGUID”域改为“是”。

(4)SQL_VARIANT 类型:一种存储 SQL Server 所支持的各种数据类型(TEXT、NTEXT、TIMESTAMP 和 SQL_VARIANT 除外)值的数据类型。

(5)TABLE 类型:一种特殊的数据类型,存储供以后处理的结果集。TABLE 数据类型只能用于定义 TABLE 类型的局部变量或定义函数的返回值。

注意:①定义数据类型时,只有字符型、精确数值型、FLOAT 型和二进制型才在后面的括号中指定长度。

②填写数据时,字符型、UNICODE 字符型、日期和时间型需要用单引号将数据引上。

2.2.2 用户定义的数据类型

用户定义的数据类型是基于系统提供的数据类型。在处理不同表或数据库中的数据元素时,用户定义的数据类型能进一步完善数据类型,确保其一致性。用户定义数据类型是针对特定的数据库来定义的。在 MODEL 数据库中创建的用户定义数据类型自动包含到以后创建的所有数据库中。

1. 创建用户定义的数据类型

在创建用户定义的数据类型时必须指定新数据类型名称和新数据类型所依据的系统数据类型。数据类型既可使用 SQL Server Management Studio 创建,也可用 T-SQL 语句创建。

方法一:使用 SQL Server Management Studio 创建用户定义的数据类型。

(1)在“对象资源管理器”窗格中依次展开要创建用户定义数据类型的数据库→可编程性→类型→用户自定义数据类型。

(2)右击“用户定义数据类型”,在弹出的快捷菜单中选择“新建用户定义数据类型”选项,进入“新建用户定义数据类型”窗口,如图 2-1 所示。

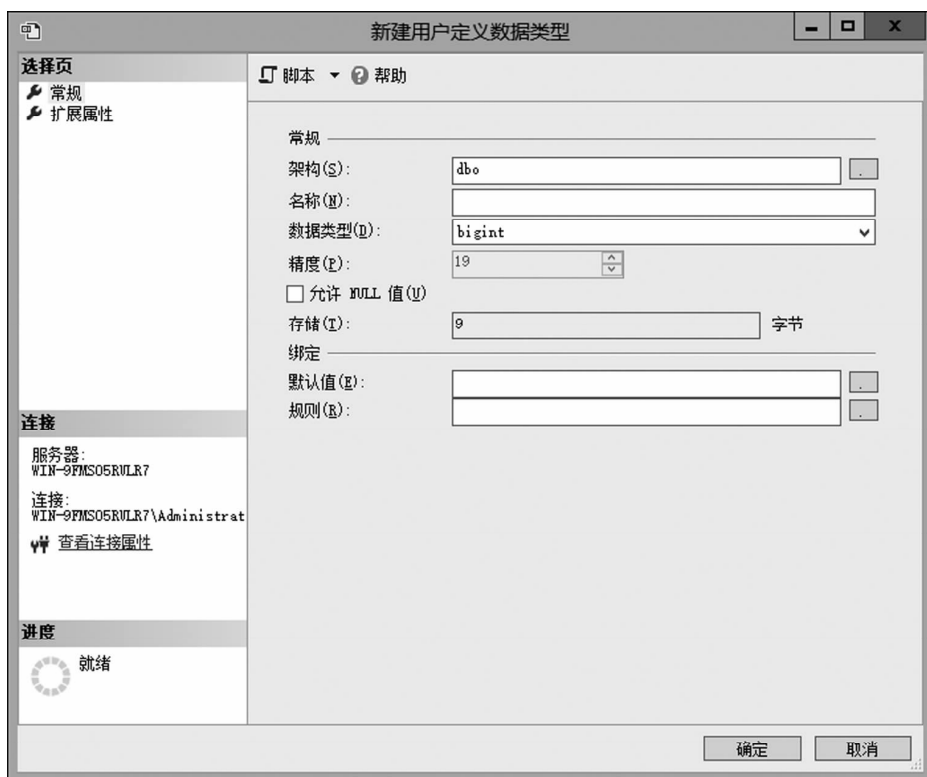


图 2-1 “新建用户定义数据类型”窗口

(3) 输入新建数据类型的名称,并在“数据类型”下拉列表框中选择数据类型。

(4) 在“精度”数值框中可以更改此数据类型可存储的最大数据长度,但只适用于 CHAR、NCHAR、VARCHAR、BINARY 和 VARBINARY 类型。

(5) 若允许此数据类型接受空值,则选中“允许 NULL 值”复选框。

(6) 在“默认值”和“规则”文本框中选择一个规则或默认值将其绑定到用户定义的数据类型上。此项将在后面进行详细介绍。

方法二:使用 T-SQL 语句创建用户定义的数据类型。

语法格式如下。

```
SP_ADDTYPE 类型名, 系统数据类型 [, 是否为空] [, 架构名]
```

说明:

- 类型名:指定新的数据类型名称,必须遵照标识符的命名规则。
- 系统数据类型:指定系统所提供的数据类型。
- 是否为空:指定新数据类型处理空值时是否允许数据值为空。空即从未在表中填写过数据,特别要注意空格不为空、数据被删除后也不为空(除非修改为 NULL)。为空用 'NULL' 表示,非空用 'NOT NULL' 表示,默认为空。

- 架构名:指定新数据类型的创建者或所有者。

【例 2.1】 创建数据类型 NEWINT,指定其所属类型为 INT,不允许为空。

```
SP_ADDTYPE NEWINT, INT, 'NOT NULL'
```

2. 删除用户定义的数据类型

方法一:使用 SQL Server Management Studio 删除用户定义的数据类型。

- (1)展开“用户定义数据类型”,右侧窗口出现已经定义的数据类型。
- (2)右击要删除的数据类型,在弹出的快捷菜单中选择“删除”命令,如图 2-2 所示。

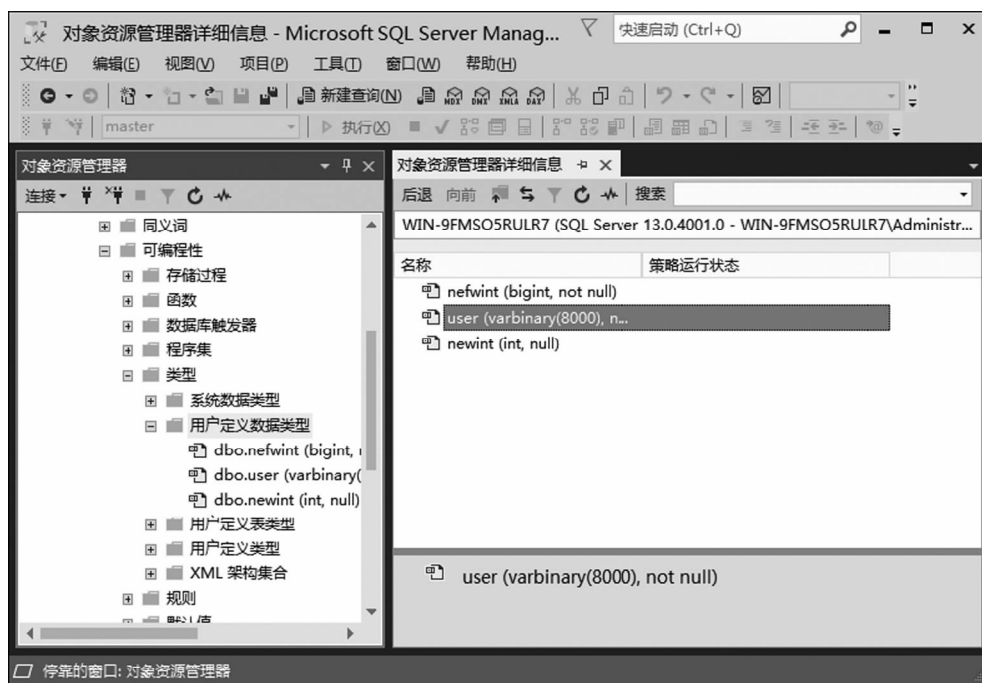


图 2-2 删除用户定义的数据类型

方法二:使用 T-SQL 语句删除用户定义的数据类型。

语法格式如下。

```
SP_DROPTYPE '用户定义的数据类型名'
```

【例 2.2】 删除用户定义的数据类型 NEWINT。

```
SP_DROPTYPE 'NEWINT'
```

2.3 表的创建

确定好数据类型后就可以创建表了。表既可通过 SQL Server Management Studio 创建,也可使用 T-SQL 语句创建。

方法一:使用 SQL Server Management Studio 创建表。

【例 2.3】 创建学生表,其结构见表 2-1。

- (1) 展开要创建表的数据库 XG,右击“表”,在弹出的快捷菜单中选择“新建表”命令。
- (2) 在弹出的“属性”窗格中输入各列的名称、数据类型、长度等,如图 2-3 所示。



图 2-3 创建表

(3) 输入完成后,单击工具栏中的“保存”按钮,保存为“学生表”。如果要将学号设置为主键,则选中学号这一行,右击并选择“设置主键”命令。表创建成功后,在左侧窗格中选中表,右击并选择“刷新”命令,则创建的“学生表”即展现出来。

方法二:使用 T-SQL 语句创建表。

语法格式如下。

```
CREATE TABLE 表名 (列名 1 数据类型 [列级完整性约束]
                    [,列名 2 数据类型 [列级完整性约束]]
                    .....
                    [,列名 N 数据类型 [列级完整性约束]]
                    )
```

说明:

- 表名:二维表的名称,要符合标识符命名规则。
- 列名:字段名,指定表中列的名称。
- 数据类型:字段的数据类型,指明了该列中的数据所允许输入的数据类型。
- 列级完整性约束:指明了该列中所允许输入的数据是否为空、是否可重复、数据的范围等。例如,PRIMARY KEY 表示主键列,表明此列中的数据不允许重复且不为空。NOT NULL 标明此列数据不允许为空。

若有多个列,则在每个列名定义完后加上逗号分隔,但最后一个列的后面不用逗号。

注意:在 T-SQL 语句中写汉字时,将输入法切换到中文状态,写其他非中文符号(如逗号、空格、括号等)时,应切换到英文状态。另外,创建表的完整格式可参见联机帮助。

【例 2.4】 创建学生表,其结构见表 2-2。

```
CREATE TABLE 学生表
(学号 CHAR(10) PRIMARY KEY,
  姓名 VARCHAR(10) NOT NULL,
  专业 VARCHAR(6) NULL,
  年龄 TINYINT NULL,
  出生日期 SMALLDATETIME NULL,
  身份证号 CHAR(18) NULL,
  特长 TEXT NULL
)
```

注意:TINYINT 格式不能在后面的括号中指定长度,此点在数据类型中已提到,应多加注意。