

第5章 数据库的创建与管理

本章主要介绍 SQL Server 中数据库的概念、数据库文件的类型、系统数据库,以及创建、打开、修改和删除数据库的方法。

5.1 SQL Server 数据库概述

5.1.1 数据库引擎

Microsoft SQL Server Database Engine(数据库引擎)是用于存储、处理和保护数据的核心服务。利用数据库引擎可控制访问权限并快速处理事务,从而满足企业内要求极高且需要处理大量数据的应用需要。

使用数据库引擎创建用于联机事务处理或联机分析处理数据的关系数据库,包括创建用于存储数据的表以及用于查看、管理和保护数据安全的数据对象(如索引、视图和存储过程)。

数据库引擎的主要任务如下。

- ① 设计并创建数据库,以保存系统所需的关系表或 XML 文档。
- ② 实现系统访问和更改数据库中存储的数据,包括实现网站或使用数据的应用程序,以及生成利用 SQL Server 工具和实用工具使用数据的过程。
- ③ 为单位或客户部署将要实现的系统。
- ④ 提供日常管理支持,以优化数据库的性能。

SQL Server 中的数据库由表集合组成。这些表包含数据以及为支持数据操作而定义的其他对象,如视图、索引、存储过程、用户定义函数和触发器。存储在数据库中的数据通常与特定的主题或过程相关,如生产仓库的库存信息。表上有几种类型的控制(如约束、触发器、默认值和自定义用户数据类型),用于保证数据的有效性。可以向表上添加完整性约束,以确保不同表中的相关数据保持一致。表上可以有索引,利用索引能够快速找到行。

数据库还可以包含使用 Transact-SQL 或 .NET Framework 编程代码的过程对数据库中数据执行操作。这些操作包括创建用于提供对表数据的自定义访问的视图,或创建用于对部分行执行复杂计算的用户定义函数。

一个 SQL Server 实例可以支持多个数据库。每个数据库可以存储来自其他数据库的相关数据或不相关数据。例如,SQL Server 实例可以有一个数据库用于存储职员数据,另一个数据库用于存储与产品相关的数据。或者一个数据库可以存储当前客户订单数据,而另一个相关数据库可以存储用于年度报告的历史客户订单。

5.1.2 文件和文件组

SQL Server 将数据库映射为一组操作系统文件。数据和日志信息保存在不同的文件中,

而且每个文件仅在一个数据库中使用。为了便于分配和管理,可以将数据文件集合起来放到文件组中,用于帮助完成数据布局和管理任务,例如备份和还原操作。

每个 SQL Server 数据库至少具有两个操作系统文件:一个数据文件和一个日志文件。数据文件包含数据和对象,例如表、索引、存储过程和视图;日志文件包含恢复数据库中的所有事务所需的信息。

1. 数据库文件

SQL Server 数据库具有如下三种类型的文件。

(1) 主数据文件

主数据文件包含数据库的启动信息,并指向数据库中的其他文件。用户数据和对象可存储在此文件中,也可以存储在次要数据文件中。每个数据库都有一个主数据文件。主数据文件的建议文件扩展名是.mdf。

(2) 辅助数据文件

除主数据文件以外的所有其他数据文件都是辅助数据文件,又称次要数据文件。辅助数据文件是可选的,由用户定义并用来存储用户数据。通过将每个文件放在不同的磁盘驱动器上,辅助数据文件可用于将数据分散到多个磁盘上。辅助数据文件可以有 0 到多个。如果数据库超过了单个 Windows 文件的最大容量,可以使用辅助数据文件,这样数据库就能继续增长。辅助数据文件的建议文件扩展名是.ndf。

(3) 事务日志文件

事务日志文件保存用于恢复数据库的日志信息。每个数据库必须至少有一个日志文件,也可以有多个日志文件。事务日志文件的建议扩展名是.ldf。

在 SQL Server 中,数据库中所有文件的位置都记录在数据库的主文件和 master 数据库中。

默认情况下,数据和事务日志被放在同一个驱动器上的同一个路径下。这是为处理单磁盘系统而采用的方法。但是,在生产环境中,这可能不是最佳的方法。建议将数据和日志文件放在不同的磁盘上。

2. 数据库逻辑和物理文件名称

SQL Server 数据库文件有两个名称。

(1) 逻辑文件名

逻辑文件名(logical_file_name)是在所有 Transact-SQL 语句中引用物理文件时所使用的名称。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中必须是唯一的。它是由 ALTER DATABASE 的 NAME 参数设置的。

(2) 物理文件名

物理文件名(os_file_name)是包括目录路径的物理文件名。它必须符合操作系统文件命名规则。它是由 ALTER DATABASE 的 FILENAME 参数设置的。

在默认 SQL Server 实例上创建的数据库的逻辑文件名和物理文件名示例如图 5.1 所示,该数据库有一个主数据库文件、两个辅助数据库文件和两个日志文件。

SQL Server 数据库文件的逻辑文件名与物理文件名的主文件名可以一致,也可以不一致,在图 5.1 中上面列出的是数据库的逻辑文件名,下面列出的是物理文件名。

3. 数据文件页

SQL Server 中数据存储的基本单位是页。为数据库中的数据文件(.mdf 或.ndf)分配的

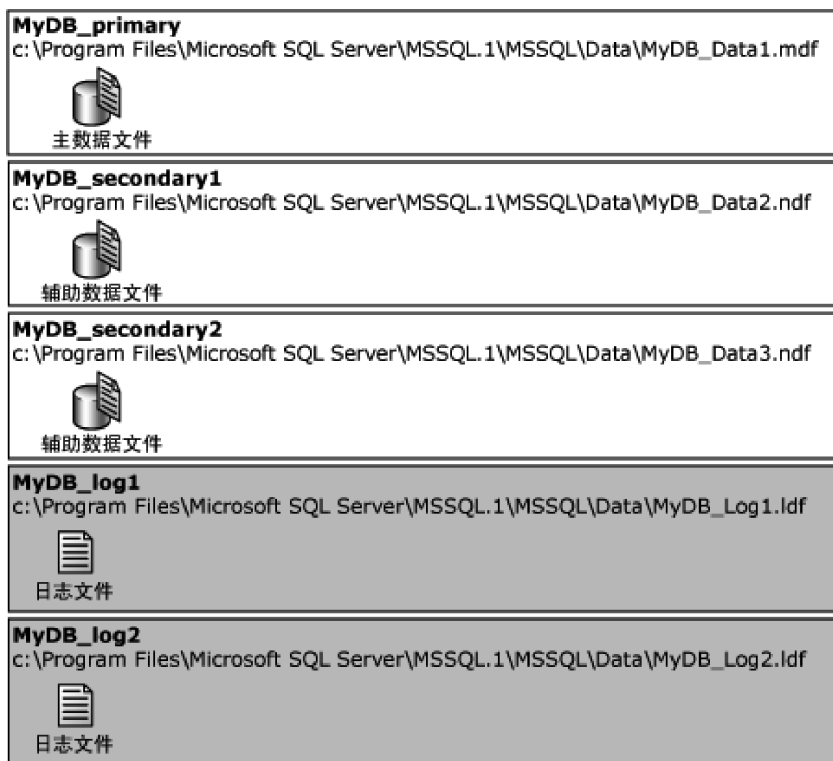


图 5.1 数据库逻辑文件名与物理文件名示例

磁盘空间可以从逻辑上划分成页(从 0 到 n 连续编号)。磁盘 I/O 操作在页级执行。也就是说,SQL Server 是以页为单位读写数据的。

区是 8 个物理上连续的页的集合,用来有效地管理页。所有页都存储在区中。

(1) 页

在 SQL Server 中,页的大小为 8KB。这意味着 SQL Server 数据库中每 MB 有 128 页。每页的开头是 96B 的标头,用于存储有关页的系统信息。此信息包括页码、页类型、页的可用空间以及拥有该页的对象的分配单元 ID。

注意: 日志文件不包含页,而是包含一系列日志记录。

(2) 区

区是管理空间的基本单位。一个区是 8 个物理上连续的页(即 64KB)。这意味着 SQL Server 数据库中每 MB 有 16 个区。

为了使空间分配更有效,SQL Server 不会将所有区分配给包含少量数据的表。SQL Server 有两种类型的区:

- ① 统一区。由单个对象所有。区中的所有 8 页只能由所属对象使用。
- ② 混合区。最多可由 8 个对象共享。区中 8 页的每页可由不同的对象所有。

(3) 文件页

SQL Server 数据文件中的页按顺序编号,文件的首页以 0 开始。数据库中的每个文件都有唯一的文件 ID 号。若要唯一标识数据库中的页,需要同时使用文件 ID 和页码。图 5.2 显示了包含 4MB 主数据文件和 1MB 辅助数据文件的数据库中的页码。

4. 文件大小

SQL Server 文件可以从最初指定的大小开始自动增长。在定义文件时,可以指定一个特

定的增量。每次填充文件时,其大小均按此增量来增长。如果文件组中有多个文件,则它们在所有文件被填满之前不会自动增长。填满后,这些文件会循环增长。

每个文件还可以指定一个最大容量。如果没有指定最大容量,文件可以一直增长到用完磁盘上的所有可用空间。

5. 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。有以下两种类型的文件组。

(1) 主要文件组

每个数据库有一个主要文件组(PRIMARY 文件组)。此文件组包含主数据文件和未放入其他文件组的所有次要文件。所有系统表都被分配到主要文件组中。

(2) 用户定义文件组

用户定义文件组是用户首次创建数据库或以后修改数据库时明确创建的任何文件组,用于将数据文件集合起来,以便于管理、分配和放置数据。用户定义文件组是通过在 CREATE DATABASE 或 ALTER DATABASE 语句中使用 FILEGROUP 关键字指定的任何文件组。

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。例如,分别在三个磁盘驱动器上创建三个文件 Data1.ndf、Data2.ndf 和 Data3.ndf,将它们分配给文件组 fgroup1。然后在文件组 fgroup1 上创建一个表。对表中数据的查询将分散到三个磁盘上,从而提高了性能。

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,或者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中的单独文件组中。

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。不管何时,只能将一个文件组指定为默认文件组。默认文件组中的文件必须足够大,能够容纳未分配给其他文件组的所有新对象。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 PRIMARY 文件组作为默认文件组。

PRIMARY 文件组是默认文件组,除非使用 ALTER DATABASE 语句进行了更改。但系统对象和表仍然分配给 PRIMARY 文件组,而不是新的默认文件组。

5.1.3 事务日志

每个 SQL Server 数据库都有一个事务日志,用于记录所有事务,以及每个事务对数据库所做的修改。事务日志是任何数据库的关键组成部分。如果系统出现故障,它将成为最新数据的唯一源,因此慎做删除或移动事务日志操作。

1. 事务日志支持的操作

事务日志支持以下操作。

(1) 恢复个别的事务

如果应用程序发出 ROLLBACK(事务回滚)语句,或者数据库引擎检测到错误,系统使用日志记录回滚未完成事务所做的修改。

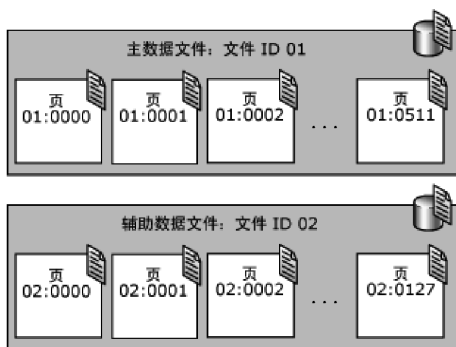


图 5.2 数据文件页示例

(2) SQL Server 启动时恢复所有未完成的事务

当运行 SQL Server 的服务器发生故障时,数据库可能处于以下不正确的状态:还没有将某些修改从缓存写入数据文件,在数据文件内有未完成事务所做的修改。当启动 SQL Server 实例时,它对每个数据库执行恢复操作。前滚日志中记录的、可能尚未写入数据文件的每个修改,在事务日志中找到的每个未完成的事务都将回滚,以确保数据库的完整性。

(3) 将还原的数据库、文件、文件组或页前滚到故障点

在硬件丢失或磁盘故障影响到数据库文件后,可以将数据库还原到故障点。首先还原上一次的完整备份和差异备份,然后将事务日志备份后续序列还原到故障点。当还原每个日志备份时,数据库引擎重新应用日志中记录的所有修改,以前滚(Roll Forward)所有事务。当最后的日志备份还原后,数据库引擎将使用日志信息回滚到该点未完成的所有事务。

(4) 支持事务复制

日志读取器代理程序监视已为事务复制配置的每个数据库的事务日志,并将已设复制标记的事务从事务日志复制到分发数据库中。

(5) 支持备用服务器解决方案

备用服务器解决方案、数据库镜像和日志传送高度依赖于事务日志。在日志传送方案中,主服务器将主数据库的活动事务日志发送到一个或多个目标服务器。每个辅助服务器将该日志还原为其本地的辅助数据库。

在数据库镜像方案中,数据库(主体数据库)的每次更新都在独立的、完整的数据库(镜像数据库)副本中立即重新生成。主体服务器实例立即将每条日志记录发送到镜像服务器实例,镜像服务器实例将传入的日志记录应用于镜像数据库,从而将其继续前滚。

2. 事务日志的特征

① 事务日志是作为数据库中的单独的文件或一组文件实现的。日志缓存与数据页缓存分开管理,从而使数据库引擎内的编码更简单、更快速和更可靠。

② 日志记录和页的格式不必遵守数据页的格式。

③ 事务日志可以在几个文件上实现。通过设置日志的 FILEGROWTH 值,可以将这些文件定义为自动扩展。这样可减少事务日志内空间不足的可能性,同时减少管理开销。

④ 重用日志文件中空间机制,速度快且对事务吞吐量影响最小。

5.1.4 数据库快照

数据库快照是 SQL Server 数据库(源数据库)的只读静态视图。创建快照时,每个数据库快照在事务上与源数据库一致。多个快照可以位于一个源数据库中,并且可作为数据库始终驻留在同一服务器实例上。当源数据库更新时,数据库快照也将更新。因此数据库快照存在的时间越长,就越有可能用完其可用磁盘空间。在被数据库所有者显式删除前,快照始终存在。

快照可用于报表。另外,如果源数据库出现用户错误,还可将源数据库恢复到创建快照时的状态。丢失的数据仅限于创建快照后数据库更新的数据。

5.2 系统数据库

SQL Server 2022 安装成功后,系统自动安装了 4 个系统数据库和 2 个示例数据库,下面分别介绍。

1. master 数据库

master 数据库记录 SQL Server 系统的所有系统级信息,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。master 数据库还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。另外,master 也记录 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。在 SQL Server 中,系统对象不再存储在 master 数据库中,而是存储在 Resource 数据库中。

表 5.1 列出了 master 数据库的数据和日志文件的初始配置值。对于 SQL Server 的不同版本,这些文件的大小可能略有不同。

表 5.1 master 数据库的初始配置

文 件	逻辑名称	物理名称	文 件 增 长
主数据文件	master	master.mdf	按 10%自动增长,直到磁盘已满
日志文件	mastlog	mastlog.ldf	按 10%自动增长,直到达到最大值 2TB

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 的所有者是 sa。
- 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 启用变更数据捕获。
- ⑨ 参与数据库镜像。
- ⑩ 删除主文件组、主数据文件或日志文件。
- ⑪ 重命名数据库或主文件组。
- ⑫ 将数据库设置为 OFFLINE。
- ⑬ 将数据库或主文件组设置为 READ_ONLY。

使用 master 数据库时的建议：

- ① 始终有一个 master 数据库的当前备份可用。
- ② 执行下列操作后,尽快备份 master 数据库：创建、修改或删除数据库；更改服务器或数据库的配置值；修改或添加登录账户。
- ③ 不要在 master 中创建用户对象。否则,必须更频繁地备份 master。
- ④ 不要针对 master 数据库将 TRUSTWORTHY 选项设置为 ON。

2. model 数据库

model 数据库是在 SQL Server 实例上创建的所有数据库的模板。因为每次启动 SQL Server 时都会创建 tempdb 数据库,所以 model 数据库必须始终存在于 SQL Server 系统中。

当发出 CREATE DATABASE 语句时,将通过复制 model 数据库中的内容来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。

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。

表 5.2 列出了 model 数据库的数据和日志文件的初始配置值。对于 SQL Server 的不同版本,这些文件的大小可能略有不同。

表 5.2 model 数据库的初始配置

文 件	逻辑名称	物理名称	文 件 增 长
主数据文件	modeldev	model.mdf	按 10% 自动增长,直到磁盘充满为止
日志文件	modellog	modellog.ldf	按 10% 自动增长,直到达到最大值 2TB

不能在 model 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。model 的所有者是 sa。
- ④ 删除数据库。
- ⑤ 从数据库中删除 guest 用户。
- ⑥ 启用变更数据捕获。
- ⑦ 参与数据库镜像。
- ⑧ 删除主文件组、主数据文件或日志文件。
- ⑨ 重命名数据库或主文件组。
- ⑩ 将数据库设置为 OFFLINE。
- ⑪ 将数据库或主文件组设置为 READ_ONLY。

⑫ 使用 WITH ENCRYPTION 选项创建过程、视图或触发器。加密密钥与在其中创建对象的数据库绑定在一起。在 model 数据库中创建的加密对象只能用于 model 中。

3. msdb 数据库

msdb 数据库由 SQL Server 代理用来计划警报和作业。

表 5.3 列出了 msdb 数据库的数据和日志文件的初始配置值。对于 SQL Server 的不同版本,这些文件的大小可能略有不同。

表 5.3 msdb 数据库的初始配置

msdb 文件	逻辑名称	物理名称	文 件 增 长
主数据	MSDBData	MSDBData.mdf	按 256KB 自动增长,直到磁盘已满
Log	MSDBLog	MSDBLog.ldf	按 256KB 自动增长,直到达到最大值 2TB

不能在 msdb 数据库中执行下列操作。

- ① 更改排序规则。默认排序规则为服务器排序规则。
- ② 删除数据库。
- ③ 从数据库中删除 guest 用户。
- ④ 启用变更数据捕获。
- ⑤ 参与数据库镜像。
- ⑥ 删除主文件组、主数据文件或日志文件。
- ⑦ 重命名数据库或主文件组。
- ⑧ 将数据库设置为 OFFLINE。
- ⑨ 将主文件组设置为 READ_ONLY。

4. tempdb 数据库

tempdb 系统数据库是连接到 SQL Server 实例的所有用户都可用的全局资源,它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 生成的工作表。

每次启动 SQL Server 时,都要重新创建 tempdb,以便在系统启动时,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。

Tempdb 用于保存以下内容。

- ① 显式创建的临时对象,例如表、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)。

tempdb 中的操作是最小日志记录操作,这样可以回滚事务,但 tempdb 不能备份或还原。

表 5.4 列出了 tempdb 数据库的数据和日志文件的初始配置值。对于 SQL Server 的不同版本,这些文件的大小可能略有不同。

表 5.4 tempdb 数据库的初始配置

文件	逻辑名称	物理名称	文件增长
主数据文件	tempdev	tempdb.mdf	按 10%自动增长,直到磁盘已满
日志文件	templog	templog.ldf	按 10%自动增长,直到达到最大值 2TB

tempdb 的大小可以影响系统性能。如果 tempdb 的尺寸太小,则每次启动 SQL Server 时,系统可能忙于处理数据库的自动增长,而不能满足工作负荷要求。可以通过增加 tempdb 的尺寸来避免此开销。

不能对 tempdb 数据库执行以下操作。

- ① 添加文件组。
- ② 备份或还原数据库。
- ③ 更改排序规则。默认排序规则为服务器排序规则。
- ④ 更改数据库所有者。tempdb 的所有者是 sa。
- ⑤ 删除数据库。
- ⑥ 从数据库中删除 guest 用户。
- ⑦ 启用变更数据捕获。
- 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 运行 DBCC CHECKALLOC。
- ⑫ 运行 DBCC CHECKCATALOG。
- ⑬ 将数据库设置为 OFFLINE。
- ⑭ 将数据库或主文件组设置为 READ_ONLY。

5.3 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。其中,所有者是创建数据库的用户。

在创建数据库之前,用户必须至少拥有 CREATE DATABASE、CREATE ANY DATABASE 或 ALTER ANY DATABASE 权限(第 12 章介绍)。

一个 SQL Server 实例最多可以创建 32 767 个数据库。数据库名称必须遵循为标识符指

定的规则。

model 数据库中的所有用户定义对象都将复制到所有新创建的数据库中。可以向 model 数据库中添加任何对象(如表、视图、存储过程和数据类型),以将这些对象包含到所有新创建的数据库中。

在 SQL Server 中,可以通过 SQL Server Management Studio 的图形工具或 Transact-SQL 语句创建数据库,下面将分别介绍。

数据库案例背景:

大学生作为思想活跃的群体,如何有效地加强其思想政治教育,是广大教育工作者面临的重要课题。红色影视作品作为思想政治教育的天然“教科书”,其蕴含的爱国主义精神、民族精神和革命精神,为高校思政教育提供了丰富的素材和案例。通过深入挖掘红色影视资源的理论内涵和时代价值,可以激励大学生立大志、明大德、成大才、担大任,努力成为堪当民族复兴重任的时代新人。

5.3.1 使用图形工具创建数据库

【例 5.1】 创建一个红色影视作品数据库, 名称为 RedMovie, 该数据库贯穿本书的后续章节。

在 SQL Server Management Studio 中创建红色影视作品数据库过程如下。

- ① 启动 Microsoft SQL Server Management Studio。
- ② 在“对象资源管理器”的树状结构中右击“数据库”,在出现的快捷菜单中选择“新建数据库”菜单项,如图 5.3 所示。

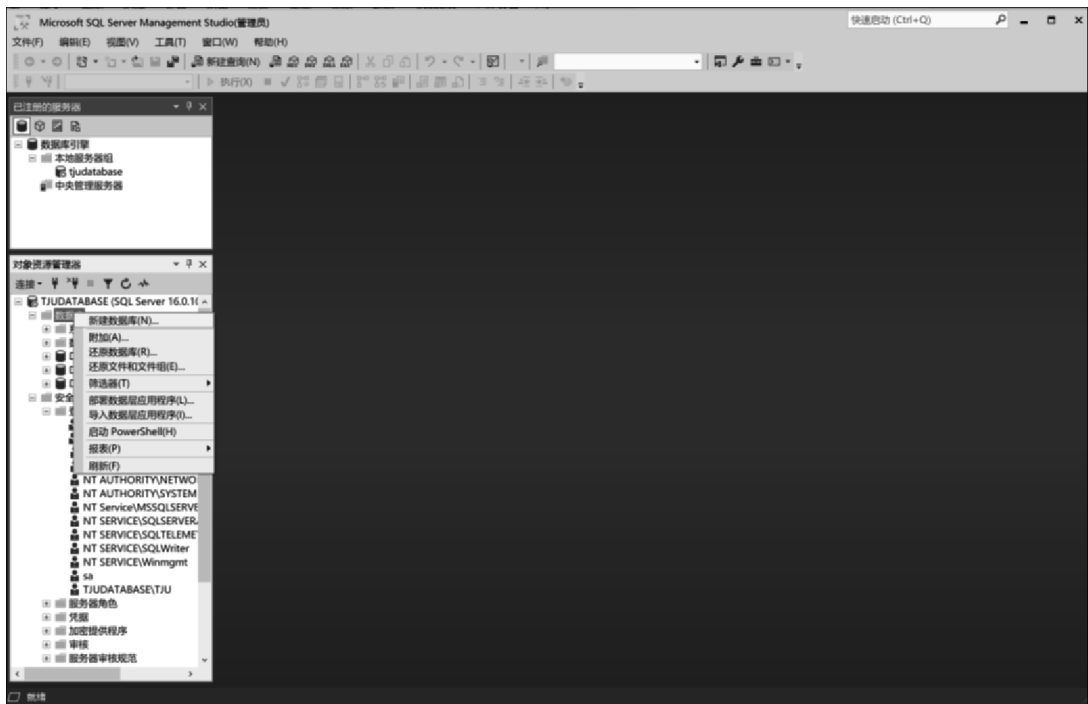


图 5.3 “新建数据库”菜单项

③ 出现“新建数据库”窗口,如图 5.4 所示。

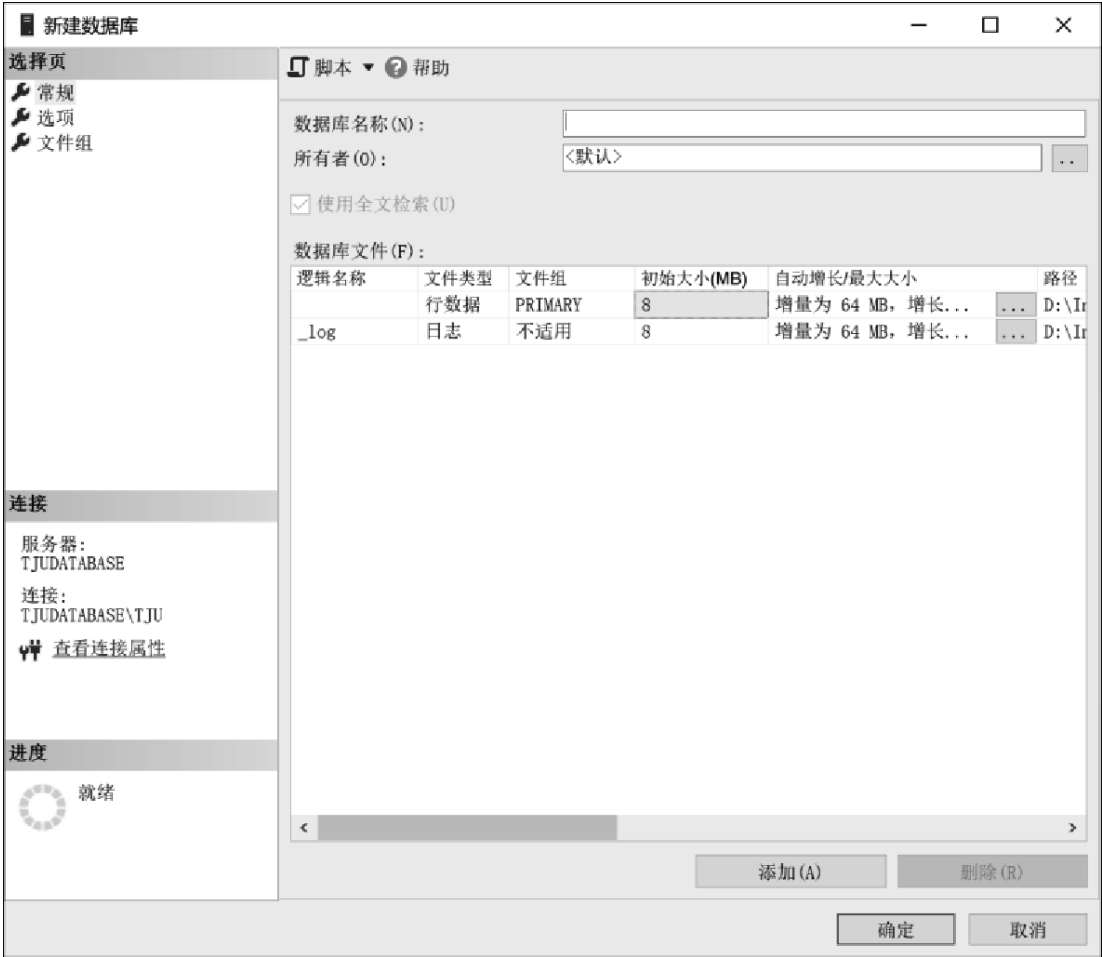


图 5.4 “新建数据库”窗口

在图 5.4 所示的“常规”选项页中“数据库名称”处输入“RedMovie”,在“逻辑名称”下输入主数据库文件的逻辑名称为“RedMovie _data”,在“初始大小”下可以设置主数据库文件的大小,单击“自动增长”后的【...】按钮,出现图 5.5 所示的“更改自动增长”对话框。



图 5.5 “更改自动增长”对话框