

## 任务二 为数据管理创建数据库环境

——创建和管理 SQL Server 数据库

### 任务

在 SSMS 中创建数据库 JXGL，并对 JXGL 进行简单管理；使用 CREATE DATABASE 语句创建数据库 JXGL，使用 T-SQL 语句对 JXGL 进行简单管理。

### 目标

- 了解 SQL Server 数据库的存储结构
- 建立事务日志的概念
- 使用 SSMS 创建数据库
- 使用 CREATE DATABASE 语句创建数据库
- 使用 SSMS 管理数据库
- 使用 T-SQL 语言管理数据库

## 2.1 认识 SQL Server 数据库

**任务：**了解 SQL Server 数据库的存储结构，建立事务日志的概念。

SQL Server 数据库的存储结构分为逻辑存储结构和物理存储结构。逻辑存储结构指的是数据库是由各种数据库对象组成的；物理存储结构是讨论数据库文件如何在磁盘上存储的。事务日志是 SQL Server 数据库的一个亮点，它对数据库恢复起着至关重要的作用。

### 2.1.1 SQL Server 数据库的存储结构

从逻辑存储结构角度看，SQL Server 数据库是由表、视图、索引、存储过程等各种数据库对象组成的，它们被数据库管理系统管理。

从物理存储结构角度看，数据库实际上是一个文件的集合，在磁盘上以文件为单位存储。数据库的物理存储结构是用来讨论数据库文件如何在磁盘上存储的。

#### 1. 数据库的物理文件

数据库是存储数据的容器，数据以文件形式存储在磁盘上，称为数据库的物理文件或操作系统文件。每个 SQL Server 2005 数据库包含两类物理文件：数据文件和日志文件。

数据文件存储数据和数据库对象，如表、索引。第一个数据文件称为主数据文件，是数据文件的起点，指向其他数据文件，建议文件名后缀为.mdf (Primary Data File)；日志文件用来存储恢复数据库中所有事务所需的信息，建议文件名后缀为.ldf (Log Data File)。

每个 SQL Server 数据库至少包含两个物理文件：一个主数据文件和一个日志文件。当数

数据库存储的数据量巨大，超过了单个 Windows 文件的最大值，或需要将重要数据与次要数据分开存储时，用户可自行创建次要数据文件。次要数据文件辅助主数据文件存储数据，文件名后缀为.ndf (Secondary Data File)；建议将不同的次要数据文件放在不同的磁盘驱动器中。

一个 SQL Server 2005 数据库只包含一个主数据文件，必须至少包含一个日志文件，可以不包含或包含多个次要数据文件。数据库中所有文件的位置信息都存储在数据库的主文件和系统数据库 master 中。

**提示：**默认情况下，数据和日志文件被放在同一个驱动器的同一路径下。但在实际生产环境中，建议将数据文件和日志文件放在不同的磁盘上。

## 2. 数据库文件的名称

每个 SQL Server 2005 数据库文件有两个文件名：逻辑文件名和操作系统文件名。

(1) 逻辑文件名。在 T-SQL 语句中引用数据库文件时使用的名称。其命名必须符合 SQL Server 标识符规则，且在同一数据库中必须唯一。

(2) 操作系统文件名。包含存储目录路径的数据库文件名称。其命名必须符合操作系统文件命名规则，且在同一数据库中必须唯一。

## 3. 数据库文件的存储

(1) 数据页。数据文件的基本存储单位为 8KB (8192B) 的连续空间，称为一个页 (Page)。

SQL Server 2005 中包含数据页、索引页、文本/图像页等 8 种类型的页。数据页用来存储除 text、ntext 和 image 类型以外的数据。页由以下三部分组成：

1) 页头。页首的 96KB 用来存储页的系统管理信息，如页类型、页的可用空间、页码及拥有该页的数据库对象的分配单元 ID 值等信息。

2) 数据区。以槽 (Slot) 为单位的空间，用来存储数据。数据记录 (行) 就存储在槽中，紧接着页头按顺序放置。

3) 行偏移数组。映射槽的相对位置，通过查询行偏移数组中某个槽的偏移量，就可以准确定位槽在数据区的位置。

一个页只能存储一个数据库对象，如表、索引。当一个对象太大容不下时，对象会被切断；当一个对象不能占满一个页时，剩余部分空闲不能再存储其他数据库对象。数据行不能跨页存储，页中每行的最大容量为 8060B。

SQL Server 2005 的读写操作都在数据页上。对表中的每条记录采用一个唯一的<页号，槽号>进行标注，称为记录标识，记录标识在唯一的数据页的特定槽中进行存储。这样系统就可以根据记录标识来操作物理数据，如图 2-1 所示。

(2) 盘区。SQL Server 2005 将 8 个物理上连续的页作为一个单元为数据库对象分配存储空间，这个单元称为盘区 (Extent)，容量为 64KB。

以盘区为分配单位，就不会因分配存储空间的操作十分繁忙而影响数据库性能，提高了空间分配效率。但将一个盘区分配给一个只包含少量数据的数据库对象时，会浪费存储空间。

SQL Server 2005 将盘区分为统一盘区和混合盘区。统一盘区只分配给单个数据库对象，而混合盘区最多由 8 个数据库对象共享。分配盘区的基本策略是：容量不满 8 个页的数据库对象尽量从已经存在的混合盘区中挑选一个进行后续分配；为容量满 8 个页的数据库对象分配统一盘区。这样大大提高了空间利用率。

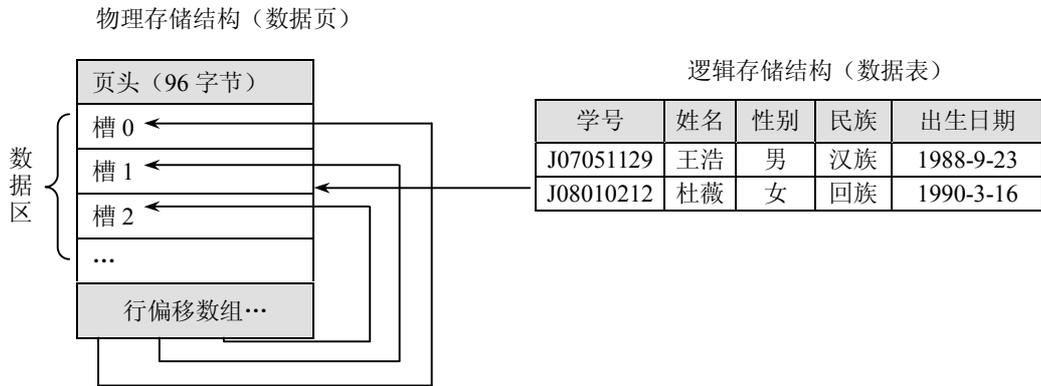


图 2-1 数据的逻辑结构和物理结构

提示：页和盘区用来管理数据文件的存储空间，日志文件的存储不以页为单位，而以日志记录形式存储。

SQL Server 2005 数据库的物理存储结构如图 2-2 所示。

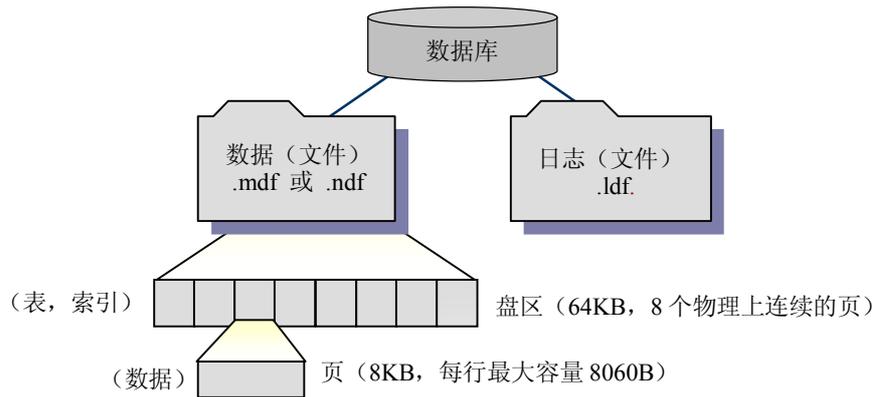


图 2-2 数据库的物理存储结构

#### 4. 数据文件组

为便于分配数据和管理文件，可以将数据文件组织到不同的文件组中。SQL Server 2005 包含两种类型的文件组：主文件组（Primary）和用户定义文件组。

主文件组包含主数据文件和未放入其他文件组的所有次数据文件。每个数据库只有一个主文件组，也是数据库的默认文件组，系统表的所有页均分配在主文件组中。

数据库包含多个数据文件时，用户也可以自定义文件组，将次文件放到自定义文件组中。

例如，一个数据库包含两个次数据文件，分别存储在不同的磁盘驱动器上。用户可以创建一个文件组 filegroupxx，并将这两个次数据文件指派到该文件组中。当对该文件组包含的数据进行查询时，查询动作会分散到两个磁盘上进行，从而提高了查询性能。

数据文件如何记录数据是由文件组控制的，文件组采取按比例填充数据的策略。当写入数据时，SQL Server 2005 根据文件中的可用空间量将数据按一定比例写入组中的每个文件，而不是将数据写满主数据文件，再写入下一个数据文件。

**提示：**文件组是命名的文件集合，只包含数据文件，日志文件不包括在任何文件组内。

### 2.1.2 事务日志

事务 (Transaction) 是需要一次完成的操作集合。事务作为 SQL Server 的单个逻辑工作单元，必须彻底执行完或根本不执行。如在数据库中创建一张数据表，对数据库中的某一数据进行更新等。

事务日志用来记录所有事务和每个事务对数据库所做的更新操作，以日志文件形式存储在磁盘中。实际上，事务日志记录了什么时候谁对数据库数据执行了哪些修改操作，好比是数据库的“黑匣子”。当数据库发生故障时，可以通过日志进行恢复。

创建一个数据库时，必须指定存储这个数据库的事务日志的文件。

#### 1. 事务日志的作用

事务日志记录事务及其修改操作，有以下三种作用：

- (1) 可以单独恢复个别事务。使用日志记录回滚掉某个未完成的事务所做的修改。
- (2) 启动 SQL Server 2005 实例时，对每个数据库执行恢复操作，前滚事务日志中记录的每个未完成的事务，保证数据库的完整。
- (3) 将恢复的数据库、文件、文件组或页回滚到某一故障点。

#### 2. 事务日志的特点

- (1) 作为单独的文件实现，可以根据需要定义文件为自动增长，以减少空间不足的可能，也减少了管理开销。
- (2) 日志文件与数据文件分开存放，向数据文件和日志文件写入数据时减少冲突。
- (3) 事务日志可以单独备份、恢复。
- (4) 不必遵守数据页的存储格式。
- (5) 日志的截断机制（减小逻辑日志文件的大小并释放磁盘空间以供重新使用）速度快，对事务的容量影响小。
- (6) 事务日志还记录了数据页的分配和释放信息。

#### 3. 事务日志的工作流程

为了维护数据库的完整性，事务必须彻底完成或根本不执行。如果一个事务只是部分执行并作用于数据库，数据库的完整性将遭到破坏。事务日志可以防止未完成的事务破坏数据库。工作过程如下：

- (1) 应用程序发出一个更新事务（数据修改操作指令）。
- (2) 事务日志记录一个事务开始标记，并将影响到的数据页从磁盘读入缓冲区。
- (3) 事务被提交后，事务包含的每一条更新语句都记录到日志文件中，并记录一个提交事务的标记。
- (4) 在缓冲区更新数据，检查点进程将更新后的数据页写入数据库中，并在事务日志中记录一个检查点标记。该标记用以在恢复数据库时确定事务的起点和终点，以判断哪些事务已作用于数据库。
- (5) 若系统出现故障，自动恢复进程使用事务日志前滚所有已提交事务，并回滚任何未完成的事务。

**提示：**日志对每一个数据库是不是必须的呢？如 Access、FoxPro 这样的桌面型数据库就

没有日志的概念。为什么 SQL Server 数据库必须建立日志呢？一个简单例子：在 2008 年的某天，用户突然需要查看数据库中 2005 年的数据。对 Access 是做不到的，因为数据只反映当前的瞬时状态。而 SQL Server 却可以完成这样的需求：只要在 2005 年之前对数据库做过备份，则可重新执行从该备份到 2005 年的日志，就可以将数据恢复到指定的时刻。

## 2.2 创建 SQL Server 数据库

**任务：**创建 JXGL 数据库。创建数据库前，必须首先明确数据库名称、数据库所有者、初始容量、最大容量、数据库文件的存储路径及增长方式等。

创建用户数据库的过程就是为数据库确定名称、大小和数据库文件名称及存放位置的过程。新建数据库的信息存放在系统数据库 master 中，属于系统级信息。创建 SQL Server 数据库主要有两种方法：使用视图化管理工具 SQL Server Management Studio (SSMS) 和在代码编辑器中使用 T-SQL 语言中的 CREATE DATABASE 语句。

### 2.2.1 使用 SSMS 创建数据库

在创建数据库前，要先进入图形界面 SSMS：选择“开始”→“所有程序”→“Microsoft SQL Server 2005”→“SQL Server Management Studio”命令。

创建示例数据库 JXGL 的操作步骤如下：

(1) 在 SSMS 中选择“视图”→“对象资源管理器”命令。

(2) 展开“数据库”节点。在“对象资源管理器”中单击“数据库”前的  按钮，展开“数据库”节点。

(3) 打开“新建数据库”窗口。在“数据库”节点上右击，在弹出的快捷菜单中选择“新建数据库”，打开“新建数据库”的“常规”选项卡窗口。

(4) 输入数据库名称。在“数据库名称”文本框中输入在数据库列表中不存在的数据库名称，如 JXGL。观察发现，随着输入数据库名称，在“数据库文件”列表框的“逻辑名称”栏中将自动产生数据文件和日志文件的逻辑文件名：JXGL 和 JXGL\_log。这个名称可以修改。

(5) 选择数据库所有者。在“所有者”文本框中输入新数据库的所有者名称。默认值为“默认值”账户，即当前登录到 SQL Server 的用户账户。如果使用 Windows 身份验证模式登录，则所有者为 Windows 用户账户；如果使用 SQL Server 身份验证模式登录，所有者是连接时使用的 SQL Server 用户账户。

(6) 设置全文索引。忽略“使用全文索引”复选框。如果想让数据库具有能搜索特定词或短语的列，则选中该复选框。如数据库引擎中有一个列包含来自网页中的一组短语，可以用全文搜索找到包含正在搜索的这组短语的网页。

(7) 设置数据文件属性。“数据库文件”列表框的第一行为主数据文件。单击“逻辑名称”栏，修改或重命名数据文件的逻辑文件名，常规命名规则为“数据库名\_data”，如 JXGL\_data。单击“初始大小”栏，重新输入或单击  按钮修改数据文件的初始容量为 4。数据库的初始容量即不包含任何数据时空数据库的大小，默认单位为 MB。单击“自动增长”栏的  按钮，弹出“更改数据文件自动增长设置”对话框。单击“启用自动增长”单选框，设置数据文件的自动增长率和最大容量值，如图 2-3 所示。文件自动增长及增长率是指数据库在超过所分配的

初始空间时是否增长以及每次按多大容量增长（默认单位为 MB）。设置最大容量可以防止数据库自动增长到填满整个磁盘。该文件自动指派给主文件组。

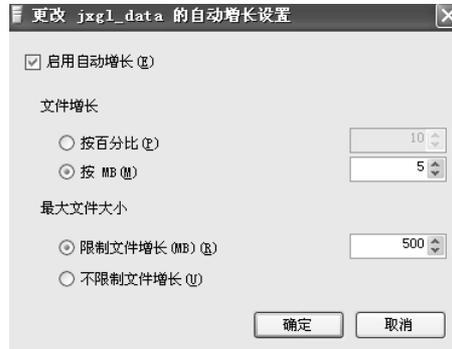


图 2-3 “更改数据文件自动增长设置”对话框

(8) 设置数据文件存储路径。单击“路径”栏的  按钮，打开“定位文件夹”对话框，选择数据文件的存储路径，如 D:\JXGL。

**提示：**只能选择已经在磁盘上存在的目录。数据库文件默认的存储路径是 SQL Server 2005 的安装路径。

(9) 设置日志文件属性。同以上 (7)、(8) 步骤，将日志文件命名为 JXGL\_log，初始容量为 5MB，设置文件增长方式为自动增长，增长率为 5%，限制文件最大容量为 1024MB(1GB)。选择文件存储路径与数据文件存储路径相同，即 D:\JXGL。

(10) 添加次数据文件。单击窗口右下角的  按钮，在“数据库文件”列表框的第三栏添加次数据文件 JXGL\_data02，初始容量为 2MB，自动增长率为 10MB，最大容量为 500MB，存储路径与主文件相同。单击“文件组”栏的  按钮，在下拉菜单中选择“新文件组”，打开“新建文件组”对话框，在“名称”文本框输入新文件组的名称 filegroup1。将次数据文件 JXGL\_data02.ndf 指派给文件组 filegroup1。

(11) 添加日志文件。单击窗口右下角的  按钮，在“数据库文件”列表框的第四栏添加第二个日志文件 JXGL\_log02，初始容量为 5MB，自动增长率为 5MB，最大容量为 500MB，存储路径为 D:\JXGL。

(12) 如果发现有的文件不需要，可单击  按钮将其删除。

(13) 单击  按钮，完成数据库 JXGL 的创建，或单击  按钮，中断并退出数据库的创建。完成数据库常规设置的“新建数据库—常规”窗口如图 2-4 所示。

(14) 回到“对象资源管理器”窗口，鼠标指向“数据库”节点，右击，在快捷菜单中选择“刷新”命令，刷新数据库节点，会看到数据库 JXGL 已排列于数据库列表中。

**提示：**在“新建数据库”窗口的“数据库文件”列表框中有一个“文件名”栏，该栏不能进行编辑，用来存储数据库的物理文件名，默认为逻辑文件名.mdf 或.ndf 或.ldf。单击  按钮完成数据库创建后，自动产生。

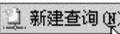
在“对象资源管理器”窗口中右击“数据库”节点，在快捷菜单中选择“刷新”命令；或选中“数据库”节点，单击“对象资源管理器”上方的  按钮，刷新“数据库”节点，会看到数据库 JXGL 已排列于数据库列表中了。



图 2-4 “新建数据库”对话框的“常规”选项卡窗口

## 2.2.2 使用 T-SQL 创建数据库

### 1. CREATE DATABASE 的基本语法

单击 SSMS “标准”工具栏中的  按钮，打开一个 SQL 编辑器窗口，使用 T-SQL 语言的 CREATE DATABASE 语句可以创建数据库。基本语法如下：

```
CREATE DATABASE database_name
    [ ON
        [ PRIMARY ] [ < filespec > [ , ...n ] ]
        [ , < filegroup > ] [ , ...n ]
    ]
    [ LOG ON
        [ < filespec > [ , ...n ] ]
    ]
```

其中：< filespec > ::=

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

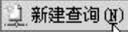
其中: <filegroup> ::=  
 FILEGROUP filegroup\_name [ DEFAULT ]  
 各参数含义如表 2-1 所示。

表 2-1 参数及其含义

| 参数                          | 含义  |
|-----------------------------|---|
| database_name               | 新数据库名称。在同一服务器中必须唯一  |
| ON                          | 指定存储数据库中数据部分的磁盘文件清单   |
| PRIMARY                     | 指定主文件组包含的文件。CREATE DATABASE 语句中列出的第一个文件将成为主文件   |
| LOG ON                      | 指定存储事务日志的磁盘文件清单   |
| NAME(logical_file_name)     | 指定数据库文件的逻辑名称。可以是字符或 Unicode 常量,也可以是常规标识符或分隔标识符。在数据库中必须唯一  |
| FILENAME ('os_file_name')   | 指定操作系统文件名,由操作系统使用的路径和文件名组成。创建数据库时指定的路径必须已经存在。不应将数据文件特别是日志文件放在压缩文件系统中  |
| SIZE                        | 文件的初始大小。默认初始值为 1MB 或与 model 数据库的主文件大小相同,默认单位是 MB  |
| MAXSIZE                     | 指定文件可增大到的最大容量。可以使用 KB、MB、GB 和 TB 为单位,默认为 MB。必须指定一个整数  |
| FILEGROWTH growth_increment | 指定文件的自动增量。不能超过 MAXSIZE 设置。该值可以 MB、KB、GB、TB 或百分比(%)为单位,默认为 MB。如果指定 %,则增量大小为发生增长时文件大小的百分比。值为 0 时表明自动增长被设置为关闭,不允许增加空间。如果未指定 FILEGROWTH,则数据文件的默认增长值为 1 MB,日志文件的默认增长比例为 10%,并且最小值为 64 KB |
| FILEGROUP filegroup_name    | 文件组逻辑名。在数据库中必须唯一,不能是系统提供的名称 PRIMARY 和 PRIMARY_LOG。名称可以是字符或 Unicode 常量,也可以是常规标识符或分隔标识符,必须符合标识符规则   |
| DEFAULT                     | 指定命名文件组为数据库中的默认文件组  |

## 2. 在 SSMS 中使用 T-SQL 语句

SQL Server 2005 的 SSMS 中提供了代码编辑器,在该环境中可以直接编辑、运行 T-SQL 代码。步骤如下:

(1) 单击 SSMS 的标准工具栏中的  按钮,打开“查询编辑器”窗口。窗口分为上、下两个窗格,上窗格是代码编辑区,编辑 T-SQL 语句;下窗格为结果/消息区,显示代码的编译、运行结果。

(2) 在代码编辑窗格中输入 T-SQL 语句。

(3) 单击“SQL 编辑器”工具栏中的  按钮,对输入的 T-SQL 语句进行语法分析。没有发现语法错误,在“结果”窗格中显示“命令已成功完成”,如图 2-5 所示。单击  按钮,在“消息”窗格中显示“命令已成功完成”,如图 2-6 所示。

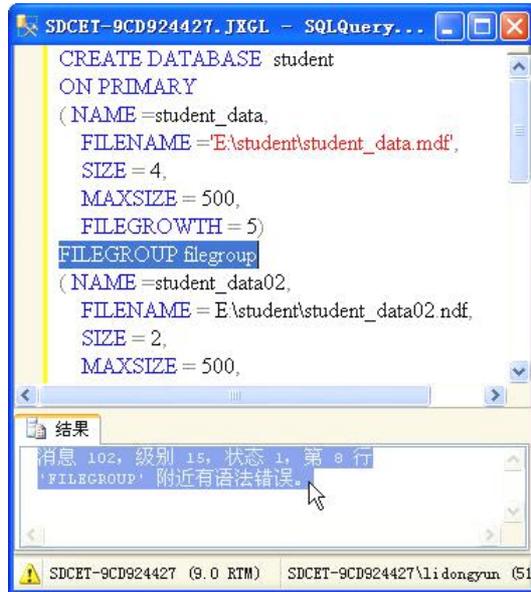


图 2-5 显示语法分析结果

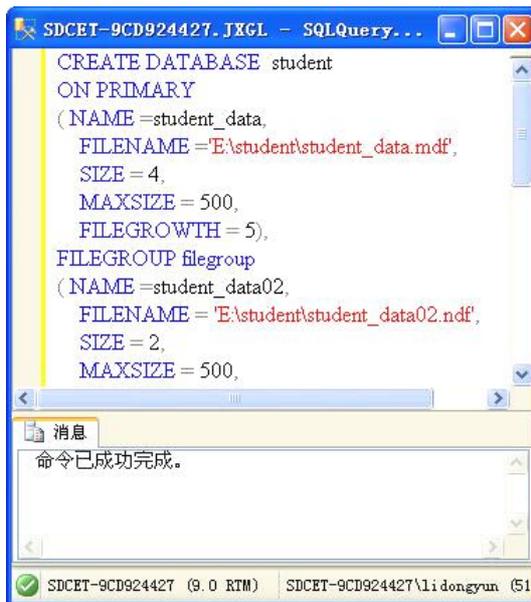


图 2-6 显示运行消息

### 3. 创建数据库示例

**【例 2-1】** 创建未指定文件的数据库 shili01。代码如下：

```
IF DB_ID ('shili01') IS NOT NULL
-- 判断当前服务中是否已有名称为 shili01 的数据库
DROP DATABASE shili01
GO
-- 若已有名称为 shili01 的数据库，则删除它
```

```
CREATE DATABASE shili01
GO
```

分析：本示例没有 <filespec> 语句项。主数据库文件的大小为 model 数据库主文件的大小 1MB，事务日志文件为下列值中较大者：512 KB 或主数据文件大小的 25%。因为没有指定 MAXSIZE，文件可以增大到填满全部可用的磁盘空间为止。

**【例 2-2】** 创建指定数据文件和日志文件的数据库 shili02。代码如下：

```
CREATE DATABASE shili02
ON
( NAME = shili02_data,
  FILENAME = 'D:\shili\shili02_data.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = shili02_log,
  FILENAME = 'D:\shili\shili02_log.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

分析：本示例没有使用关键字 PRIMARY，第一个文件 shili02\_data 将成为主文件。在 shili02\_data.mdf 文件的 SIZE 参数中没有指定单位，将使用 MB 并按 MB 分配空间。

**【例 2-3】** 创建指定多个数据和事务日志文件的数据库 shili03。代码如下：

```
CREATE DATABASE shili03
ON
PRIMARY
( NAME = shili03_data,
  FILENAME = 'D:\shili\shili03_data.mdf',
  SIZE = 2,
  MAXSIZE = 200,
  FILEGROWTH = 2),
( NAME = shili03_data302,
  FILENAME = 'D:\shili\shili03_data302.ndf',
  SIZE = 5,
  MAXSIZE = 500,
  FILEGROWTH = 5%)
LOG ON
( NAME = shili03_log301,
  FILENAME = 'D:\shili\shili03_log301.ldf',
  SIZE = 10,
  MAXSIZE = 200,
  FILEGROWTH = 10%),
( NAME = shili03_log302,
  FILENAME = 'D:\shili\shili03_log302.ldf',
  SIZE = 5,
```

```
MAXSIZE = 1024,  
FILEGROWTH = 2)  
GO
```

分析：本示例要创建的数据库 shili03 具有两个数据文件和两个日志文件。主数据文件使用 PRIMARY 关键字显式指定，日志文件在 LOG ON 关键字后指定。

**【例 2-4】**创建具有文件组的数据库 shili04。代码如下：

```
CREATE DATABASE shili04  
ON PRIMARY  
( NAME =shili04_data,  
  FILENAME = 'd:\shili\shili04_data.mdf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 15% ),  
( NAME =shili04_data402,  
  FILENAME = 'd:\shili\shili04_data402.ndf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 15% ),  
FILEGROUP shiligroup1  
( NAME = shili04_data403,  
  FILENAME = 'd:\shili\shili04_data403.ndf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 5 ),  
( NAME = shili04_data404,  
  FILENAME = 'd:\shili\shili04_data404.ndf',  
  SIZE = 10,  
  MAXSIZE = 50,  
  FILEGROWTH = 5 )  
LOG ON  
( NAME = shili04_log,  
  FILENAME = 'd:\shili\shili04_log.ldf',  
  SIZE = 5MB,  
  MAXSIZE = 25MB,  
  FILEGROWTH = 5MB )  
GO
```

分析：本示例创建的数据库 shili04 包含以下两个文件组：①包含文件 shili04\_data 和 shili04\_data402 的主文件组 PRIMARY；②名为 shiligroup1 的文件组，包含数据文件 shili04\_data403 和 shili04\_data404。

#### 4. 创建“高职院校教学管理系统”数据库 JXGL

**【例 2-5】**创建本教材实例数据库 JXGL。该数据库具有两个文件组：①包含主数据文件 JXGL\_data 的主文件组，文件初始容量为 4MB，递增率为 5MB，最大容量为 500MB；②filegroup1 文件组，包含数据文件 JXGL\_data02，文件初始容量为 2MB，递增率为 10%，最大容量为 500MB。包含两个日志文件 JXGL\_log 和 JXGL\_log02，文件初始容量为 5MB，递增

率为 5%和 5MB，最大容量为 1GB 和 500MB。

代码如下：

```
CREATE DATABASE JXGL
ON PRIMARY
( NAME =JXGL_data,
  FILENAME =D:\JXGL\JXGL_data.mdf,
  SIZE = 4,
  MAXSIZE = 500,
  FILEGROWTH = 5),
FILEGROUP filegroup1
( NAME =JXGL_data02,
  FILENAME = 'D:\JXGL\JXGL_data02.ndf',
  SIZE = 2,
  MAXSIZE = 500,
  FILEGROWTH = 10% )
LOG ON
( NAME = JXGL_log,
  FILENAME = 'D:\JXGL\JXGL_log.ldf',
  SIZE = 5,
  MAXSIZE = 1024,
  FILEGROWTH = 5% ),
( NAME = JXGL_log02,
  FILENAME = 'D:\JXGL\JXGL_log02.ldf',
  SIZE = 5,
  MAXSIZE = 500,
  FILEGROWTH = 5 )
GO
```

创建用户数据库时要注意以下几点：

- (1) 默认情况下，只有系统管理员可以创建数据库。
- (2) 当新的用户数据库创建后，master 数据库的系统表 sysdatabases 立即被更新，因此创建数据库后应及时备份 master 数据库。
- (3) 所有用户数据库都是 model 数据库的副本，数据库的系统表包含在主文件中。model 数据库的数据文件为 1MB，日志文件也是 1MB。这意味着用户数据库的最小容量为 2MB。
- (4) 在一个数据库实例上，最多可创建 32767 个数据库。一个数据库最多可包含 32767 个数据文件和 32767 个日志文件。
- (5) 创建完成新数据库后，想在“对象资源管理器”的数据库列表中看到刚创建的数据库，需要刷新“数据库”节点。

## 2.3 管理 SQL Server 数据库

**任务：**使用 SSMS 及 T-SQL 语言两种方法对数据库 JXGL 进行简单管理。

### 2.3.1 使用 SSMS 管理数据库

#### 1. 查看数据库属性

在“对象资源管理器”中展开“数据库”节点。选择 JXGL 数据库，右击，在弹出的快捷菜单中选择“属性”命令，打开“数据库属性”对话框。

在“数据库属性”对话框的“选择页”列表中，包含“常规”、“文件”、“文件组”、“选项”、“权限”、“扩展属性”、“镜像”和“事务日志传送”等选项卡。单击任意一个选项卡，在窗口右侧的显示区列出数据库的相关属性。如图 2-7 所示为 JXGL 数据库的属性对话框。

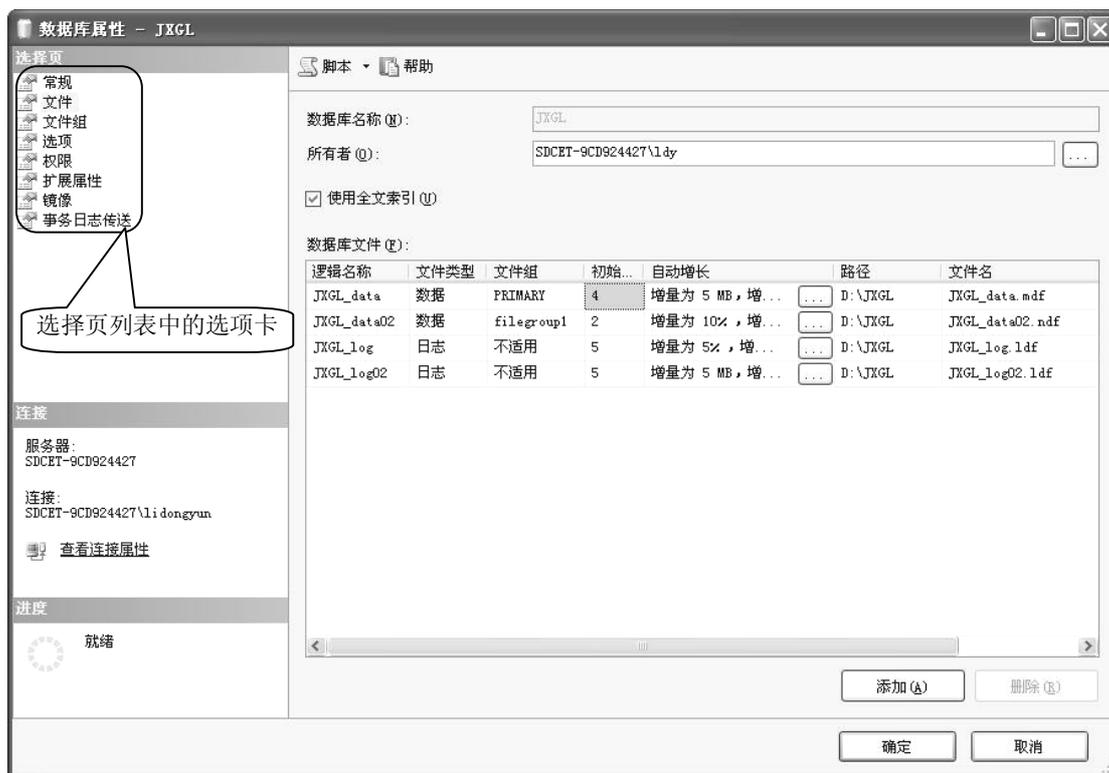


图 2-7 数据库 JXGL 的属性对话框

#### 2. 修改数据库

在“数据库属性”对话框中选择不同的选项，可以对数据库相关属性进行以下修改：

(1) “文件”选项卡窗口：修改数据库所有者，修改数据文件或日志文件的逻辑文件名、初始容量、自动增长属性等，添加新文件或删除原来的文件。但不能修改文件类型、所在文件组和文件路径。

(2) “文件组”选项卡窗口：添加或删除文件组，修改文件组的只读属性等。

(3) “选项”选项卡窗口：设置数据库相关性值。

(4) “权限”选项卡窗口：删除数据库用户，或添加数据库新用户并授权。

(5) “扩展属性”选项卡窗口：指定新的字符排序规则，或添加其他属性。

(6) “镜像”选项卡窗口：设置镜像该数据库过程中所涉及的数据库服务器。

(7) “事务日志传送”选项卡窗口：设置该数据库启用事务日志传送配置中的主数据库。在“权限”选项卡窗口中，数据库选项的含义如下：

(1) “排序规则”：设置数据库字符的排序规则。

(2) “兼容级别”：设置数据库所支持的 MS SQL Server 的最新版本。提供 SQL Server 7.0 (70)、SQL Server 2000 (80) 和 SQL Server 2005 (90) 三种版本。

(3) “恢复模式”：设置数据库恢复模式。提供“完整”、“大容量日志”、“简单”三种模式（将在后续章节中介绍）。

(4) “数据库为只读”：设置数据库是否为只读状态。设置数据库为只读状态时，用户不能修改数据库中的任何数据。

(5) “限制访问”：设置数据库用户数量。提供 single（单用户）、multiple（多用户）和 restricted（限制用户）三种形式。当设置为 restricted 时，只有 db\_owner、db\_creator 和 sysadmin 数据库角色才有权访问该数据库。

(6) “自动关闭”：在最后一个用户退出连接后，数据库是否完全关闭并释放系统资源。

(7) “自动收缩”：设置数据库自动缩减容量（将在下一节介绍）。

### 3. 删除数据库

选择某一数据库如 shili01，右击，在快捷菜单中选择“删除”命令，即可将数据库 shili01 删除。

**提示：**①对数据库执行删除命令，不但数据库将被删除，数据库文件也将从磁盘中被删除。②删除数据库将更新 master 数据库中的系统表。若 master 需要还原，则从上次备份 master 之后删除的所有数据库仍然在系统表中被引用，因而可能导致出错。

### 4. 数据库更名

数据库被创建后一般不要更改名称，否则所有与该数据库的连接都将出错。

若必须重新命名数据库，则首先打开“数据库属性”对话框的“选项”选项卡窗口，在“其他选项”区中展开“状态”下拉列表，选择“限制访问”选项值为 single。此时，在数据库列表中该数据库名称后出现“单个用户”标志。右击数据库名，在快捷菜单中选择“重命名”命令。更名后，须将“限制访问”的值改回 multiple。

### 5. 生成创建数据库脚本

在“新建数据库”窗口的顶部，有一个  按钮，单击其右边的  按钮，弹出快捷菜单，指示要把数据库脚本保存在何处，有四个不同的目的地：新建查询窗口、文件、剪贴板和作业，如图 2-8 所示。

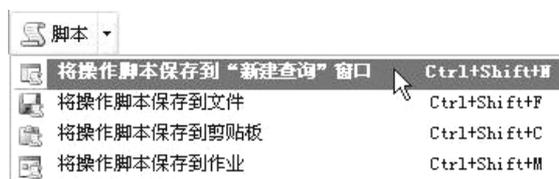


图 2-8 将生成的创建数据库脚本保存到不同位置

对已经存在的数据库，也可以生成创建脚本，并有三种方式供用户选择。选中数据库 JXGL，右击，选择“编写数据库脚本为”→“CREATE 到”，选择下列命令之一：

(1) “新建查询编辑器窗口”命令。将生成的创建数据库脚本写入新的“查询编辑器”窗口。

(2) “文件”命令。将生成的创建数据库脚本以.sql 文件形式保存到指定磁盘上。

(3) “剪贴板”命令。将生成的创建数据库脚本复制到剪贴板。

**提示：**表、视图等数据库对象也可以相同方式生成创建该对象的 SQL 脚本。

### 2.3.2 使用 T-SQL 管理数据库

#### 1. 查看数据库属性

使用系统存储过程 `sp_helpdb` 查看指定数据库或当前所有数据库信息。基本语法如下：

`sp_helpdb [database_name]`

`database_name` 缺省时，查看当前所有数据库信息。

**【例 2-6】**查看数据库实例下所有数据库信息，查看 JXGL 数据库信息。代码如下：

(1) `sp_helpdb --查看数据库实例下所有数据库信息`

运行结果如图 2-9 所示，只显示数据库基本信息。

|    | name            | db_size   | owner                     | dbid | created    | status             |
|----|-----------------|-----------|---------------------------|------|------------|--------------------|
| 1  | JXGL            | 5.00 MB   | SDCET-9CD924427\lidongyun | 7    | 10 31 2008 | Status=ONLINE, Upd |
| 2  | master          | 4.50 MB   | sa                        | 1    | 04 8 2003  | Status=ONLINE, Upd |
| 3  | model           | 1.69 MB   | sa                        | 3    | 04 8 2003  | Status=ONLINE, Upd |
| 4  | msdb            | 5.44 MB   | sa                        | 4    | 10 14 2005 | Status=ONLINE, Upd |
| 5  | mydb            | 18.00 MB  | SDCET-9CD924427\lidongyun | 8    | 10 31 2008 | Status=ONLINE, Upd |
| 6  | ReportServer    | 3.94 MB   | SDCET-9CD924427\lidongyun | 5    | 10 8 2008  | Status=ONLINE, Upd |
| 7  | ReportServer... | 2.94 MB   | SDCET-9CD924427\lidongyun | 6    | 10 8 2008  | Status=ONLINE, Upd |
| 8  | shli01          | 1.68 MB   | SDCET-9CD924427\lidongyun | 9    | 10 30 2008 | Status=ONLINE, Upd |
| 9  | shli02          | 15.00 MB  | SDCET-9CD924427\lidongyun | 10   | 10 30 2008 | Status=ONLINE, Upd |
| 10 | shli03          | 500.00 MB | SDCET-9CD924427\lidongyun | 11   | 10 30 2008 | Status=ONLINE, Upd |
| 11 | shli04          | 65.00 MB  | sa                        | 12   | 10 30 2008 | Status=ONLINE, Upd |
| 12 | tempdb          | 8.75 MB   | sa                        | 2    | 10 31 2008 | Status=ONLINE, Upd |

图 2-9 查看数据库实例下的所有数据库信息

(2) `sp_helpdb JXGL --查看 JXGL 数据库信息`

运行结果如图 2-10 所示，显示数据库及数据库文件基本信息。

|   | name | db_size  | owner                     | dbid | created    | status                                |
|---|------|----------|---------------------------|------|------------|---------------------------------------|
| 1 | JXGL | 16.25 MB | SDCET-9CD924427\lidongyun | 7    | 11 23 2008 | Status=ONLINE, Updateability=READ_WRI |

|   | name        | fileid | filename                | filegroup  | size    | maxsize    | growth  | usage     |
|---|-------------|--------|-------------------------|------------|---------|------------|---------|-----------|
| 1 | JXGL_data   | 1      | D:\JXGL\JXGL_data.mdf   | PRIMARY    | 4096 KB | 512000 KB  | 5120 KB | data only |
| 2 | JXGL_log    | 2      | D:\JXGL\JXGL_log.ldf    | NULL       | 5120 KB | 1048576 KB | 5%      | log only  |
| 3 | JXGL_data02 | 3      | D:\JXGL\JXGL_data02.ndf | filegroup1 | 2048 KB | 512000 KB  | 10%     | data only |
| 4 | JXGL_log02  | 4      | D:\JXGL\JXGL_log02.ldf  | NULL       | 5120 KB | 512000 KB  | 5120 KB | log only  |

图 2-10 查看 JXGL 数据库信息

**提示：**数据库信息中的 status 指当前数据库选项的值；compatibility\_level 表示数据库兼容级别。数据库文件信息中的 usage 表示文件的用法，如数据文件的用法为 data only（仅数据），日志文件的用法为 log only（仅日志）。

## 2. 修改数据库

通过 ALTER DATABASE 语句可以添加、删除数据库文件、文件组；可以更改文件或文件组的属性，如名称、容量等；还可以更改数据库名称；但不能移动数据库的存储位置。基本语法如下：

```
ALTER DATABASE <database_name>
ADD FILE <data file> | ADD FILE <log file>
| REMOVE FILE <logic file_name>
| ADD FILEGROUP <filegroup name> | REMOVE FILE <filegroup name>
| MODIFY FILE <file_name> | MODIFY NAME <new database_name>
| MODIFY FILEGROUP <filegroup_name>
| SET <option>
```

**【例 2-7】**使用 ALTER DATABASE 语句对数据库 JXGL 进行以下修改。添加一个次数据文件 JXGL\_data03，初始大小为 5MB，最大容量为 200MB，增长率为 2%。代码如下：

```
USE JXGL
GO
ALTER DATABASE JXGL
ADD FILE
( NAME=JXGL_data03,
  FILENAME='D:\JXGL\JXGL_data03.ndf',
  SIZE = 5,
  MAXSIZE = 200,
  FILEGROWTH = 2%)
GO
```

代码运行结果如图 2-11 所示。

| 逻辑名称        | 文件类型 | 文件组        | 初始大小... | 自动增长           | 路径      | 文件名             |
|-------------|------|------------|---------|----------------|---------|-----------------|
| JXGL_data   | 数据   | PRIMARY    | 4       | 增量为 5 MB，增长... | D:\JXGL | JXGL_data.mdf   |
| JXGL_data02 | 数据   | filegroup1 | 2       | 增量为 10%，增长...  | D:\JXGL | JXGL_data02.ndf |
| jwgl_data03 | 数据   | PRIMARY    | 5       | 增量为 2%，增长的...  | D:\JXGL | jwgl_data03.ndf |
| JXGL_log    | 日志   | 不适用        | 5       | 增量为 5%，增长的...  | D:\JXGL | JXGL_log.ldf    |
| JXGL_log02  | 日志   | 不适用        | 5       | 增量为 5 MB，增长... | D:\JXGL | JXGL_log02.ldf  |

图 2-11 为 JXGL 数据库添加新数据文件

**【例 2-8】**使用 ALTER DATABASE 语句对数据库 JXGL 进行以下修改：将日志文件 JXGL\_log02 的初始容量由 5MB 改为 10MB，删除数据文件 JXGL\_data03。代码如下：

```
USE JXGL
GO
ALTER DATABASE JXGL
MODIFY FILE
( NAME=JXGL_log02,SIZE = 10MB )
GO
ALTER DATABASE JXGL
```

```
REMOVE FILE JXGL_data03
GO
```

### 3. 移动数据库位置

通过 ALTER DATABASE 语句的 FILENAME 子句指定数据库文件的新位置，可以移动系统数据库文件和用户数据库文件（资源数据库文件除外）。

**【例 2-9】**使用 ALTER DATABASE 语句将数据库 shili02 移动到 D:\shiliyidong。

(1) 移动数据文件位置，代码如下：

```
ALTER DATABASE shili02
MODIFY FILE
(NAME=shili02_data,FILENAME='D:\shiliyidong\shili_data02.mdf')
GO
ALTER DATABASE shili02
MODIFY FILE
(NAME=shili02_log, FILENAME ='D:\shiliyidong\shili_log02.ldf')
GO
```

代码运行结果如图 2-12 所示。

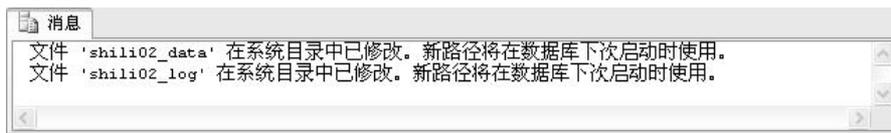


图 2-12 在消息窗格显示移动文件位置成功

(2) 验证数据文件位置已更改，代码如下：

```
SELECT name,physical_name
FROM sys.master_files
WHERE database_id=DB_id('shili02')
GO
```

运行结果如图 2-13 所示。

|   | name         | physical_name                   |
|---|--------------|---------------------------------|
| 1 | shili02_data | d:\shiliyidong\shili_data02.mdf |
| 2 | shili02_log  | d:\shiliyidong\shili_log02.ldf  |

图 2-13 验证移动文件位置成功

**提示：**移动数据库文件，需要知道数据库文件的逻辑名。可以通过查询 sys.master\_files 目录视图中的 name 列获取。不能移动或重命名资源数据库文件，若该文件已重命名或移动，SQL Server 将不启动。

### 4. 删除数据库

使用 DROP DATABASE 语句删除数据库。基本语法如下：

```
DROP DATABASE <database_name>
```

**【例 2-10】**删除不再需要的数据库 shili01。代码如下：

```
DROP DATABASE shili01
```

执行代码后，刷新“数据库”节点，观察发现，数据库 shili01 已不在数据库列表中。

### 5. 数据库更名

使用系统存储过程 sp\_renamedb 可以对数据库进行重命名。基本语法如下：

```
sp_renamedb <old_database_name> ,<new_database_name>
```

如将数据库 shili04 改名为 shili。代码如下：

```
sp_renamedb shili04,shili
```

## 2.4 知识进阶

**任务：**对数据库容量进行理论估算，对数据库实施收缩，节省存储空间。

### 2.4.1 数据库空间的估算

创建数据库是数据库管理员的主要任务之一，在创建数据库时需要为每个文件指明容量。文件容量不能随意指定，必须首先对数据库容量进行理论估算，才能避免浪费磁盘空间，或因空间不足造成数据库容量不够。

许多因素影响数据库的最终容量，如每条记录的大小、记录的数量、表的数量、数据库的增长率等，在估算数据库容量时都必须考虑在内，如下所示：

(1) 记录的大小。每条记录都是由一列或多列数据组成，列的大小决定了记录的大小。

(2) 记录的数量。对不同的表，有的可能长时间没有变化，有的可能经常发生变化。如 JXGL 数据库中的排课表 (Teacher\_class\_course)，每个学期都会有很大变化，应该按数据变化最大时估算记录的数量。一般情况下，应该按数据表所代表的实体或关系的实际数据量估算该表要存储的记录数，而不是凭空想象。

(3) 表的数量。在数据库逻辑设计阶段的一个重要任务，就是确定表的个数。按关系模型的特点和数据库设计的规范化原则确定表的数量。

(4) 索引的大小。索引的大小取决于索引关键列的大小、索引中包含的数据量（记录数）和索引填充因子。索引填充因子用来指定创建索引后，对每个索引的叶节点索引页的填充比例，值是 0%~100%。填充因子设置为 100%时，意味着在索引页中没有闲置空间；而填充因子设置为 50%时，索引页中将有一半的空间是空闲的。填充因子的值越小，索引页上的空闲空间越大，这样可以大大减少在索引增长过程中引起的索引页分裂，但需要的存储空间也会越大。

(5) 索引的数量。每个表中都至少包含一个索引，这些索引都会占用额外的数据库空间。

(6) 事务日志的大小。不同数据库事务日志的大小差异很大，这主要取决于对数据的修改频率和对事务日志的备份频率。日志记录了事务对数据的更新操作，修改数据频率越高，就需要更多的空间存放日志。而如果经常备份事务日志，则事务日志会越小，因为每次备份都会截断事务日志，抽取出未用空间。

(7) 其他数据库对象的大小和数量。除数据表外，数据库还包含很多其他数据库对象，如索引、视图、存储过程等。有些数据库对象（如索引）会占用更多存储空间。应仔细分析并估算所有数据库对象的大小和数量。

(8) 数据库的计划增长率。有很多数据库从创建之初就基本没有数据变化，数据库容量改变不大。而有些数据库经常有大幅度的数据增长，如教学管理数据库，每学年都有新的

学生数据产生,随之还会产生新的排课、成绩数据,使数据库容量大幅增长。此时,应预先对数据库存储需求估算得高一些,因为数据库在创建完成后再进行扩展增容会带来很多不必要的麻烦。

### 2.4.2 增加和收缩数据库容量

SQL Server 采用预先指定、分配存储空间的方法来创建数据库。当数据库的数据增长到要超过定义的使用空间时,必须加大数据库容量,增加数据库容量即为数据库提供额外的存储空间。在“数据库属性”对话框中,选择“文件”选项卡,打开“更改文件自动增长设置”对话框,修改数据文件容量即可。

反之,当分配的空间没有被充分利用时,会造成空间浪费。收缩数据库功能允许对数据库中的每个文件进行收缩,删除已经分配但没有使用的页。不能将整个数据库收缩到比其初始容量还小。

收缩数据库的动作在后台执行,不影响数据库内的用户活动,但不能在备份数据库时收缩数据库,也不在收缩数据库时创建或备份数据库。

#### 1. 使用 T-SQL 语言手工收缩数据库

语法如下:

```
DBCC SHRINKDATABASE | SHRINKFILE  
( database_name | file_name [ , target_percent ]  
[ , TRUNCATEONLY | NOTRUNCATE ] )
```

其中:

- **target\_percent**: 数据库收缩后,数据库文件中剩余的可用空间的百分比值。
- **TRUNCATEONLY**: 将数据库文件中所有分配的未使用空间释放给操作系统。文件空间默认被释放给操作系统。
- **NOTRUNCATE**: 将 DBCC 压缩释放的空间保留给数据库文件,文件的物理空间不更改。

**【例 2-11】**假设数据库 JXGL 的当前可用空间为数据库大小的 75%,则收缩 JXGL,使其可用空间为 40%。代码如下:

```
DBCC SHRINKDATABASE (JXGL, 40)
```

**提示:** DBCC (Data Base Consistency Checker, 数据库控制台命令)是一个实用命令集,用来检查数据库的逻辑及物理一致性。命令分为维护、杂项、性能调节和验证四类,其中 DBCC SHRINKDATABASE | SHRINKFILE 为维护类命令,收缩指定数据库的数据文件或收缩指定数据文件或日志文件。

#### 2. 使用 SSMS 手工收缩数据库

步骤如下:

(1) 启动 SSMS,打开“对象资源管理器”。选中要收缩的数据库 JXGL,右击,在快捷菜单中选择“任务”→“收缩”→“数据库”命令,打开“收缩数据库”对话框。

(2) 在“数据库大小”区查看“当前分配的空间”和“可用空间”,这是决定是否收缩数据库的重要依据。

(3) 在“收缩操作”区的“收缩后文件中的最大可用空间”文本框内设置收缩数据库后

数据文件中最大可用空间的百分比值。若选择“在释放未使用的空间前重新组织文件”单选按钮，则指定在收缩数据库前将已使用的页移动至文件的起始位置。

(4) 单击  按钮，完成收缩数据库操作。

### 3. 自动收缩数据库

有以下两种方法：

(1) 应用 ALTER DATABASE 语句可以将数据库设计为自动收缩。基本语法如下：

```
ALTER DATABASE <database_name>  
SET AUTO_SHRINK on/off
```

其中，on 表示将数据库设置为自动收缩。

(2) 使用 SSMS 设置数据库为自动收缩。选择要收缩的数据库，打开“数据库属性”对话框。在“选择页”列表中单击“选项”选项卡，展开“其他选项”区的“自动”下拉列表，将“自动收缩”选项的值设置为 true。

文件收缩操作始终从文件末端开始反向进行。例如，要将一个 5GB 的文件收缩至 4GB，系统将从文件的最后一个 1GB 开始释放尽可能多的空间。文件中被释放的部分若含有使用过的页，则将这些页重新定位到保留的区域（即文件的开始位置，也即重新组织文件）。只能将数据库收缩到没有剩余的可用空间为止。如 5GB 的数据库已有 4GB 数据（可用空间只有 20%），若指定数据库收缩到 2GB（数据库的 40%），则实际上只会释放 1GB 空间。

## 项目实训二

### 1. 实训题目

创建“销售合同管理系统”数据库 CONTRACTDB。

### 2. 实训目的

- (1) 掌握 SQL Server 数据库的存储结构。
- (2) 掌握通过 SSMS 和 T-SQL 创建数据库的方法。
- (3) 掌握通过 SSMS 和 T-SQL 管理数据库的方法。

### 3. 实训内容

(1) 分析“销售合同管理系统”数据库空间需求，估算数据库初始容量，分析每张表所占的记录。

- 1) 部门信息：按 20 条记录估算。
- 2) 员工信息：按 1000 条记录估算。
- 3) 合同信息、合同应收款信息、合同产品信息、客户信息、联系人信息、服务信息：每年 100 个合同，每个合同对应 20 个合同产品信息，每个合同和其他数据表按一一对应关系，按三年的容量估算。
- 4) 产品信息：按 10000 条记录计算。
- 5) 厂商信息：按 10000 条记录计算。

(2) 使用 SSMS 创建“销售合同管理系统”数据库 CONTRACTDB，并删除数据库。基本操作步骤：

- 1) 在 SSMS 中选择“视图”→“对象资源管理器”命令。

- 2) 展开“数据库”节点，打开“新建数据库”的“常规”选项卡窗口。
- 3) 输入数据库名称 CONTRACTDB。
- 4) 选择数据库所有者。
- 5) 按数据库空间估算值设置各个数据库文件和日志文件。
- 6) 选择数据库 CONTRACTDB，右击，在快捷菜单中选择“删除”命令。

(3) 使用 CREATE DATABASE 语句创建“销售合同管理系统”数据库 CONTRACTDB。

代码如下：

```
CREATE DATABASE CONTRACTDB
ON PRIMARY
( NAME = contratdb_data,
  FILENAME = 'D:\db\contractdb_data.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = contractdb_log,
  FILENAME = 'D:\db\contractdb_log.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

(4) 在 SSMS 中查看“销售合同管理系统”数据库 CONTRACTDB 的当前可用空间，若为数据库大小的 80%，则收缩数据库，使其可用空间为 40%。操作步骤如下：

- 1) 启动 SSMS，选中要收缩的数据库 CONTRACTDB，打开“收缩数据库”对话框。
- 2) 查看“当前分配的空间”和“可用空间”，如果可用空间为数据库大小的 80%，则收缩数据库，使其可用空间为 40%。

## 习题二

### 一、填空题

1. SQL Server 存储数据的基本单位是\_\_\_\_\_，其大小是\_\_\_\_\_，盘区是\_\_\_\_\_个物理上连续的\_\_\_\_\_的集合。
2. SQL Server 2005 数据库文件有两个文件名：\_\_\_\_\_和\_\_\_\_\_。
3. 一个 SQL Server 数据库至少包含两个文件，一个是\_\_\_\_\_文件，一个是\_\_\_\_\_文件。
4. 数据库文件容量的单位默认为\_\_\_\_\_，也可自定义其他单位。
5. \_\_\_\_\_记录了什么时候谁对数据库数据执行了哪些修改操作，好比是数据库的“黑匣子”。

### 二、选择题

1. 用来记录 SQL Server 实例中所有系统级别信息的数据库是（ ）数据库。

- A) master                      B) model                      C) tempdb                      D) jxgl
2. SQL Server 数据库主数据文件的后缀是 ( )。
- A) .sql                              B) .mdf                              C) .ldf                              D) .ndf
3. 删除 SQL Server 数据库的 T-SQL 语句是 ( )。
- A) CREATE DATABASE                              B) DELETE DATABASE  
C) DROP DATABASE                              D) ALTER DATABASE
4. 使用 T-SQL 语言修改数据库时不能完成的操作是 ( )。
- A) 更改数据库名称                              B) 添加、删除、修改数据文件和日志文件  
C) 更改数据库所有者                              D) 移动数据库文件的存储位置
5. CREATE DATABASE 语句的 ( ) 子句用来指定数据库文件的逻辑文件名。
- A) FILEGROUP                              B) NAME  
C) FILENAME                              D) FILEGROWTH

### 三、简答题

1. 一个 SQL Server 数据库包含哪几种文件？每个数据库文件有哪几个文件名？
2. 数据库文件组包含哪些文件？其作用是什么？
3. 数据库一旦创建完成，不要轻易进行更名。如果必须更名，需要先进行哪些设置？
4. 删除数据库前应备份哪个数据库？
5. 收缩数据库的含义是什么？如何收缩数据库？