

第 3 章 SQL Server 数据库

SQL Server 2019 的数据库是所涉及的对象以及数据的集合,它不仅反映数据本身的内容,而且反映对象以及数据之间的联系。对数据库进行操作是开发人员的一项重要工作。

本章主要介绍 SQL Server 2019 数据库的基本概念,以及创建、删除、修改数据库等基本操作。

通过学习本章,读者应掌握以下内容:

- 了解数据库及其对象;
- 熟练掌握用对象资源管理器创建和管理数据库的方法;
- 熟练掌握用 T-SQL 语句创建和管理数据库的方法。

3.1 SQL Server 数据库概述

SQL Server 2019 数据库就是存放有组织的数据集合的容器,以操作系统文件的形式存储在磁盘上,由数据库系统进行管理和维护。数据库中的数据和日志信息分别保存在不同的文件中,而且各文件仅在一个数据库中使用。文件组是命名的文件集合,用于帮助数据布局和管理任务,例如备份和还原操作。

3.1.1 数据库文件

数据库文件是存放数据库数据和数据库对象的文件。一个数据库可以有一个或多个数据库文件,一个数据库文件只属于一个数据库。

1. 数据库文件的分类

SQL Server 2019 数据库具有以下 3 种类型的文件。

(1) 主数据文件:主数据文件包含数据库的启动信息,是数据库的起点,指向数据库中的其他文件;存储用户数据和对象;SQL Server 数据库的主体,每个数据库有且仅有一个主数据文件。实际的文件都有两种名称,即操作系统文件名和逻辑文件名(在 T-SQL 语句中使用)。主数据文件的默认文件扩展名是 mdf。

(2) 次要数据文件:除主数据文件以外的所有其他数据文件都是次要数据文件,也称辅助数据文件,可用于将数据分散到多个磁盘上。如果数据库超过了单个 Windows 文件的最大大小,可以使用次要数据文件,这样数据库就能继续增长;数据库中可以有多个或者没有次要数据文件;名字尽量与主数据文件名相同。次要数据文件的默认文件扩展名是 ndf。



SQL Server
数据库概述

(3) 事务日志文件: 用来记录数据库更新情况的文件, 每个数据库至少有一个事务日志文件, 事务日志文件不属于任何文件组。凡是对数据库进行的增加、删除、修改等操作都会记录在事务日志文件中。当数据库被破坏时可以利用事务日志文件恢复数据库的数据, 从而最大限度地减少由此带来的损失。SQL Server 中采用“提前写”方式的事务, 即对数据库的修改先写入事务日志, 再写入数据库。

日志文件还可以通过事务有效地维护数据库的完整性。与数据文件不同, 日志文件不存放数据, 不包含数据页, 由一系列的日志记录组成, 日志文件也不包含在文件组内。日志文件的默认扩展名是 ldf。

SQL Server 2019 不强制使用 mdf、ndf 和 ldf 文件扩展名, 但使用它们有助于标识文件的各种类型和用途。

2. 逻辑文件名和物理文件名

SQL Server 2019 的文件拥有两个名称, 即逻辑文件名和物理文件名。当使用 T-SQL 命令语句访问某个文件时, 必须使用该文件的逻辑名。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。

(1) 逻辑文件名(logical_file_name): 它是在所有 T-SQL 语句引用物理文件时使用的名称。逻辑文件名必须符合 SQL Server 标识符规则, 而且在数据库中的逻辑文件名必须是唯一的。

(2) 物理文件名(os_file_name): 它是包括目录路径的物理文件名, 必须符合操作系统的文件命名规则。

3. 文件大小

SQL Server 2019 数据文件除需要描述物理文件名与逻辑文件名外, 还需要描述文件大小, 包含初始大小(size)、最大值(maxsize)和增量(filegrowth) 3 个参数。文件的大小可以从最初指定的初始大小开始按增量来增长, 当文件增量超过最大值时将出错, 文件无法正常建立, 也就是数据库无法创建。

如果没有指定最大值, 文件可以一直增长到用完磁盘上的所有可用空间。如果 SQL Server 作为数据库嵌入某个应用程序, 而该应用程序的用户无法迅速与系统管理员联系, 则不指定文件最大值就特别有用, 用户可以使文件根据需要自动增长, 以减轻监视数据库中的可用空间和手动分配额外空间的管理负担。

3.1.2 数据库文件组

为便于分配和管理, 可以将数据库对象和文件一起分成文件组。SQL Server 2019 有以下两种类型的文件组。

(1) 主文件组: 包含主数据文件和任何没有明确分配给其他文件组的数据文件。系统表的所有页都分配在主文件组中。在 SQL Server 2019 中用 PRIMARY 表示主文件组的名称。主文件组由系统自动生成, 供用户使用, 用户不能修改或删除。

(2) 用户自定义文件组：在 CREATE DATABASE 或 ALTER DATABASE 语句中使用 FILEGROUP 关键字指定的任何文件组。

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。

一个文件不可以是多个文件组的成员。表、索引和大型对象数据可以与指定的文件组相关联,在这种情况下,它们的所有页将被分配到该文件组;也可以对表和索引进行分区,已分区表和索引的数据被分割为单元,每个单元可以放置在数据库的单独文件组中。

每个数据库中均有一个文件组被指定为默认文件组。如果创建表或索引时未指定文件组,则将假定所有页都从默认文件组分配。注意,一次只能将一个文件组作为默认文件组。db_owner 固定数据库角色成员可以将默认文件组从一个文件组切换到另一个。如果没有指定默认文件组,则将主文件组作为默认文件组。

SQL Server 2019 包含 4 个系统数据库和若干数据库对象。

3.1.3 数据库对象

SQL Server 2019 数据库中的数据在逻辑上被组织成一系列对象,当一个用户连接到数据库后,所看到的是这些逻辑对象,而不是物理的数据库文件。

SQL Server 2019 中有以下数据库对象。

(1) 表：数据库中的表与日常生活中使用的表格类似,由列和行组成。其中每列都代表一个相同类型的数据。每列又称一个字段,每列的标题称为字段名。每行包括若干列信息。一行数据称为一个元组或一条记录,它是有一定意义的信息组合,代表一个实体或联系。一个数据表由一条或多条记录组成,没有记录的表称为空表。每个表中通常都有一个主关键字,用于唯一标识一条记录。

(2) 索引：某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单,它提供了数据库中编排表中数据的内部方法。

(3) 视图：视图看上去和表相似,具有一组命名的字段和数据项,但它其实是一个虚拟的表,在数据库中并不实际存在。视图是由查询数据表或其他视图产生的,它限制了用户能看到和修改的数据。由此可见,视图可以用来控制用户对数据的访问,并能简化数据的显示,即通过视图只显示那些需要的数据信息。

(4) 关系图表：关系图表其实就是数据表之间的关系示意图,利用它可以编辑表与表之间的关系。

(5) 默认值：默认值是当在表中创建列或插入数据时对没有指定其具体值的列或列数据项赋予事先设定好的值。

(6) 约束：SQL Server 实施数据一致性和数据完整性的方法,或者说是一套机制,它包括主键(PRIMARY KEY)约束、外键(FOREIGN KEY)约束、唯一键(UNIQUE)约束、检查(CHECK)约束、默认值(DEFAULT)约束和允许空 6 种机制。

(7) 规则：用来限制数据表中字段的有限范围,以确保列中数据完整性的一种方式。

(8) 触发器：一种特殊的存储过程,与表格或某些操作相关联。当用户对数据进行插入、修改、删除或对数据表进行建立、修改、删除时激活,并自动执行。

(9) 存储过程：一组经过编译的可以重复使用的 T-SQL 代码的组合。它是经过编译存储到数据库中的，所以运行速度要比执行相同的 SQL 语句快。

(10) 登录：SQL Server 访问控制允许连接到服务器的账户。

(11) 用户：用户是指拥有一定权限的数据库的使用者。

(12) 角色：数据库操作权限的集合，可以将角色关联到同一类级别的用户。

3.1.4 系统数据库

SQL Server 2019 包含 master、model、msdb、tempdb 4 个系统数据库。在创建任何数据库之前，用户在 SQL Server Management Studio 工具中可以看到这些系统数据库。

1. master 数据库

master 数据库记录 SQL Server 2019 实例的所有系统级信息，包括实例范围的元数据（如登录账户）、端点、链接服务器和系统配置设置；记录所有其他数据库是否存在以及这些数据库文件的位置；记录 SQL Server 的初始化信息。因此，如果 master 数据库不可用，则 SQL Server 无法启动。

注意：不能在 master 数据库中创建任何用户对象（如表、视图、存储过程或触发器）。master 数据库包含 SQL Server 实例使用的系统级信息（如登录信息和配置选项设置）。

用户不能在 master 数据库中执行下列操作。

- (1) 添加文件或文件组。
- (2) 更改排序规则。默认排序规则为服务器排序规则。
- (3) 更改数据库所有者。master 数据库归 dbo 所有。
- (4) 创建全文目录或全文索引。
- (5) 在数据库的系统表上创建触发器。
- (6) 删除数据库。
- (7) 从数据库中删除 guest 用户。
- (8) 参与数据库镜像。
- (9) 删除主文件组、主数据文件或日志文件。
- (10) 重命名数据库或主文件组。
- (11) 将数据库设置为 OFFLINE。
- (12) 将数据库或主文件组设置为 READ ONLY。

2. model 数据库

model 数据库是在 SQL Server 2019 实例上创建的所有数据库的模板，对 model 数据库进行的修改（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数据库。

在用 CREATE DATABASE 语句新建数据库时，将通过复制 model 数据库中的内容创建数据库的第一部分，然后用空页填充新数据库的剩余部分。在 SQL Server 实例上创

建的新数据库的内容在开始创建时和 model 数据库完全一样。

如果修改 model 数据库,之后创建的所有数据库都将继承这些修改。可以设置权限、数据库选项或者添加对象,如表、函数或存储过程等。

3. msdb 数据库

msdb 数据库由 SQL Server 代理计划警报和作业,以及与备份和恢复相关的信息,尤其是 SQL Server Agent 需要使用它来执行安排工作和警报、记录操作者等操作。

4. tempdb 数据库

tempdb 数据库是连接到 SQL Server 2019 实例的所有用户都可用的全局资源,保存所有临时表和临时存储过程。另外,它还用来满足所有其他临时存储要求,例如存储 SQL Server 2019 生成的工作表。

每次启动 SQL Server 时都要重新创建 tempdb 数据库,以保证系统启动时该数据库总是空的;在断开连接时会自动删除临时表和存储过程,并且在系统关闭后没有活动连接,因此 tempdb 数据库中不会有任何内容从一个 SQL Server 会话保存到另一个会话。

tempdb 数据库用于保存以下内容。

- (1) 显式创建的临时对象,如表、存储过程、表变量或游标。
- (2) 所有版本的更新记录(如果启用了快照隔离)。
- (3) SQL Server Database Engine 创建的内部工作表。
- (4) 创建或重新生成索引时临时排序的结果(如果指定了 SORT IN TEMPDB)。

3.2 创建数据库

若要创建数据库,必须确定数据库的名称、所有者、大小以及存储该数据库的文件和文件组。

创建数据库的注意事项如下。

(1) 创建数据库需要一定许可,在默认情况下,只有系统管理员和具有创建数据库角色的登录账户的拥有者才可以创建数据库。数据库被创建后,创建数据库的用户自动成为该数据库的所有者。

(2) 创建数据库的过程实际上就是为数据库设计名称、设计所占用的存储空间和存放文件位置的过程等,数据库名称必须遵循 SQL Server 命名规范。

(3) 所有的新数据库都是系统样本数据库 model 的副本。

(4) 单个数据库可以存储在单个文件上,也可以跨越多个文件存储。

(5) 数据库的大小可以被增大或者收缩。

(6) 当新的数据库创建时,SQL Server 自动更新 sysdatabases 系统表。

在 SQL Server 2019 中创建数据库主要有两种方式:一种方式是在 SQL Server Management Studio 中使用对象资源管理器创建数据库;另一种方式是通过在查询窗口中执行 T-SQL 语句创建数据库。



创建数据库

3.2.1 使用对象资源管理器创建数据库

使用对象资源管理器创建数据库比使用 T-SQL 语句更容易,具体操作如下。

(1) 单击“开始”按钮,选择 SQL Server Management Studio 命令,打开 SQL Server Management Studio 窗口,设置好服务器类型、服务器名称、身份验证、用户名和密码,并单击“连接”按钮。

(2) 在“对象资源管理器”窗格中选择“数据库”选项并右击,在弹出的快捷菜单中选择“新建数据库”命令,如图 3.1 所示。

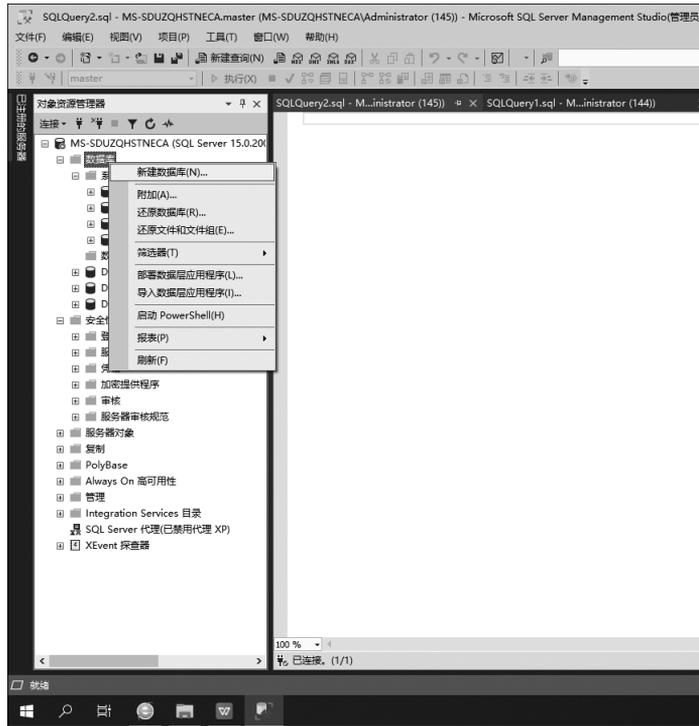


图 3.1 使用“对象资源管理器”创建数据库

(3) 弹出“新建数据库”对话框,该对话框由“常规”“选项”“文件组”3 个选项组成。例如创建 jxgl 学生管理数据库,可在“常规”选项的“数据库名称”文本框中输入 jxgl,如图 3.2 所示。在这里可以通过“常规”选项对数据文件和日志文件的路径进行设置,单击“路径”列(向右滑动图 3.2 中下方滚动条可见)后面的“浏览”按钮,弹出“定位文件夹”对话框,如图 3.3 所示。

(4) 在“新建数据库”对话框的各个选项中可以设置它们的参数值,例如在“数据库文件”编辑框中的“逻辑名称”列输入文件名;在“初始大小”列设置初始值大小,单击“自动增长/最大大小”列后的“浏览”按钮,弹出“更改自动增长”对话框,可以按多种方式设置自动增长的大小等,如图 3.4 所示。



图 3.2 “新建数据库”对话框

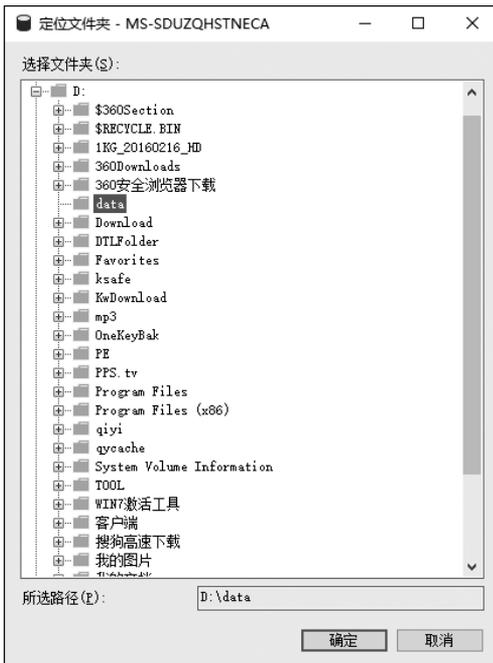


图 3.3 “定位文件夹”对话框

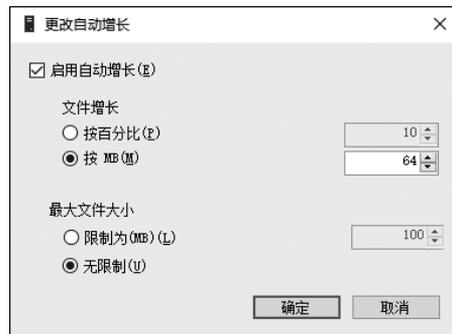


图 3.4 “更改自动增长”对话框

(5) 单击“确定”按钮,在“数据库”的树形结构中就可以看到新建的 jxgl 数据库,如图 3.5 所示。

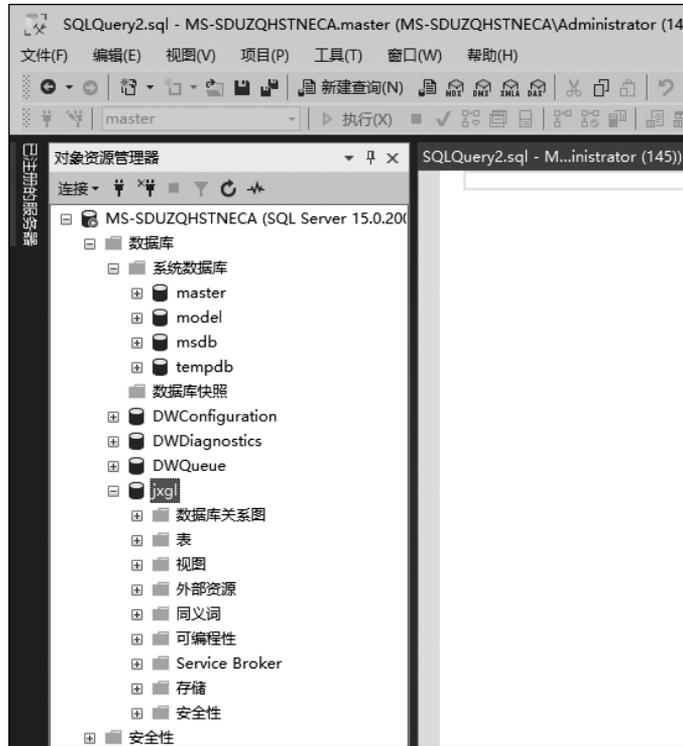


图 3.5 jxgl 数据库创建成功

3.2.2 使用 T-SQL 语句创建数据库

SQL 提供了创建数据库语句 CREATE DATABASE,其语法格式如下:

```
CREATE DATABASE database_name
[ON [PRIMARY] [<filespec>[, ...n]] [,<filegroupspec>[, ...n]]]
[LOG ON(<filespec>[, ...n] ) ]
[COLLATE collation_name]
[FOR ATTACH]
```

进一步把<filespec>定义为

```
[PRIMARY]
([NAME=logical_file_name,]
FILENAME='os_file_name'
[, SIZE=size]
[, MAXSIZE=(max_size|UNLIMITED)]
[, FILEGROWTH=growth_increment]) [, ...n]
```

把<filegroupspec>定义为

```
FILEGROUP filegroup_name<filespec>[DEFAULT][,...n]
```

1. 语法中的符号及参数说明

- (1) []: 表示可选语法项,省略时各参数取默认值。
- (2) [...n]: 表示该选项的内容可以重复多次。
- (3) { } : 表示必选项,当有相应子句时,{ }中的内容是必选的。
- (4) < > : 表示在实际的语句中要用相应的内容替代。
- (5) 文字大写: 说明该文字是 T-SQL 的关键字。
- (6) 文字小写: 说明该文字是用户提供的 T-SQL 语法的参数。
- (7) database_name: 用户所要创建的数据库名称,最长不能超过 128 个字符,在一个 SQL Server 实例中,数据库名称是唯一的。
- (8) ON: 指定存放数据库的数据文件信息,说明数据库是根据后面的参数创建的。
- (9) PRIMARY: 用于指定主文件组中的文件。主文件组的第一个由<filespec>指定的文件是主数据文件。若不指定 PRIMARY 关键字,则在命令中列出的第一个文件将被默认为主数据文件。
- (10) LOG ON: 指定日志文件的明确定义。如果没有此项,系统会自动创建一个为所有数据文件总和 1/4 或 512KB 的日志文件。
- (11) COLLATE collation_name: 指定数据库默认排序规则,规则名称可以是 Windows 排序规则名称,也可以是 SQL 排序规则名称。
- (12) <filespec>: 指定文件的属性。
 - NAME=logical_file_name: 定义数据文件的逻辑名,此名称在数据库中必须唯一。
 - FILENAME='os_file_name': 定义数据文件的物理名,包括物理文件使用的路径名和文件名。
 - SIZE=size: 在文件属性中定义文件的初始值,指定为整数。
 - MAXSIZE=max_size: 在文件属性中定义文件可以增长到的最大值,可以使用 KB、MB、GB 或 TB 单位,默认是 MB,指定为整数。如果没有指定或写为 UNLIMITED,那么文件将增长到磁盘变满为止。
 - FILEGROWTH=growth_increment: 定义文件的自动增长,growth_increment 定义每次增长的大小。
- (13) FILEGROUP filegroup_name: 定义文件组的控制。
 - filegroup_name: 必须是数据库中唯一的,不能是系统提供的名称 PRIMARY。
 - DEFAULT: 指定命名文件组为数据库中的默认文件组。

2. 注意事项

- (1) 创建用户数据库后要备份 master 数据库。

(2) 所有数据库都至少包含一个主文件组,所有系统表都分配在主文件组中。数据库还可以包含用户自定义文件组。

(3) 每个数据库都有一个所有者,可在数据库中执行某些特殊的活动。数据库所有者是创建数据库的用户,也可以使用 `sp_changedbowner` 更改数据库所有者。

(4) 创建数据库的权限默认地授予 `sysadmin` 和 `dbcreator` 固定服务器角色的成员。

【例 3.1】 创建一个名为 `jxgl` 的数据库。其中主数据文件为 10MB,最大文件大小不受限制,每次增长 1MB;事务日志文件大小为 1MB,最大文件大小不受限制,文件每次增长 10%。

```
CREATE DATABASE jxgl                /* 数据库名 */
ON
PRIMARY                            /* 主文件组 */
(NAME='jxgl',                       /* 主数据文件逻辑名 */
FILENAME='D:\Data\jxgl.mdf ',
SIZE=10MB, MAXSIZE=UNLIMITED, FILEGROWTH=1MB)
LOG ON
(NAME='jxgl_log',
FILENAME=' D:\Data\jxgl_log.ldf',
SIZE=1MB,MAXSIZE=UNLIMITED, FILEGROWTH=10%)
GO
```

语句输入完成后,按 F5 键或单击工具栏中的“执行”按钮,将执行所输入的语句,创建数据库,如图 3.6 所示。

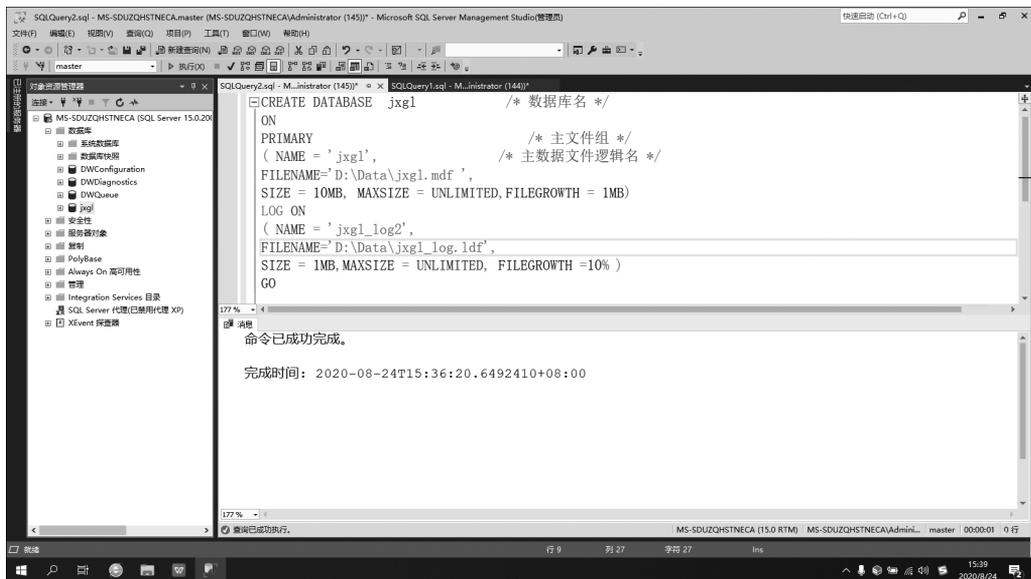


图 3.6 例 3.1 的执行结果

【例 3.2】 创建 `test` 数据库,其包含一个主文件组和两个用户自定义文件组。

```

CREATE DATABASE test
ON PRIMARY                                /* 定义在主文件组上的文件 */
(NAME=pri_file1,
FILENAME=' D:\Data\pri_file1.mdf ',
SIZE=10,MAXSIZE=50,FILEGROWTH=15%),
(NAME=pri_file2,
FILENAME=' D:\Data\pri_file2.ndf ',
SIZE=10,MAXSIZE=50,FILEGROWTH=15%),
FILEGROUP Grp1                            /* 定义在用户自定义文件组 Grp1 上的两个文件 */
(NAME=Grp1_file1,
FILENAME=' D:\Data\Grp1_file1.ndf ',
SIZE=10,MAXSIZE=50,FILEGROWTH=5),
(NAME=Grp1_file2,
FILENAME=' D:\Data\Grp1_file2.ndf ',
SIZE=10,MAXSIZE=50,FILEGROWTH=5),
FILEGROUP Grp2                            /* 定义在用户自定义文件组 Grp2 上的两个文件 */
(NAME=Grp2_file1,
FILENAME=' D:\Data\Grp2_file1.ndf ',
SIZE=10,MAXSIZE=50,FILEGROWTH=5),
(NAME=Grp2_file2,
FILENAME=' D:\Data\Grp2_file2.ndf ',
SIZE=10,MAXSIZE=50,FILEGROWTH=5)
LOG ON                                    /* 定义事务日志文件 */
(NAME='test_log',
FILENAME=' D:\Data\test_log.ldf ',
SIZE=5,MAXSIZE=25,FILEGROWTH=5)
GO

```

3.2.3 事务日志

SQL 创建数据库的时候会同时创建事务日志文件。

事务日志用于存放恢复数据时所需的信息，是数据库中已发生的所有修改和执行每次修改的事务的一连串记录。当数据库损坏时，管理员可以使用事务日志还原数据库。每个数据库必须至少拥有一个事务日志文件，并允许拥有多个日志文件。事务日志文件的扩展名为 ldf，日志文件的大小至少是 512KB。

事务日志是针对数据库改变所做的记录，它可以记录针对数据库的任何操作，并将记录结果保存在独立的文件中。对于任何事务过程，事务日志都有非常全面的记录，根据这些记录可以将数据文件恢复到事务前的状态。

1. 事务日志文件和数据文件必须分开存放

其优点如下。

- (1) 事务日志文件可以单独备份。
- (2) 有可能从服务器失效的事件中将服务器恢复到最近的状态。
- (3) 事务日志不会抢占数据库的空间。
- (4) 可以很容易地监测到事务日志的空间。
- (5) 在向数据文件和事务日志文件写入数据时会产生较少的冲突,这有利于提高系统的性能。

SQL Server 事务日志采用提前写入的方式。

2. 事务日志的工作过程

在 SQL Server 中,事务是一个完整的操作的集合。虽然一个事务中可能包含很多 SQL 语句,但在处理上就像它们是同一个操作一样。

为了维护数据的完整性,事务必须彻底完成或者根本不执行。如果一个事务只是被部分执行,并作用于数据库,那么数据库可能被损坏或数据的一致性遭到破坏。

SQL Server 使用数据库的事务日志来防止没有完成的事务破坏数据。

事务日志的工作过程如下。

- (1) 应用程序发出一个修改数据库中的对象的事务。
- (2) 当这个事务开始时,事务日志会记录一个事务开始的标志,并将被影响的数据页从磁盘读入缓冲区。
- (3) 事务中的每个数据更改语句都被记录在日志文件中,日志文件将记录一个提交事务的标记。每个事务都会以这种方式记录在事务日志中并被立即写到硬盘上。
- (4) 在缓冲区中修改相应的数据,这些数据一直在缓冲区中,在检查点进程发生时检查点进程把所有修改过的数据页写到数据库中,并在事务日志中写入一个检查点标志,这个标志用于在数据库恢复过程中确定事务的起点和终点,以及哪些事务已经作用于数据库。

随着数据库数据的不断变化,事务日志文件不断增大,因此必须把它们备份出来,为更多的事务提供空间。在备份时,事务日志文件会被截断。

事务日志文件包含了在系统发生故障时恢复数据库需要的所有信息。一般来说,事务日志文件的初始大小是以数据文件大小的 10%~25% 为起点的,根据数据增长的情况和修改的频率进行调整。SQL Server 2019 中的数据文件和事务日志文件不能存放在压缩文件系统或共享网络目录等远程的网络驱动器上。



管理和维护
数据库

3.3 管理和维护数据库

在创建数据库后开始使用数据库,通常有打开、查看、修改、删除等方式对数据库进行管理和维护。

3.3.1 打开或切换数据库

当用户登录数据库服务器连接 SQL Server 后,用户需要连接数据库服务器中的数据库才能使用数据库中的数据,默认情况下用户连接的是 master 数据库。

打开或切换数据库可以通过对象资源管理器,也可以使用 T-SQL 语句打开或切换数据库。在对象资源管理器中打开数据库只需展开该数据库结点,并单击要打开的数据库,此时右侧窗格中列出的是当前打开的数据库的对象。用户还可以直接使用数据库下拉列表打开并切换数据库,如图 3.7 所示。

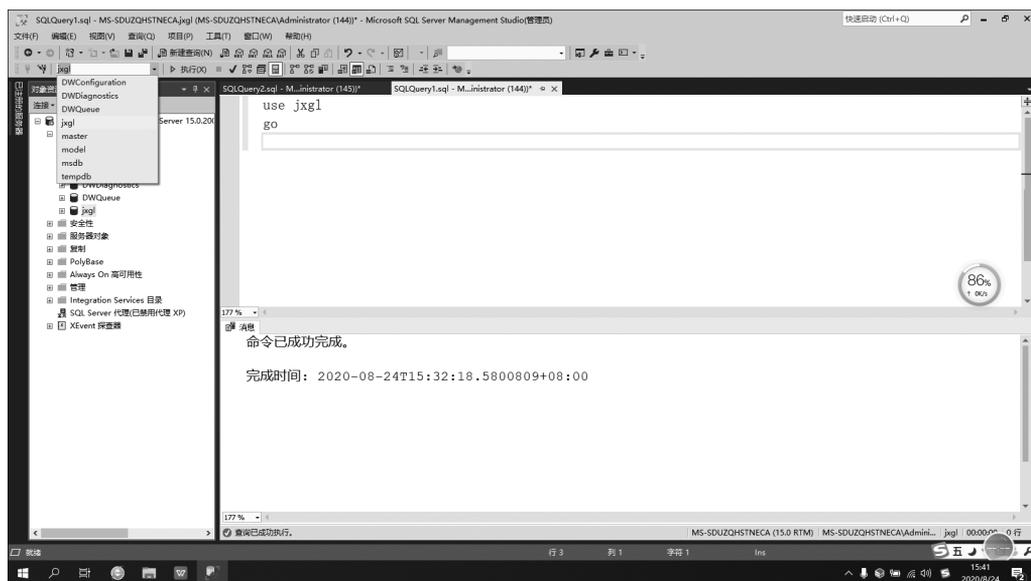


图 3.7 使用数据库下拉列表打开并切换数据库

使用 T-SQL 语句打开数据库的语法格式如下:

```
USE database_name
```

database_name 为要打开并切换的数据库名。

3.3.2 查看数据库信息

对于已有的数据库,可以使用 SQL Server Management Studio 或者 T-SQL 语句查看数据库信息。

1. 使用 SQL Server Management Studio 查看数据库信息

在 SQL Server Management Studio 中右击数据库名,在弹出的快捷菜单中选择“属性”命令,出现如图 3.8 所示的“数据库属性-jxgl”对话框,显示 jxgl 数据库上次备份的日

期、数据库日志上次备份的日期、名称、所有者、创建日期、大小、可用空间、用户数、排序规则名称等信息。

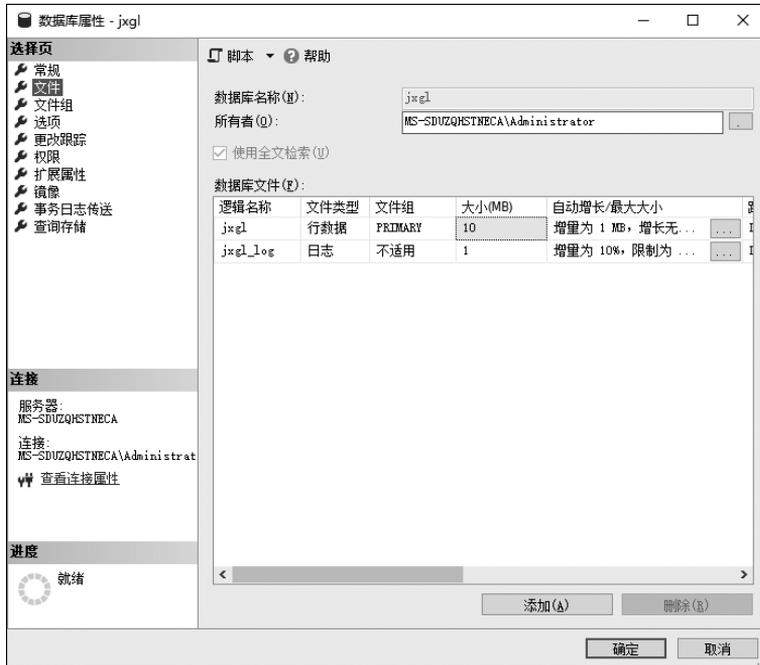


图 3.8 “数据库属性-jxgl”对话框

单击“文件”“文件组”“选项”“权限”“扩展属性”“镜像”“事务日志传送”属性页,可以查看数据库文件、文件组、数据库选项、权限、扩展属性、数据库镜像、事务日志传送等属性。

2. 使用 T-SQL 语句查看数据库信息

使用系统存储过程 `sp_helpdb` 可以查看数据库信息,其语法格式如下:

```
[EXECUTE] sp_helpdb [database_name]
```

在执行该存储过程时,如果给定了数据库名作为参数,则显示该数据库的相关信息。如果省略“数据库名”参数,则显示服务器中所有数据库的信息。

3.3.3 修改数据库配置

1. 使用对象资源管理器修改数据库配置

(1) 启动 SQL Server Management Studio,连接数据库实例,展开“对象资源管理器”中的树形目录,定位到要修改的数据库上。

(2) 右击要修改的数据库,如 `jxgl`,在弹出的快捷菜单中选择“属性”命令(见图 3.8),

然后选择左侧相应的选项,在右侧窗格中会弹出相关内容,其余操作与创建数据库的过程相似,请读者自己完成。

2. 使用 T-SQL 语句修改数据库配置

用户可以使用 T-SQL 语句修改数据库,其语法格式如下:

```
ALTER DATABASE database_name
{ ADD FILE <filespec>[,...n][TO FILEGROUP filegroup_name]
  |ADD LOG FILE <filespec >[,...n]
  |REMOVE FILE logical_file_name
  |ADD FILEGROUP filegroup_name
  |REMOVE FILEGROUP filegroup_name
  |MODIFY FILE <filespec >
  |MODIFY NAME=new_database_name
  |MODIFY FILEGROUP filegroup_name
  (filegroup_property|NAME=new_filegroup_name)
  |SET <optionspec>[,...n]
}
```

对语法中的各参数说明如下。

- (1) ADD FILE <filespec> [,...n][TO FILEGROUP filegroup_name]: 向指定的文件组添加新的数据文件。
- (2) ADD LOG FILE < filespec >[,...n]: 增加新的日志文件。
- (3) REMOVE FILE logical_file_name: 从数据库系统表中删除文件描述和物理文件。
- (4) ADD FILEGROUP filegroup_name: 增加一个文件组。
- (5) REMOVE FILEGROUP filegroup_name: 删除指定的文件组。
- (6) MODIFY FILE < filespec >: 修改物理文件。
- (7) MODIFY NAME=new_database_name: 重命名数据库。
- (8) MODIFY FILEGROUP filegroup_name: 修改指定文件组的属性。
- (9) SET <optionspec>[,...n]: 按<optionspec>的指定设置数据库的一个或多个选项。

注意: 只有 sysadmin/dbcreator/db_owner 角色的成员能执行该语句。

【例 3.3】 用 T-SQL 语句把 jxgl 重命名为“学生管理数据库”。

```
ALTER DATABASE jxgl
MODIFY NAME=学生管理数据库
```

执行后会得到如图 3.9 所示的结果。

【例 3.4】 将两个数据文件和一个事务日志文件添加到 test 数据库中,代码如下:

```
ALTER DATABASE test
ADD FILE
(NAME=test1,
```

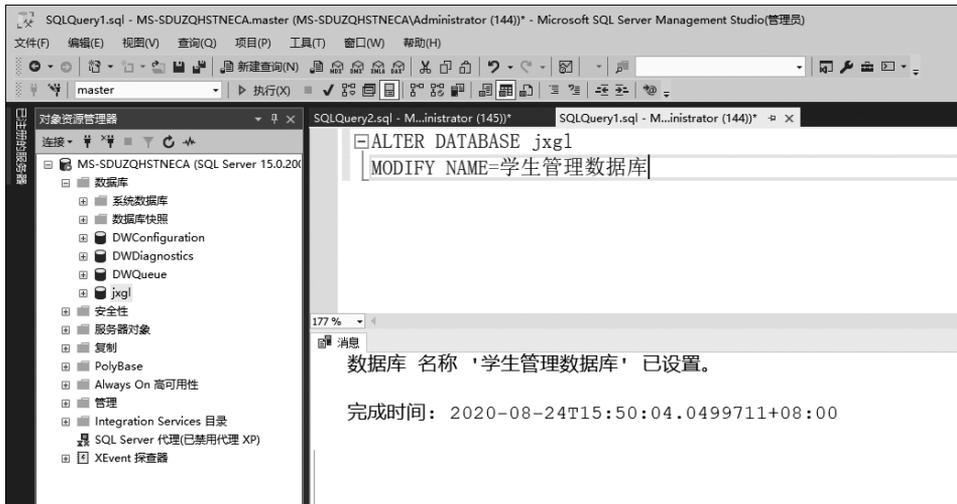


图 3.9 例 3.3 的执行结果

```

FILENAME= 'D:\Data\test1.ndf',
SIZE=5MB,MAXSIZE=100MB,FILEGROWTH=5MB)
(NAME=test2,
FILENAME= 'D:\Data\test2.ndf'
SIZE=3MB,MAXSIZE=20MB,FILEGROWTH=3MB)
GO
ALTER DATABASE test
ADD LOG FILE
(NAME=test_log1,
FILENAME= 'D:\Data\test_log1.ldf'
SIZE=5MB,MAXSIZE=100MB,FILEGROWTH=5MB)
GO

```

【例 3.5】 为数据库 jxgl 添加一个新的文件组。

```

ALTER DATABASE test
ADD FILEGROUP jxgl_group

```

【例 3.6】 对于 test 数据库中逻辑名为 test 的数据文件,将其初始大小、最大文件大小及增长量分别更改为 50MB、200MB 和 10MB,代码如下:

```

ALTER DATABASE test
MODIFY FILE
(NAME=test,SIZE=50MB,MAXSIZE=200MB,FILEGROWTH=10MB)

```

3.3.4 分离与附加数据库

1. 分离数据库

在 SQL Server 运行时,在 Windows 中不能直接复制 SQL Server 数据库文件,如果

想复制,就要先将数据库文件从 SQL Server 服务器中分离出去。

分离就是将数据库从 SQL Server 实例中删除,使其数据文件和日志文件在逻辑上脱离服务器。经过分离后,数据库的数据文件和日志文件变成了操作系统中的文件,与服务器脱离,但保存了数据库的所有信息。当想备份数据库或移动到其他地方时,只要保存和转移这些数据文件和日志文件(两者缺一不可)即可。

1) 使用对象资源管理器

在“对象资源管理器”窗格中右击要分离的数据库,在弹出的快捷菜单中选择“任务”→“分离”命令,然后单击“确定”按钮,即可完成分离数据库的操作,如图 3.10 所示。

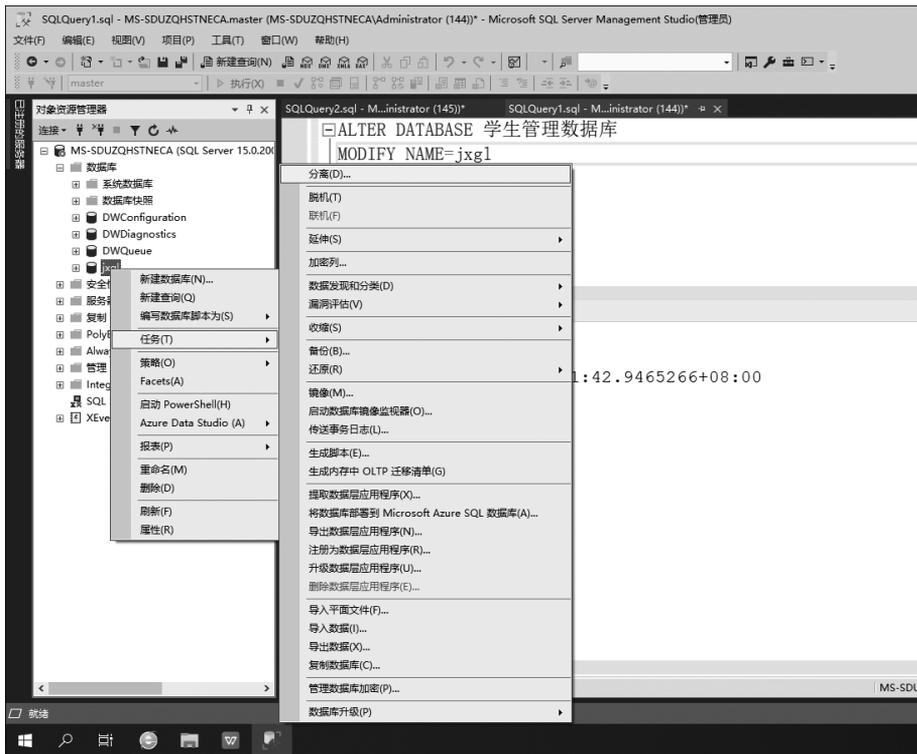


图 3.10 分离数据库

2) 使用 T-SQL 语句

其语法格式如下:

```
sp_detach_db database_name
```

2. 附加数据库

附加数据库的操作是分离数据库的逆操作,通过附加数据库,可以将没有加入 SQL Server 服务器的数据库文件加入服务器中。

1) 使用对象资源管理器

在“对象资源管理器”窗格中右击“数据库”选项,在弹出的快捷菜单中选择“附加”命令,弹出“附加数据库”对话框,如图 3.11 所示。单击“添加”按钮,弹出“定位数据库文件”对话框,如图 3.12 所示,选择分离数据库时的数据文件,然后单击“确定”按钮,即可完成附加数据库的操作。

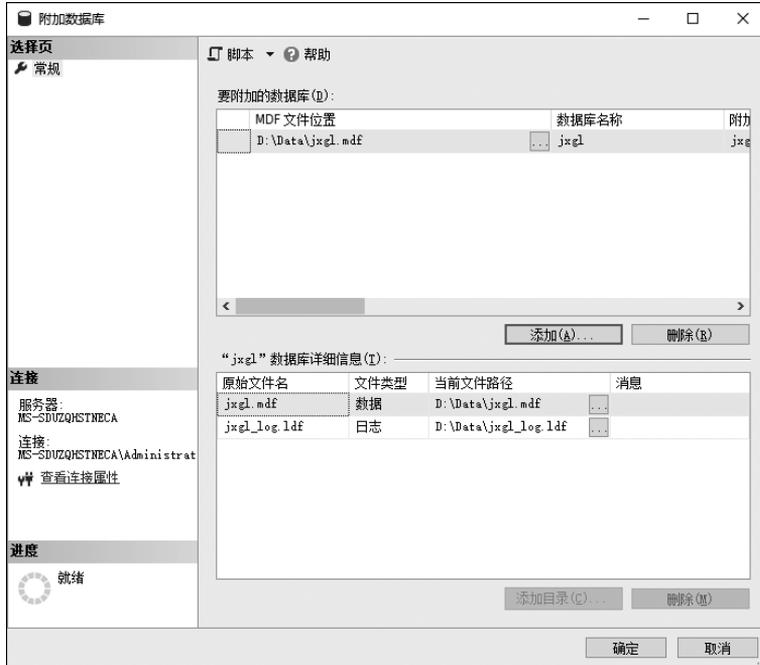


图 3.11 “附加数据库”对话框



图 3.12 “定位数据库文件”对话框

2) 使用 T-SQL 语句

一种方法是使用 CREATE DATABASE 中的 [FOR ATTACH] 选项, 另一种方法是使用系统存储过程, 语法格式如下:

```
sp_attach_db[@dbname=]'dbname',[@filename1=]'filename_n'[,...n]
```

- (1) [@dbname=]'dbname': 要附加的数据库名。
- (2) [@filename1=]'filename_n'[,...n]: 数据库文件的物理名称, 包括路径。

3.3.5 删除数据库

如果数据库不再需要了, 则应将其删除。用户只能根据自己的权限删除用户数据库, 不能删除当前正在使用(正打开供用户读写)的数据库, 更无法删除系统数据库(msdb、model、master、tempdb)。删除数据库意味着将删除数据库中所有的对象, 包括表、视图、索引等。如果数据库没有备份, 则不能恢复。

在 SQL Server 2019 中提供了两种删除方式: 一种方式是在 SQL Server Management Studio 中删除数据库, 另一种方式是通过 T-SQL 语句删除数据库。

1. 在 SQL Server Management Studio 平台删除数据库

在 SQL Server Management Studio 平台上右击要删除的数据库, 在弹出的快捷菜单中选择“删除”命令即可删除数据库。此时系统会弹出确认是否要删除数据库的窗口, 如图 3.13 所示, 单击“确定”按钮删除该数据库。

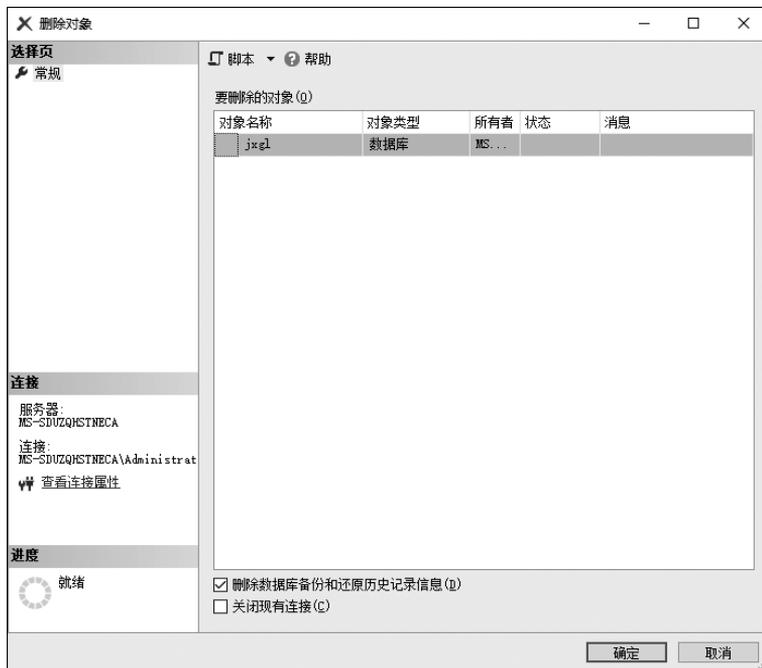


图 3.13 “删除对象”窗口

2. 用 T-SQL 语句删除数据库

T-SQL 中提供了数据库删除语句 DROP DATABASE,其语法格式如下:

```
DROP DATABASE database_name [, ...n]
```

【例 3.7】 删除已经创建的“学生管理数据库”。

```
DROP DATABASE 学生管理数据库
GO
```

3.4 实训项目：数据库基本操作

1. 实训目的

- (1) 掌握数据库的创建及使用方法。
- (2) 掌握管理数据库的方法。

2. 实训内容

1) 按要求创建数据库

(1) 使用 CREATE DATABASE 命令创建一个名为 stu01db 的数据库,包含一个主数据文件和一个事务日志文件。主数据文件的逻辑名为 stu01data,物理文件名为 stu01data.mdf,初始容量为 5MB,最大容量为 10MB,每次的增长量为 20%。事务日志文件的逻辑名为 stu01log,物理文件名为 stu01log.ldf,初始容量为 5MB,最大容量不受限制,每次的增长量为 2MB。这两个文件都放在当前服务器实例的默认数据库文件夹中。

(2) 使用 CREATE DATABASE 命令创建一个名为 stu02db 的数据库,包含一个主数据文件、两个次要数据文件和一个事务日志文件。主数据文件的逻辑名为 stu02data,物理文件名为 stu02data.mdf,初始容量为 5MB,最大容量为 10MB,每次的增长量为 20%。用户自定义文件组名为 stufgrp,建立两个文件,一个文件名为 stu02sf01,初始容量为 1MB,最大容量为 5MB,每次的增长量为 10%;另一个文件名为 stu02sf02,初始容量为 1MB,最大容量为 5MB,每次的增长量为 10%。事务日志文件的逻辑名为 stu02log,物理文件名为 stu02log.ldf,初始容量为 5MB,最大容量不受限制,每次的增长量为 2MB。

2) 查看数据库信息

3) 修改 stu01db 数据库

(1) 修改 stu01db 数据库,增加一个次要数据文件,并且将这两个次要数据文件划归到新的文件组 stufgrp 中。次要数据文件的逻辑名为 stu01sf01,初始容量为 1MB,最大容量为 5MB,按 10%增长。

(2) 将 stu01db 数据库的数据文件 stu01sf01 的初始空间和最大空间分别由原来的 1MB 和 5MB 修改为 2MB 和 6MB。

4) 创建 marketing 数据库

请创建一个 marketing 数据库,并将其存放到 E 盘用自己的名字命名的文件夹中。