

## 8.4 SQL Server 的管理工具

常用的 SQL Server 的管理工具包括管理控制台（SQL Server Management Studio）、配置管理器（SQL Server Configuration Manager）和数据库引擎优化顾问（Database Engine Tuning Advisor）等。

### 8.4.1 SQL Server Configuration Manager

#### 1. SQL Server Configuration Manager

SQL Server 2005 以后的配置管理器综合了 SQL Server 2000 中的服务管理器、服务器网络实用工具和客户端使用工具的功能。配置管理器是一种工具，用于管理与 SQL Server 相关联的服务、配置 SQL Server 使用的网络协议以及从 SQL Server 客户端的网络连接配置。

SQL Server 配置管理器是一种可以通过“开始”菜单访问的 Microsoft 管理控制台管理单元，也可以将其添加到任何其他 Microsoft 管理控制台的显示界面中。使用 SQL Server 配置管理器可以启动、暂停、恢复或停止服务，还可以查看或更改服务属性。

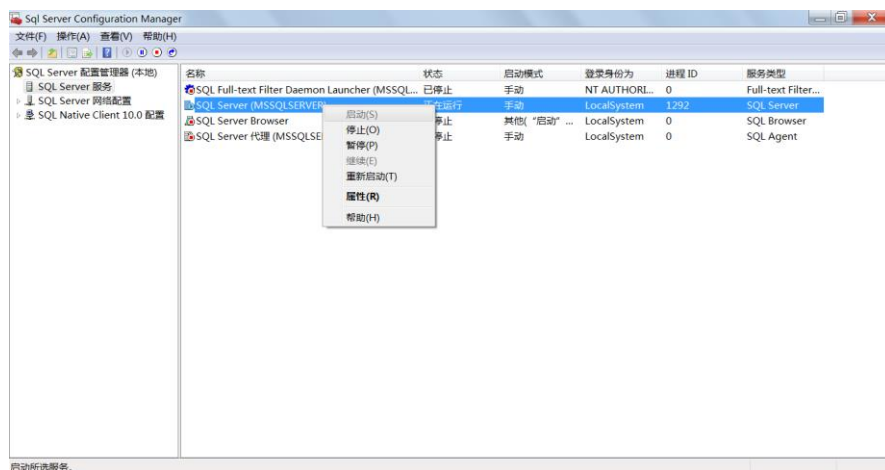


图 8-1 SQL Server Configuration Manager 主界面

启动 SQL Server 配置管理器的方法：从“开始”菜单中选择“所有程序”→Microsoft SQL Server→“配置工具”→SQL Server 配置管理器命令启动 SQL Server 配置管理器，如图 8-1 所示。

配置管理器可以对 SQL Server 服务、网络、协议等进行配置，配置好后客户端才能顺利地连接和使用 SQL Server。

#### 2. 实例的概念

在介绍 SQL Server 服务之前，我们首先需要理解一个概念——实例(Instance)。各个数据库厂商对实例的解释不完全一样。在 SQL Server 中这样理解实例：当在一台计算机上安装一次 SQL Server 时，就形成了一个实例。

实例标志一组 SQL Server 服务。安装 SQL Server 服务器组件，就是创建一个新的 SQL Server 实例，SQL Server 允许在同一个操作系统中创建多个实例。如果是在计算机上第一次安装 SQL Server，则 SQL Server 安装向导会提示用户选择把这次安装的 SQL Server 作为默认实例还是命名实例（默认选项是默认实例）。一台计算机只能有一个默认实例，用当前计算机的网络名作为其实例名。

一台计算机除了安装 SQL Server 的默认实例外，还可以安装多个命名实例。命名实例在安装的时候为实例指定了一个实例名，然后就可以用该名称来访问该实例，访问的方法是：在客户端工具中输入“计算机名/命名实例名”。

### 3. 启动 SQL Server 服务

SQL Server 的每个实例都提供了一组服务，包括数据库引擎、分析服务、报表服务及集成服务等。其中数据库引擎是核心服务，一般情况下，要完成 SQL Server 的基本操作，比如创建数据库、表等都必须启动该服务。

以下介绍如何启动 SQL Server 服务。

单击图 8-1 所示窗口左边的 SQL Server 服务节点，在窗口的右边会列出已安装的 SQL Server 服务，包括不同实例（如果有的话）的服务。其中 SQL Server 服务是 SQL Server 数据库的核心服务，也就是数据库引擎，SQL Server 的其它服务都是围绕这个服务进行。只有启动了这个服务，SQL Server 数据库管理系统才能发挥作用，用户也才能建立与服务器的连接。

在图 8-1 中，用户可以右击 SQL Server(MSSQLSERVER)服务（MSSQLSERVER 代表实例名），在弹出的下拉菜单中执行“启动”、“暂停”和“停止”命令即可实现该服务的启动、暂停和停止操作。

其中，暂停和停止服务操作的含义解释如下：暂停服务是指拒绝新的客户机连接请求，但是已经建立的客户机连接不受影响，可以继续执行。停止服务则从内存中清除所有有关的 SQL Server 服务器的进程，除了不允许新的用户继续登录服务器外，已经建立的连接也会立即发生中止。在 DBA 的实际管理中，一般会先选择暂停，在确认没有客户机连接后选择关闭服务。

在如图 8-1 所示的下拉菜单中选择“属性”命令，可以弹出如图 8-2 所示的“属性”窗口。在“登录”选项卡中可以设置启动服务的账户，在“服务”选项卡中可以设置服务的启动方式（如图 8-3 所示）：自动、手动和已禁用。

- 1) 自动：表示每当操作系统启动时自动启动该服务；
- 2) 手动：表示每次使用该服务时都需要用户手工启动；
- 3) 已禁用：表示禁止该服务的启动。



图 8-2 SQL Server 服务的“属性”窗口

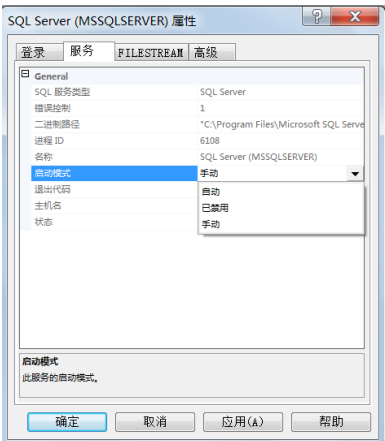


图 8-3 设置服务的启动方式

## 8.4.2 SQL Server Management Studio

SQL Server Management Studio（简称为 SSMS）是 SQL Server 中最重要的管理工具，它

融合了 SQL Server 2000 的查询分析器和企业管理器、OLAP 分析器等多种工具的功能，为管理人员提供了一个单一的集成环境。此集成环境使用户可以在一个界面内执行各种任务，例如，备份数据、编辑查询和执行常见函数等，还可以和 SQL Server 的所有组件协同工作，例如 Reporting Services、Integration Services 等。

### 1. 启动 SQL Server Management Studio

注意： 在启动 SSMS 前，要先开启 SQL Server 服务。

从“开始”菜单中选择“所有程序”→Microsoft SQL Server →SQL Server Management Studio 命令启动 SSMS。首先弹出的是“连接到服务器”对话框，如图 8-4 所示。

在该窗口中，各选项含义如下：

- 1) 服务器类型：该下拉框列出了 SQL Server 服务器所包含的服务，可见图 8-5。当前连接的是数据库引擎，即 SQL Server 服务。
- 2) 服务器名称：指定要连接的数据库服务器的实例名。SSMS 能够自动扫描当前网络中的 SQL Server 实例。这里连接的是默认实例，其实例名就是计算机名。



图 8-4 “连接到服务器”窗口

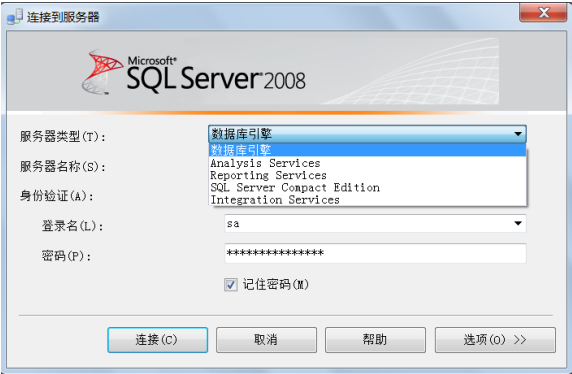


图 8-5 服务器类型

- 3) 身份验证：选择用哪种身份连接到数据库服务器。有两种选择：Windows 身份验证和 SQL Server 身份验证。

**Windows 身份验证：**指使用操作系统的用户账户和密码连接数据库服务器。如果选择的是该选项，那么不用输入用户名和密码，SQL Server 会选用当前登录到 Windows 的用户作为其连接用户，如图 8-4 所示。

**SQL Server 身份验证：**窗口形式如图 8-6 所示，这是需要输入 SQL Server 身份验证的登录名和相应的密码。该模式要求该数据库服务器的身份验证模式必须是“混合身份验证”模式。该模式的验证方式是首先用 SQL Server 的用户和密码验证，若是有效的登录名和正确的密码，则接收该用户的连接；否则，请求 Windows 操作系统进行验证。身份验证模式可以在安装的时候指定，也可以在安装以后在 SSMS 工具中进行修改。具体介绍见 14.1 节。

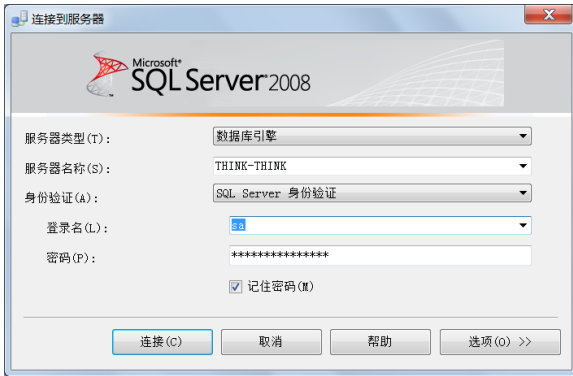


图 8-6 SQL Server 身份验证

按以上步骤连接成功后，即可进入 SSMS 操作界面，如图 8-7 所示。

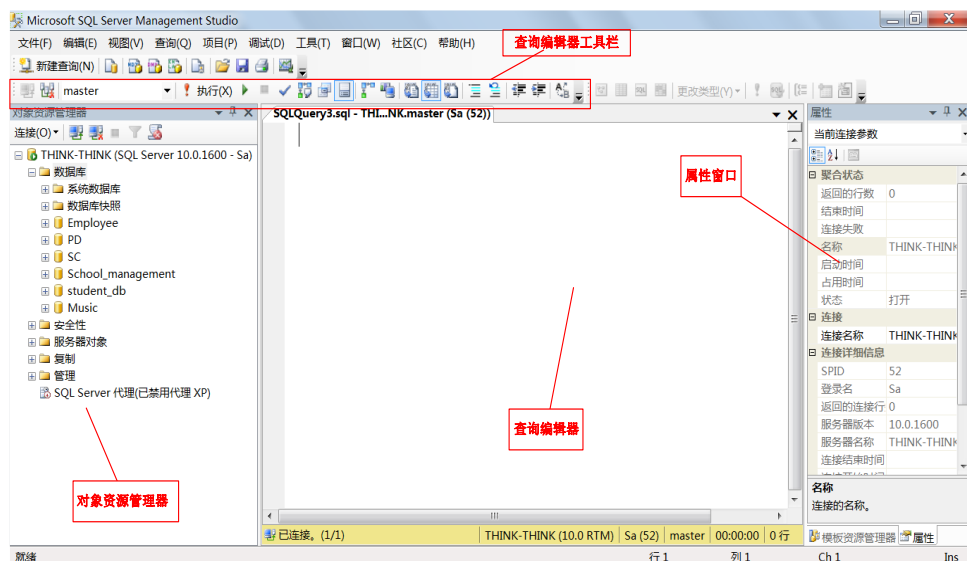


图 8-7 SQL Server Management Studio 主界面

SQL Server Management Studio 包含以下工具：

#### 1) 对象资源管理器

可以通过“视图”菜单访问对象资源管理器。该组件使用了类似于“Windows 资源管理器”的树结构，根节点是当前实例，子节点是该服务器的所有管理对象和可以执行的管理任务，分为“数据库”、“安全性”、“服务器对象”、“复制”、“管理”和“SQL Server 代理”等。

#### 2) 解决方案资源管理器

用于将相关脚本组织并存储为项目的一部分。可以通过“视图”菜单访问该工具。

#### 3) 属性窗口

用于显示当前选定对象的属性。

#### 4) 查询编辑器

查询编辑器取代了 SQL Server 早期版本中包含的查询分析器，可用于编写和编辑脚本，是一种功能丰富的脚本编辑器。在 SSMS 的“标准”工具栏上点击“新建查询”即可打开查询编辑器。此时，在工具栏上会增加一个“查询编辑器”工具条（见图 8-7）。因为一个实例可有拥有多个数据库，可以根据需要在“查询编辑器”工具条的“可用数据库”下拉列表选择当前要操作的数据库。

#### 5) 已注册的服务器：

可以通过“视图”菜单访问该工具。如图 8-8 所示，该组件存储注册服务器数据库引擎的名称信息。用户可以通过该组件设置数据库引擎，包括启动、停止服务器和对服务器属性进行设置等，并且可以导入、导出其它服务器，用户能够通过“已注册的服务器”组件新建一个服务器，还可以注册到网络中其它 SQL Server 服务器等。

#### 6) 模板资源管理器

可以通过“视图”菜单访问该工具。该组件提供多种模版，可以在查询编辑器中快速构造代码。

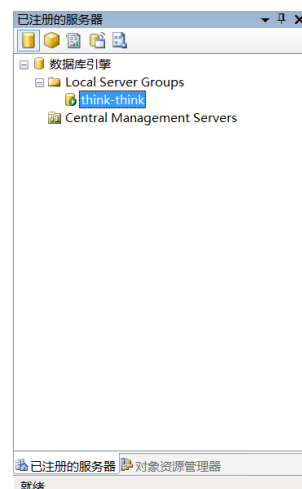


图 8-8 已注册的服务器

## 第 10 章 管理数据库

SQL Server 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。用户可以通过创建数据库来存储不同类别或者形式的数据。

本章的学习要点：

- \* 数据库对象及组成
- \* 数据库的创建
- \* 数据库的管理和维护

### 10.1 数据库的组成

SQL Server 中的数据库由包含数据的表集合和其它对象（如视图、索引等）组成，目的在于为执行与数据有关的活动提供支持。SQL Server 支持在一个数据库实例中创建多个数据库，每个数据库在物理和逻辑上都是独立的。

SQL Server 数据库的概念可以从以下不同的角度描述：从模式层次角度看，可以分别描述为物理数据库和逻辑数据库；从创建对象角度看，可以分为系统数据库和用户数据库。

---

说明：实际上，系统数据库和用户数据库都是基于逻辑数据库的概念。

---

在安装完 SQL Server 2008 后，在 SQL Server Management Studio 集成工作环境中的“对象资源管理器”窗口中依次展开“数据库”节点、“系统数据库”子节点和“master”数据库节点，该数据库对象由“表”、“视图”、“同义词”、“可编程性”等若干个对象组成，这就是 SQL Server 数据库的逻辑组成（图 10-1）。在物理存储上，该“master”数据库被映射成两个操作系统文件，文件名分别为“master.mdf”和“mastlog.ldf”，其存储路径为“Microsoft SQL Server \MSSQL10. MSSQLSERVER\MSSQL\DATA”。

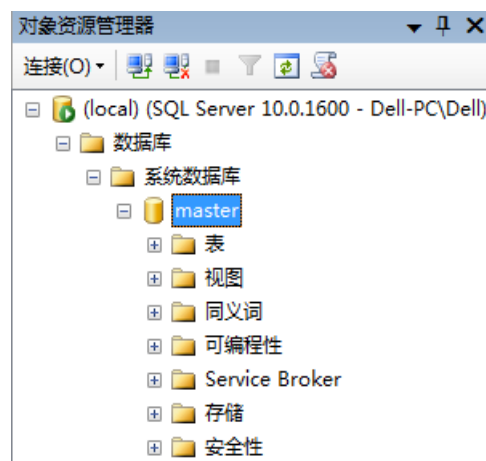


图 10-1 master 数据库的逻辑组成

以下首先我们来具体描述物理数据库和逻辑数据库。

#### 10.1.1 物理数据库与文件

物理数据库是从数据库的物理角度描述数据库，它将数据库映射到一组操作系统文件上，即物理数据库是构成数据库的物理文件（操作系统文件）的集合。

##### 1. 数据库文件

每个 SQL Server 数据库有数据文件和事务日志文件。它们是数据库系统真实存在的物理文件基础，而逻辑数据库是建立在该基础上的关于数据库的逻辑结构的抽象。数据文件包含数据和对象，例如表、索引、存储过程和视图；事务日志文件包含恢复数据库中的所有事

务所需的信息。数据文件又可分为主数据文件和辅助数据文件。

1) 主数据文件 (Primary): 简称为主文件。是数据库的关键文件, 包含数据库的启动信息, 并指向数据库中的其他文件, 并可存储部分或全部数据。每个数据库有且只有一个主数据文件, 默认文件扩展名是 .mdf。

2) 辅助数据文件 (Secondary): 简称为辅助文件, 又称为次要数据文件, 用于存储未包含在主数据文件内的其他数据。辅助数据文件是可选的, 一个数据库可有 0 个或多个辅助数据文件。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。另外, 如果数据库文件超过了单个 Windows 文件的最大大小, 可以使用辅助数据文件, 这样数据库就能继续增长。辅助数据文件的默认文件扩展名是 .ndf。

3) 事务日志文件 (Transaction Log): 简称日志文件。

事务是工作单元, 该单元的工作要么全部完成, 要么全部不完成。SQL Server 系统具有事务功能, 可以保证数据库操作的一致性和完整性。事务的功能是通过使用数据库的事务日志来实现。

通常情况下, 事务日志记录了对数据库的所有修改操作。因此, 事务日志文件保存用于恢复数据库的事务日志信息, 是用来存储数据库更新情况的文件。因为事务的恢复主要靠日志来完成 (关于事务及事务的恢复在本书第 7 章有详细介绍), 所以事务日志文件在数据库文件中是必备的, 每个数据库必须至少有一个日志文件。事务日志的默认文件扩展名是 .ldf。

说明:	<ul style="list-style-type: none"><li>● 上述文件的名字是操作系统文件名, 由系统使用, 不由用户直接使用。虽然 SQL Server 不强制使用 .mdf, .ndf, .ldf 文件扩展名, 仍建议在创建数据库时使用这些默认扩展名, 以便标识文件用途。</li><li>● 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, 在实际应用环境中, 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。</li></ul>
-----	---

## 2. 逻辑和物理文件名称

SQL Server 的数据库文件有以下两个名称:

1) 逻辑文件名 (Logical\_file\_name)

逻辑文件名是在所有 T-SQL 语句中引用该文件时使用的文件名。逻辑文件名必须符合 SQL Server 的标识符规范, 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)

物理文件名包含目标路径, 它必须符合操作系统的文件命名规则。



### 3. 页和区

在理解 SQL Server 数据库物理结构时，需要注意两个概念：页和区。

1) 页 (Page)：SQL Server 中数据存储的基本单位是页，是 SQL Server 使用的最小存储单元。为数据库中的数据文件 (.mdf 或 .ndf) 分配的磁盘空间可以从逻辑上划分成页（从 0 到 n 连续编号）。每页大小是 8KB，即 8192 字节，这意味着 SQL Server 数据库中每 MB 有 128 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一个唯一的文件 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 10- 2 中显示了包含 4-MB 主数据文件和 1-MB 次要数据文件的数据库中的页码。

SQL Server 的数据记录全部以页为单位存储(事务日志文件除外)，表中的每一行数据都不能跨页存储。事务日志文件不包含页，而是包含一系列日志记录。

SQL Server 包括 8 种页类型，分别为：数据 (Data) 页，索引 (Index) 页，文本/图像 (Text/Image) 页，全局、共享全局分配映射 (Global、Shared Global Allocation Map) 页，可用空间 (Page Free Space) 页，索引分配映射表 (Index Allocation Map) 页，大容量更改映射表 (Bulk Changed Map) 页，差异更改映射表 (Differential Changed Map) 页。这些页的介绍见表 10- 1。

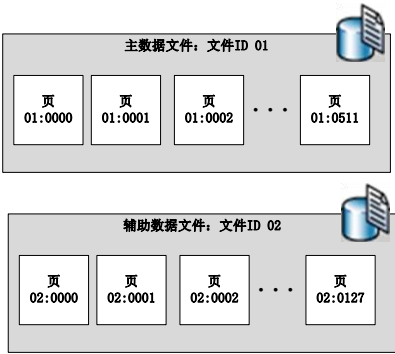


图 10- 2 数据页示例

表 10- 1 SQL Server 的页类型

页类型	内容
数据页	当 text in row 设置为 ON 时，包含除 text、 ntext、 image、nvarchar(max)、 varchar(max)、 varbinary(max) 和 xml 数据之外的所有数据
索引页	用于存储索引数据
文本/图像页	存储：（1）大型对象数据类型：text、 ntext、 image、nvarchar(max)、 varchar(max)、 varbinary(max) 和 xml 数据；（2）数据行超过 8 KB 时的可变长度数据类型列： varchar、 nvarchar、 varbinary 和 sql_variant
全局、共享全局分配映射页	有关区是否分配的信息。
可用空间页	有关页分配和页的可用空间的信息
索引分配映射页	有关每个分配单元中表或索引所使用的区的信息
大容量更改映射	有关每个分配单元中自最后一条 BACKUP LOG 语句之后的大容量操作所修改的区的信息
差异更改映射页	有关每个分配单元中自最后一条 BACKUP DATABASE 语句后更改的区的信息

其中,存储用户数据的数据页的结构如图 10-3 所示,它由三个主要部分组成:标头、数据行和行偏移表。

(1) 标头 (Header)

标头占用每个数据页前面的 96 字节 (剩下的 8096 字节用于数据、行系统开销和行偏移),用于存储有关页的体系信息。标头信息包含页码、页类型、页的可用空间以及拥有该页的对象的分派单位 ID。

(2) 数据行 (Data Row)

在数据页上,数据行紧接着标头按顺序放置。单个数据行的最大容量是 8060 字节的行内数据。

(3) 行偏移表 (Row Offsets)

页的末尾是行偏移表,它最少占用 36 字节。对应页中的每一行,每个行偏移表都包含一个 2 字节的条目用于记录对应行的第一个字节与页首的距离。

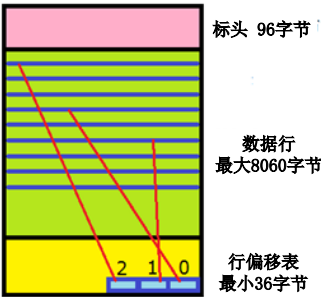


图 10-3 一个数据页的结构

偏移表中的条目顺序与数据行的顺序相反,而且只表示了页中数据行的逻辑顺序。例如,一张表有个聚集索引,那么行偏移矩阵的 slot0 引用聚集索引键顺序中的第一行,slot1 引用第二行,依此类推。注意:这些行的物理位置可以是页面上的任意位置。

行偏移表最少为 36 字节,因此默认最多只可以记录 18 个数据行。当数据行超过 18 行时,系统将动态调整数据行占用的字节数与行偏移表占用的字节数,注意:数据行与行偏移表总共是 8096 字节。

---

在 SQL Server 的数据页中,单个行的最大数据量和开销是 8060 字节。有的表中一行就可能超过 8KB,如果行数据中有 varchar、nvarchar、varbinary 和 sql\_variant 数据列的话,就单独用文本/图像页来存储这些类型的数据,两种页之间用指针来联系。

---

2) 区 (Extent): 又称为扩展盘区、盘区,是管理空间的基本单位。所有页都存储在区中。一个区是八个物理上连续的页 (即 64 KB),用来有效地管理页。这意味着 SQL Server 数据库中每 MB 有 16 个区。

区分为统一区和混合区,如图 10-4 所示。

(1) 统一区属于单个对象所有,只存储单个对象的数据。

(2) 混合区存放多个对象的数据,最多可由八个对象共享,区中八页的每页可由不同的对象所有。



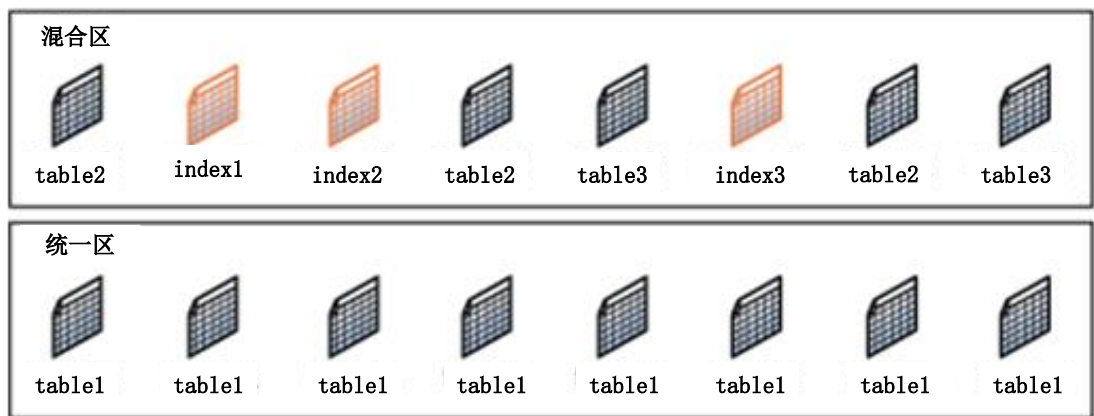


图 10-4 混合区和统一区

SQL Server 分配区的方法：

- (1) 不满 8 个页的数据尽量从已经存在的混合区中挑选一个满足的区进行分配；
- (2) 已经满 8 个页的数据则统一分配统一区。这样就可以尽可能地利用存储空间，提高空间利用率。
- (3) 对于新表或新的索引，从混合区分配页面以便尽可能地利用空间。当表或索引增长到 8 页时，将分配统一区。
- (4) 如果对现有表创建索引，并且该表包含的行足以在索引中生成 8 页，则对该索引的所有分配都使用统一区进行。

---

说明： 操作系统分配存储空间是以区为单位的，原因是页空间太小，如果按照页为单位来分配，分配空间次数过多，会造成系统性能下降。

---

通过理解数据库的空间管理，可以估算数据库的设计尺寸。数据库的大小等于所有表大小与索引大小之和。假设某个数据库只有一个表，该表的数据行字节是 800Byte。这时，一个数据页上最多只能放 10 行数据。如果该表大约有 100 万行的数据，那么该表占用 10 万个数据页的空间。因此，该数据库的大小估计为： $100000 \times 8KB = 800000KB = 781.25MB$ 。根据数据库大小的估计值，再考虑其他的因素，就可以得到数据库的设计值。

#### 4. 日志文件的结构

日志文件是 SQL Server 数据库中的单独的文件或一组文件。日志的存储不是以页为单位的，而是以一条一条的、大小不等的日志记录为单位进行的。若干条相邻的日志记录构成一个完整的事务，表明用户对数据库进行了某项完整的操作。日志文件的存储如图 10-5 所示。



图 10-5 事务日志文件的存储

## 5. 文件组 (File Group)

为了更好地管理数据库文件，从 SQL Server 7.0 开始引入了文件组的概念。可以将若干个分布在不同的硬盘驱动器上的数据文件组织到一个文件组中。通过文件组，可以将特定

的数据库对象与该文件组相关联，对数据库对象的操作都将在该文件组中完成，可以提高数据的查询性能。例如，可以分别在三个磁盘驱动器上创建三个文件 Data1.ndf、Data2.ndf 和 Data3.ndf，然后将它们分配给文件组 fgroup1。然后，可以明确地在文件组 fgroup1 上创建一个表，这样对表中数据的查询将分散到三个磁盘上，从而提高了性能。

SQL Server 提供了三种文件组类型，分别是主文件组 (Primary)、自定义文件组 (User\_defined) 和默认文件组 (Default)。

1) 主文件组：创建数据库时，系统自动创建主文件组，并将主数据文件和系统表的所有页都分配到主文件组中，此文件组还包含未放入其他文件组的所有辅助数据文件。每个数据库有且只有一个主文件组。

2) 自定义文件组：由用户创建的文件组。使用时，可以通过 SSMS 图形界面或者 T-SQL 语句中的 FILEGROUP 子句指定需要的文件组。

3) 默认文件组：在每个数据库中，同一时间只能有一个文件组是为默认文件组。如果在数据库中创建对象时没有指定对象所属的文件组，对象将被分配给默认文件组。如果没有指定默认文件组，则主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）。

- 
- 说明：
- 一个文件或文件组只能被一个数据库使用。
  - 一个文件只能是一个文件组的成员。
  - 事务日志文件不属于任何文件组。
- 

## 10.1.2 逻辑数据库与数据库对象

逻辑数据库是从数据库的逻辑角度描述数据库，是关于数据库的逻辑结构的描述。逻辑数据库将数据库视为数据库对象。

### 1. 数据库对象

数据库常见对象包括表 (Table)、视图 (View)、索引 (Index)、存储过程 (Stored Procedure)、触发器 (Trigger) 等，如表 10-2 所示。

表 10-2 数据库对象及功能描述

数据库对象名	功能描述
表	由行和列构成的集合，数据库中实际存储数据的对象
视图	是从一个或多个基本（数据）表中导出的表，也被称为虚表

续

数据库对象名	功能描述
索引	为数据提供快速检索的支持
约束	是实施数据一致性和完整性的方法
数据库关系图	包括各种表、每一张表的列名以及表之间的关系
存储过程	由 T-SQL 语言编写的程序
触发器	一种特殊的存储过程，当条件满足时，自动执行，用于保证数据完整性

### 10.1.3 系统数据库与用户数据库

#### 1. 系统数据库

系统数据库是由系统创建和维护的数据库，其中记录着 SQL Server 的配置情况、用户数据库的情况等信息，是系统管理的依据。

##### 1) master 数据库

master 数据库是 SQL Server 中最重要的数据库，记录着 SQL Server 系统的所有系统级信息，包括实例范围的元数据（例如登录帐户）、端点、链接服务器和系统配置设置。此外，master 数据库还记录了所有其他数据库的存在、数据库文件的位置以及 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。由于 master 数据库的重要性，所以禁止用户直接访问，并确保在修改之前有完整的备份。

说明：	使用 master 数据库时，建议：
	● 始终有一个 master 数据库的当前备份可用。
	● 执行下列操作后，尽快备份 master 数据库：1) 创建、修改或删除任意数据库；2) 更改服务器或数据库的配置值；3) 修改或添加登录帐户。
	● 不要在 master 数据库中创建用户对象

##### 2) model 数据库

model 数据库是模板数据库。每次创建新数据库时，SQL Server 都会生成 model 的副本作为新数据的基础。当发出 CREATE DATABASE（创建数据库）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。

例如，用户可以在 model 系统数据库中创建希望自动添加到所有新建数据库中的对象（如表、视图、数据类型、存储过程等），那么在新建数据库时，系统自动将 model 数据库中的所有用户自定义的对象都复制到新建的数据库中。

##### 3) msdb 数据库

msdb 数据库是代理服务数据库，给 SQL Server 代理提供必要的信息来运行作业，为其报警、任务调度和记录操作员的操作提供存储空间。因此是 SQL Server 中重要的数据库之一。

说明:	SQL Server 代理是 SQL Server 中的一个 Windows 服务，用以运行任何已创建的计划作业（例如包含备份处理的作业）。 作业是 SQL Server 中定义的自动运行的一系列操作，它不需要任何手工干预来启动。
-----	--

4) tempdb 数据库

tempdb 是一个临时数据库，它为所有的临时表、临时存储过程及其它临时操作提供存储空间。tempdb 数据库由整个系统的所有数据库使用，不管用户使用哪个数据库，他们所建立的所有临时表和存储过程都存储在 tempdb 上。SQL Server 每次启动时，tempdb 数据库被重新建立。当用户与 SQL Server 断开连接时，其临时表和存储过程自动被删除，并且在系统关闭后没有活动连接，因此 tempdb 中不会有什么内容从一个 SQL Server 会话保存到另一个会话。由于临时存放中间结果，因此无法备份和恢复 tempdb 数据库。

2. 系统数据库的物理文件

在安装 SQL Server 时，系统会自动创建系统数据库的数据文件和事务日志文件，默认安装位置为 Microsoft SQL Server\MSSQL10. MSSQLSERVER\MSSQL\DATA。

系统数据库文件如表 10- 3 所示。

表 10- 3 SQL Server 系统数据库文件

系统数据库	主文件	逻辑名称	物理名称	文件增长
master	主数据	master	master.mdf	按 10% 自动增长，直到磁盘已满
	Log	mastlog	mastlog.ldf	按 10% 自动增长，直到达到最大值 2TB
msdb	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
model	主数据	modeldev	model.mdf	按 10% 自动增长，直到磁盘已满
	Log	modellog	modellog.ldf	按 10% 自动增长，直到达到最大值 2TB
tempdb	主数据	tempdev	tempdb.mdf	按 10% 自动增长，直到磁盘已满
	Log	templog	templog.ldf	按 10% 自动增长，直到达到最大值 2TB

10.2 数据库的操作

在 SQL Server 中，用户可以自己创建用户数据库，并且可以对数据库进行修改和删除操作。

10.2.1 创建数据库

创建数据库就是为数据库确定名称、大小、存放位置、文件名和所在文件组的过程。若要创建数据库，必须至少拥有 CREATE DATABASE、CREATE ANY DATABASE 或 ALTER ANY DATABASE 权限。创建数据库的用户将成为该数据库的所有者。在一个 SQL Server 2008 实例中，最多可以创建 32767 个数据库。数据库的名称必须满足系统的标识符规则，在命名数据库时，一定要使数据库名称简短并有一定的含义。

在 SQL Server 中创建数据库的方法主要有两种：一是在 SSMS 窗口中通过图形化向导创建；二是通过编写 T-SQL 语句创建。下面以“Music 库”的创建为例，说明创建数据库的

### 1. 在 SSMS 中使用向导创建数据库

1) 打开 SSMS 窗口, 并使用 Windows 或 SQL Server 身份验证建立连接, 进入 SSMS 主界面。

3) 执行上述操作后, 会弹出“新建数据库”对话框, 在这个对话框中有三个页, 分别是“常规”、“选项”和“文件组”页。默认进入“常规”页, 如图 10-7 所示。

当输入数据名时，在下面的“逻辑名称”列中也有了相应的逻辑文件名，用户可以对这些名字进行修改。

图 10-6 选择“新建数据库”

(1) 逻辑名称——指定该文件的文件名。默认的主数据文件的逻辑文件名同数据库名；默认的第一个事务日志文件名为：“数据库名” + “log”。

(2) 文件类型——用于区别当前文件是数据文件还是日志文件。用户新建文件时，可以通过此列指定文件的类型。

(3) 文件组——显示当前数据库文件所属的文件组。一个数据库文件只能存在于一个文件组里。默认情况下，所有的数据文件都属于 **PRIMARY** 主文件组。事务日志文件不属于任何文件组。

(4) 初始大小——指定该文件的初始容量，在 SQL Server 中数据文件的默认值为 3MB，日志文件的默认值为 1MB。

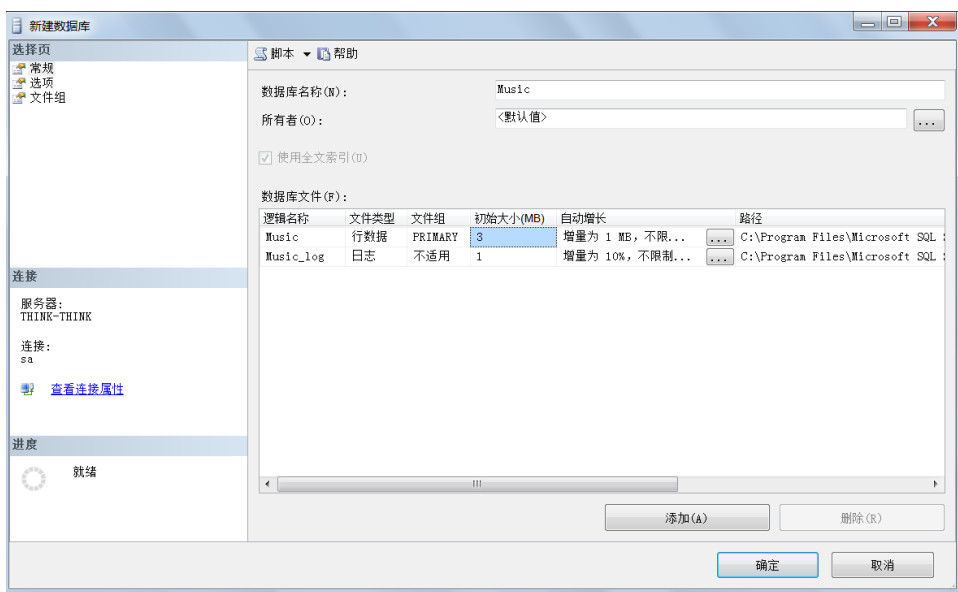


图 10-7 “新建数据库”的“常规”页

说明：在指定主数据文件的大小时，其大小不能小于 model 数据库中主数据文件的大小。因为系统是将 model 数据库主数据文件的内容拷贝到用户数据库的主数据文件上。

(5) 自动增长——用于设置在文件的容量不够用时，文件根据何种增长方式自动增长。通过单击“自动增长”列中的 按钮，打开“更改自动增长设置”窗口进行设置，可以更改文件的生长方式最大文件大小，如图 10-8 所示。

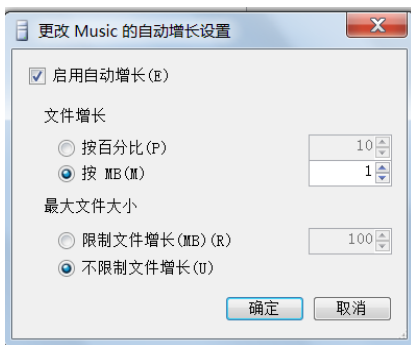


图 10-8 “更改自动增长设置”窗口

(6) 路径——指定存放该文件的物理存储位置。在默认情况下，SQL Server 将存放路径设置为 SQL Server 安装目录下的 data 子目录。单击该列中的 按钮可以打开“定位文件夹”对话框更改数据库的存放路径。

(7) 在图 10-8 中，默认选中“启用自动增长”复选框，即当数据库的初始空间用完后，系统自动扩大数据库的空间，这样可以防止因数据库空间用完而造成的不能插入新数据或不



能进行数据操作的错误。选中“启用自动增长”复选框后，可进一步设置每次文件增加的大小（或百分比）以及最大文件大小。如果选择“不限制文件增长”，则在有磁盘空间的情况下，文件一直增长，直至充满整个磁盘空间。

（8）单击图 10-7 中的“添加”或“删除”按钮，可以向数据库添加或删除辅助数据文件和事务日志文件，如图 10-9 所示。

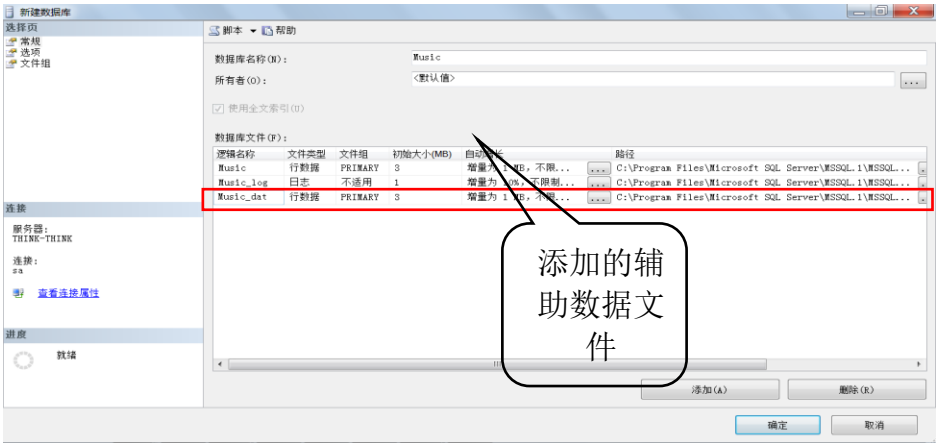


图 10-9 添加数据库文件

（9）单击“选项”按钮，可以设置数据库的一些选项，包括排序规则、恢复模式、兼容级别和其他需要设置的内容，如图 10-10 所示。

（10）单击“文件组”项可以添加或删除用户文件组，并可以设置默认文件组。如图 10-11 所示，添加了 usergroup 文件组。

（11）完成以上操作后，就可以单击“确定”按钮，关闭“新建数据库”对话框。至此，成功创建了一个数据库，可以通过“对象资源管理器”窗格查看新建的数据库。



图 10-10 新建数据库“选项”页



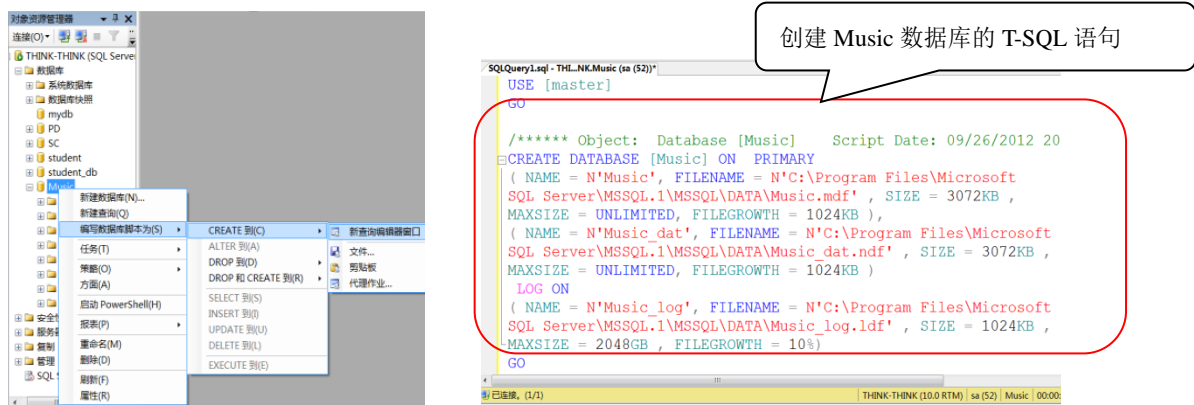


图 10-12 查看创建数据库的 T-SQL 语句

其中：

```

<filespec>::={ ([NAME=logica_file_name,]
FILENAME='os_file_name' [,SIZE=size[KB|MB|GB|TB]]
[,MAXSIZE={max_size[KB|MB|GB|TB]|UNLIMITED}]
[,FILEGROWTH=growth_increment[KB|MB|%]]) [1,...n]}
  
```

```

<filegroup>::={FILEGROUP filegroup_name <filespec> [1,...n]}
  
```

**例 10-1：**建立一个 Mytest 数据库，所有参数取默认值。

步骤如下：

- 1) 单击工具栏上的“新建查询”按钮，创建一个查询输入窗口；
- 2) 在查询编辑器中输入“CREATE DATABASE Mytest”；
- 3) 按<F5>键或点击工具栏上的“执行”按钮；
- 4) 右击“对象资源管理器”中的“数据库”节点，单击“刷新”按钮。可以看到 Mytest 数据库。结果如图 10-13 所示。

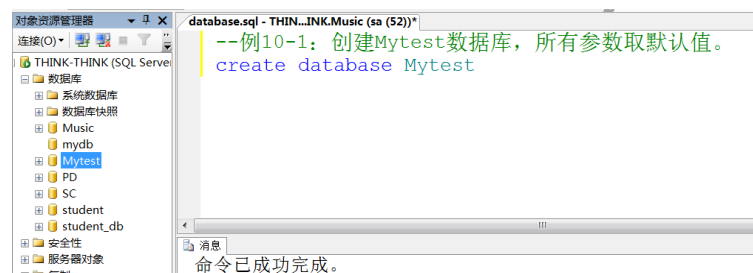


图 10-13 创建 Mytest 数据库

如果数据库中的数据文件或日志文件多于 1 个，则文件之间使用逗号隔开。当数据库有两个或两个以上的数据文件时，需要指定哪一个数据文件是主数据文件。默认情况下，第一个数据文件就是主数据文件，也可以使用 **PRIMARY** 关键字来指定主数据文件。

**例 10-2:** 创建 Employee 数据库，要求如下：

1) 主数据库文件名为 Employee，物理文件名为 Employee.mdf，初始大小为 5MB，最大文件大小为 100MB，增长幅度为 1MB；

2) 在文件组 usergroup1 上建立辅助数据文件 Employee\_dat，物理文件名为 Employee\_dat.ndf，初始大小为 3MB，最大为无限大，增幅为 1MB；

3) 日志文件逻辑文件名和物理文件名均为 Employee\_log，初始大小为 3MB，最大为 20MB，增幅为 10%。以上文件均存储在为 E:\mssql2008\data 文件夹中。

SQL 语句如下：

```
CREATE DATABASE Employee
--定义主数据文件
ON PRIMARY
( NAME =Employee, FILENAME = 'E:\mssql2008\data\Employee.mdf' ,
  SIZE = 5, MAXSIZE = 100 , FILEGROWTH = 1KB ),
--定义辅助数据文件
FILEGROUP usergroup1
( NAME = Employee_dat,
  FILENAME = 'E:\mssql2008\data\Employee_dat.ndf' ,
  SIZE = 3 , MAXSIZE = UNLIMITED, FILEGROWTH = 1)
--定义日志文件
LOG ON
(NAME = Employee_log,
 FILENAME = 'E:\mssql2008\data\Employee_log.ldf' ,
 SIZE = 3, MAXSIZE = 20, FILEGROWTH = 10%)
```

## 10.2.2 修改数据库

修改数据库最常用的两种方法为，通过图形界面和 ALTER DATABASE 语句。下面分别介绍这两种修改数据库的方法。

### 1. 使用 SSMS 图形界面修改数据库

右击“对象资源管理器”→“数据库”→Music 数据库，在弹出菜单中选择“属性”选项，可进入数据库属性窗口来快捷地修改数据库，如图 10-14 所示。

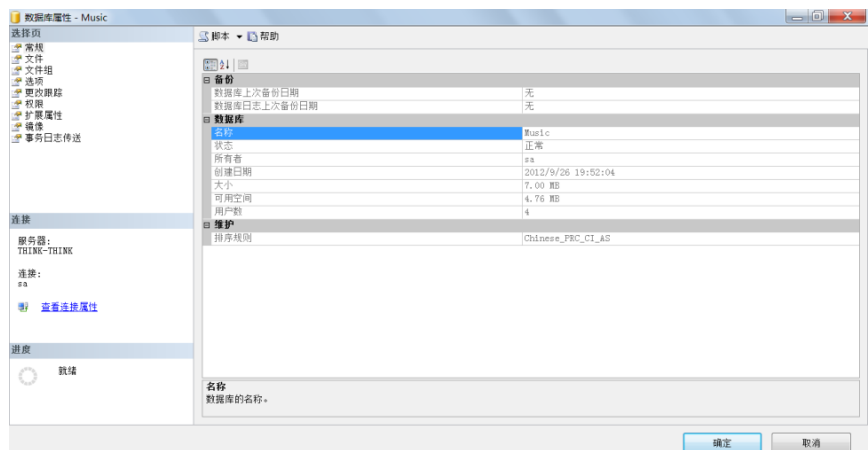


图 10-14 数据库属性窗口

通过选择不同的选项页，可以修改数据库的各种属性和设置。各选项页的说明如下：

- 1) “常规”页：显示当前数据库的基本情况，如数据库名称、状态、所有者、可用空间和排序规则等信息，这些信息只读，不能修改。
- 2) “文件”页：显示当前数据库文件信息。可在此窗口中修改数据库的所有者以及数据文件、事务日志文件的各种属性，也可添加/删除辅助数据文件和事务日志文件。
- 3) “文件组”页：显示当前数据库的文件组信息，可添加/删除文件组以及设置默认文件组。
- 4) “选项”页：显示并可修改当前数据库的排序规则、恢复模式、恢复选项和状态选项组等信息。
- 5) “权限”页：显示当前数据库的用户或角色以及他们相应的权限信息。可以为当前数据库添加、删除用户或角色，以及修改他们相应的权限。
- 6) “扩展属性”页：可以为当前数据库建立、删除扩展属性。
- 7) “镜像”页：显示当前数据库的镜像设置属性。
- 8) “事务日志传送”页：显示当前数据库的日志传送配置信息，可以为当前数据库设置事务日志备份计划、辅助数据库实例以及监视服务器实例。

## 2. 使用 ALTER DATABASE 语句

在 SQL Server 中，可用 ALTER DATABASE 语句修改数据库。

```
ALTER DATABASE database

{ ADD FILE < filespec > /* 添加新的数据文件 */
[ ,...n ]

[ TO FILEGROUP /* 将要添加的数据文件添加到指定的文件组中 */
```

ADD LOG FILE < filespec >	/* 添加新的事务日志文件 */
[ ,...n ]	
REMOVE FILE	/* 删除某一文件 */
logical_file_name	
ADD FILEGROUP filegroup_name	/* 添加一个文件组 */
REMOVE FILEGROUP	/* 删除一个文件组 */
filegroup_name	
MODIFY FILE < filespec >	/* 修改某个文件的属性 */
MODIFY NAME = new_dbname	/* 修改数据库的名字 */
MODIFY FILEGROUP filegroup_name	/* 修改某个文件组的属性或为文件组定义一个新名字。文件组的属性有三种: READONLY(只读)、READWRITE(读写)、Default(默认) */
{filegroup_property NAME=new_filegroup_name }	

其中< filespec >的格式同 CREATE DATABASE 中的< filespec >。

**例 10-3:** 为 Employee 数据库的 usergroup2 文件组添加一个辅助数据文件 Employee\_dat2, 要求: 文件存储在 E:\mssql2008\data 文件夹下, 初始大小为 10MB, 最大为 20MB, 增幅为 5MB。

对应的 SQL 语句为:

```
Alter DATABASE employee
Add file ( NAME = employee_dat2,
FILENAME='E:\mssql2008\data\Employee_dat2.mdf',
SIZE = 10, MAXSIZE = 50, FILEGROWTH = 5) to filegroup usergroup2
```

如果要增加日志文件, 可以使用 ADD LOG FILE 子句;

说明: 在一个 ALTER DATABASE 语句中, 一次可以增加多个辅助数据文件或事务日志文件, 多个文件之间需要使用“,”分开。

**例 10-4:** 删除 Employee 数据库的文件组 usergroup2。

因为 Employee 数据库的文件组 usergroup2 中包含有一个辅助数据文件 Employee\_dat2(见例 10-3), 如果直接删除文件组, 将会出现图 10-15 的错误。

图 10-15 删除文件组出错页面

因此, 需要注意: 删除文件组必先删除文件组中所包含的文件。



本例对应的语句为：

```
--删除文件组中的文件
alter database employee remove file employee_dat2

--删除文件组
alter database employee remove filegroup usergroup2
```

---

说明： 删除数据库文件意味着从操作系统上删除该文件，不能恢复。

---

**例 10-5：**修改 Employee 中的数据文件 Employee\_dat,将其初始大小改为 10MB，最大容量改为 20MB，增幅设为 2MB。对应的 SQL 语句为：

```
ALTER DATABASE employee
MODIFY FILE
(NAME = Employee_dat, SIZE = 10, MAXSIZE = 20, FILEGROWTH = 2)
```

---

修改数据库文件时，须注意：

- 说明：
- 一个 ALTER DATABASE 语句只能修改一个文件（不论是数据文件还是事务日志文件）；
  - 修改文件时，只需要指定文件的逻辑文件名，不须指出文件的物理位置；
  - 如果需要修改文件大小，新数值必须大于原来文件的大小。
- 

### 10.2.3 删除数据库

当不再需要用户定义的数据库，或者已经将其移到其他数据库或服务器上，即可删除数据库。删除数据库时，须注意：只有 sysadmin 和数据库的拥有者有删除数据库的权限；不能删除系统数据库；不能删除当前正在使用的数据库。

执行删除数据库操作会从 SQL Server 实例中删除数据库，并删除该数据库使用的物理磁盘文件。如果在执行删除操作时，数据库或它的任意一个文件处于脱机状态，则不会删除对应的磁盘文件，用户可从磁盘上手动删除这些文件。

---

说明： 在删除数据库之后，建议备份 master 数据库，因为删除数据库将更新 master 数据库的信息。

---

在 SQL Server 中，有两种删除数据库的方法：使用图形界面和 DROP DATABASE 语句。

#### 1. 使用 SSMS 图形界面删除数据库

- 1) 在“对象资源管理器”窗格中选中要删除的数据库，右击选择“删除”命令。
  - 2) 在弹出的“删除对象”对话框中，单击“确定”按钮确认删除，如图 10-16 所示。
- 删除操作完成后会自动返回 SSMS 窗口。

#### 2. DROP DATABASE 语句

使用 DROP DATABASE 语句删除数据库的语法如下：

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

其中，database\_name 为要删除的数据库名，[,...n]表示可以有多于一个数据库名。

例如，要删除数据库“Mytest”，可使用如下的 DROP DATABASE 语句：

```
DROP DATABASE Mytest
```

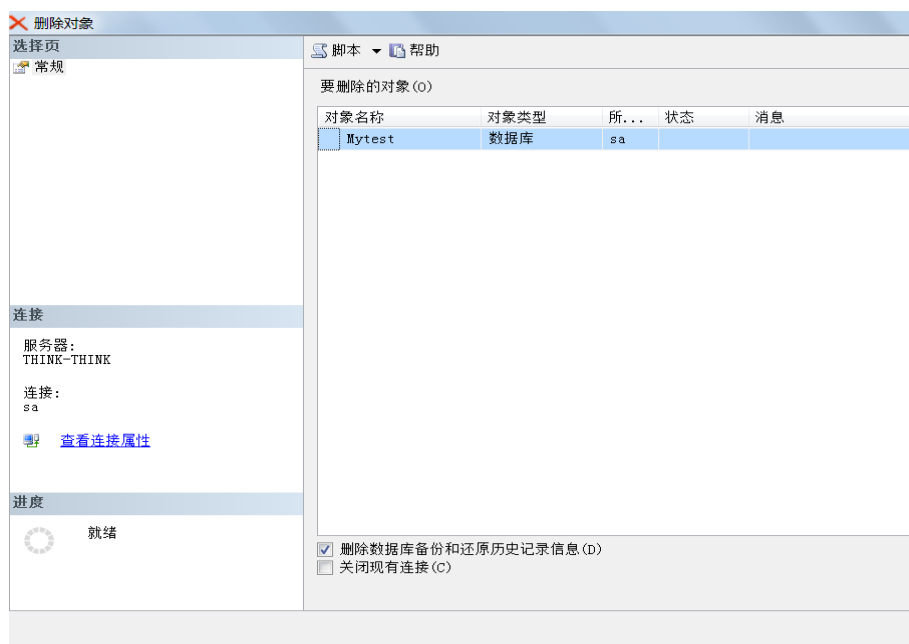


图 10- 16 删除数据库窗口

## 10.2.5 分离和附加数据库

SQL Server 服务器由若干个数据库组成，除了 master、model 和 tempdb 这 3 个系统数据库外，其余的数据库都可以从服务器的管理中分离出来，脱离服务器的管理，同时保持数据文件和日志文件的完整性和一致性，这样分离出来的数据库的日志文件和数据文件可以附加到其他 SQL Server 服务器上构成完整的数据库，附加的数据库和分离时完全一致。

### 1. 分离数据库

分离数据库就是指将数据库从 SQL Server 2008 的实例中分离出去，但是不会删除该数据库的文件和事务日志文件，这样，该数据库可以再附加到其他的 SQL Server 的实例上去。下面以“Music”数据库为例来介绍分离和附加操作。

#### 1) 利用图形化界面分离数据库

可以使用图形界面来执行分离数据库的操作。步骤如下：

(1) 在“对象资源管理器”窗格中右击拟要分离的数据库（如 Music 数据库），选择“任务”→“分离”命令。

(2) 在打开的“分离数据库”对话框，查看在“数据库名称”列中的数据库名称，验证这是否为要分离的数据库，如图 10- 17 所示。

(3) 图 10- 17 中，在“状态”列中的是如果显示的是“未就绪”，则“消息”列将显示有关数据库的超链接信息。当数据库涉及复制时，“消息”列将显示 Database Replicated。

(4) 数据库有一个或多个活动连接时，“消息”列将显示“<活动连接数>个活动连接”。在可以分离数据列之前，必须启用“删除连接”复选框来断开与所有活动连接的连接。

(5) 分离数据库准备就绪后，单击“确定”按钮。

2) 使用系统存储过程分离数据库

可以使用 sp\_detach\_db 存储过程来执行分离数据库操作。

**例 10-8:** 要分离 Music 数据库，则该执行语句如下所示：

```
EXEC sp_detach_db Music
```

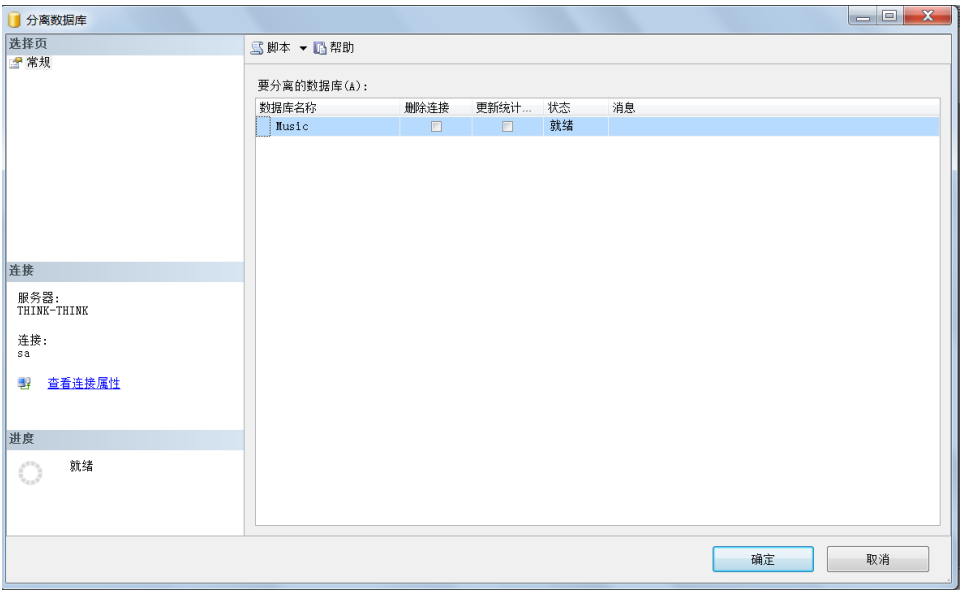


图 10- 17 “分离数据库”界面

2. 附加数据库

分离后的数据库的数据文件和事务日志文件，可以重新附加到同一或其他 SQL Server 的实例中。在附加数据库时，所有数据库文件（.mdf 和.ndf 文件）都必须是可用的。如果任何数据文件的路径与创建数据库或上次附加数据库时的

路径不同，则必须指定文件的当前路径。

附加数据库可以很方便地在 SQL Server 服务器之间利用分离后的数据文件和日志文件组织成新的数据库。因此，在实际工作中，分离数据库作为对数据基本稳定的数据库的一种备份的办法来使用。

下面就将分离后的“Music”数据库再附加到当前数据库实例中。

1) 使用图形界面附加数据库

具体操作步骤如下所示：

(1) 在“对象资源管理器”窗格中，右击“数据库”节点并选择“附加”命令，出现图 10- 18 的“附加数据库”对话框。

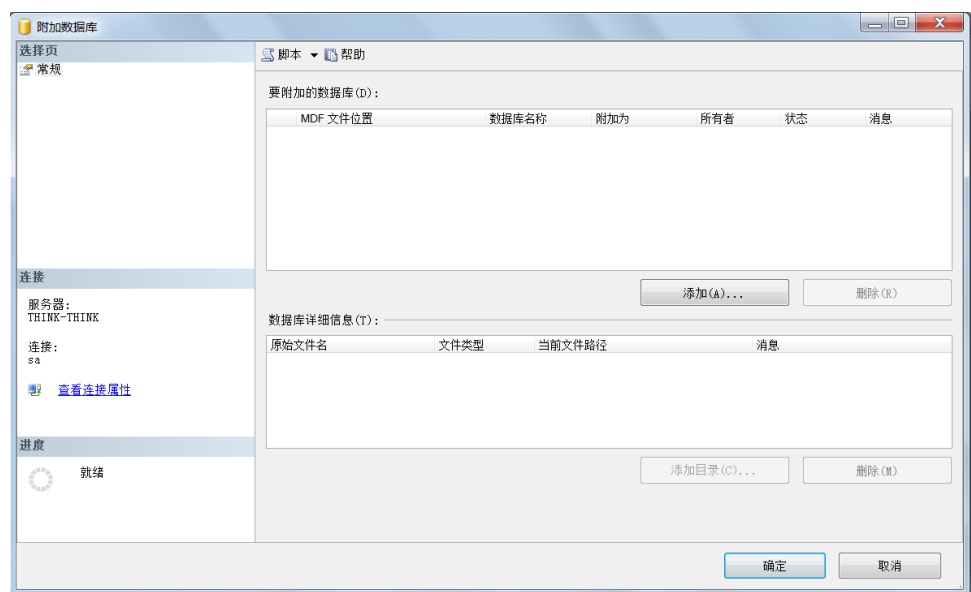


图 10- 18 “附加数据库”界面

(2) 在打开的“附加数据库”对话框中单击“添加”按钮，从弹出的“定位数据库文件”对话框中选择要附加的数据库所在的位置，再依次单击“确定”按钮返回，如图 10- 19 所示。

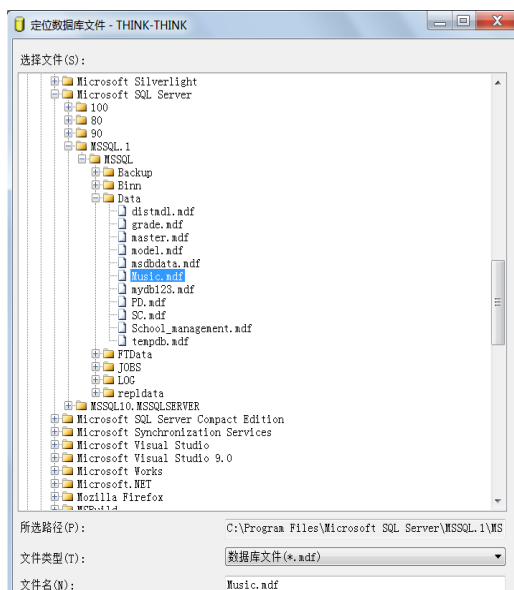


图 10-19 “定位数据库文件”

(3) 出现如图 10- 20 所示的“附加数据库”界面，确定无误后，单击“确定”按钮。

图 10-20 已选定数据库文件的“附加数据库”界面

(4) 回到“对象资源管理器”中，展开“数据库”节点，将看到 Music 数据库已经成功附加到了当前的实例数据库。

## 2) 利用 T-SQL 语句来附加数据库

CREATE DATABASE 语句可通过结合 ON 子句和 FOR ATTACH 语句进行数据库附加操作,附加时会加载该数据库所有的文件,包括主数据文件、辅助数据文件和事务日志文件。执行语句语法如下:

```
CREATE DATABASE database_name
ON <filespec> [ ,...n ] FOR ATTACH
```

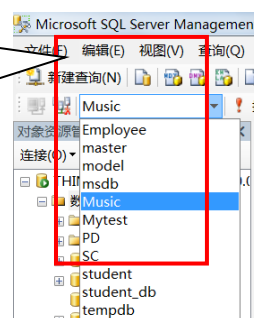
**例 10-9:** 以上介绍的图形化附加 Music 数据库的操作也可用以下的 T-SQL 语句来完成:

```
Create database Music
ON (filename='C:\Program Files\Microsoft SQL Server\ MSSQL.1 \MSSQL
\Data\Music.mdf') for attach
```

## 10.2.6 设置当前数据库

一个 SQL Server 数据库实例中最多可以包含 32768 个数据库，当前数据库指当前可以操作的数据库。在用 T-SQL 语句创建表、视图等数据库对象或对这些对象进行操作时，若无显式的指定数据库名，系统默认是在当前数据库中进行。SQL Server 默认的当前数据库是系统数据库 master。用户可以设定某个数据库作为当前数据库。

“可用数据库”列表



设定当前数据库的方法有两种：

图 10- 21 “可用数据库”列表

一是通过选择“可用数据库”下拉列表进行选择，图 10- 21 就是从“可用数据库”列表中选择“Music”作为当前数据库。

二是可以在查询编辑器窗口中使用 USE 语句。其语法格式为：

```
USE database_name /* database_name 是设定为当前数据库的名称 */
```

若要设“Music”为当前数据库，就可用以下语句：

```
Use Music
```