

# 第6章

## 数据库管理



目录内容  
[返回总目录](#)

- └ 创建数据库
- └ 数据库属性设置
- └ 浏览数据库
- └ 更改数据库
- └ 删除数据库
- └ 压缩数据库
- └ 移动数据库
- └ 数据库维护计划
- └ 系统数据库
- └ 实例数据库



管理数据库及其对象是 *SQL Server* 的主要任务。本章将介绍使用 *SQL Server* 来管理和操作数据库的基本知识。

## 6.1 创建数据库

### 6.1.1 文件与文件组

在 SQL Server 中数据库是由数据库文件和事务日志文件组成的。一个数据库至少应包含一个数据库文件和一个事物日志文件。

#### 1. 数据库文件 (Database File)

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。当有多个数据库文件时，有一个文件被定义为主数据库文件 (Primary Database File)，扩展名为 **mdf**，它用来存储数据库的启动信息和部分或全部数据，一个数据库只能有一个主数据库文件。其它数据库文件被称为次数据库文件 (Secondary Database File)，扩展名为 **ndf**，用来存储主文件没存储的其它数据。采用多个数据库文件来存储数据的优点体现在：

- 数据库文件可以不断扩充，而不受操作系统文件大小的限制；
- 可以将数据库文件存储在不同的硬盘中，这样可以同时对几个硬盘做数据存取，提高了数据处理的效率。对于服务器型的计算机尤为有用。

#### 2. 事务日志文件 (Transaction Log File)

事务日志文件是用来记录数据库更新情况的文件，扩展名为 **ldf**。例如使用 **INSERT**、**UPDATE**、**DELETE** 等对数据库进行更改的操作都会记录在此文件中，而如 **SELECT** 等对数据库内容不会有影响的操作则不会记录在案。一个数据库可以有一个或多个事务日志文件。

SQL Server 中采用“Write-Ahead (提前写)”方式的事务，即对数据库的修改先写入事务日志中，再写入数据库。其具体操作是，系统先将更改操作写入事务日志中，再更改存储在计算机缓存中的数据，为了提高执行效率，此更改不会立即写到硬盘中的数据库，而是由系统以固定 4 的时间间隔执行 **CHECKPOINT** 命令，将更改过的数据批量写入硬盘。SQL Server 有个特点，它在执行数据更改时会设置一个开始点和一个结束点，如果尚未到达结束点就因某种原因使操作中断，则在 SQL Server 重新启动时会自动恢复已修改的数据，使其返回未被修改的状态。由此可见，当数据库破坏时，可以用事务日志恢复数据库内容。

#### 3. 文件组 (File Group)


文件组是将多个数据库文件集合起来形成的一个整体。每个文件组有一个组名。与数据库文件一样，文件组也分为主文件组 (Primary File Group) 和次文件组 (Secondary File Group)。一个文件只能存在于一个文件组中，一个文件组也只能被一个数据库使用。主文件组中包含了所有的系统表。当建立数据库时，主文件组包括主数据库文件和未指定

组的其它文件。在次文件组中可以指定一个缺省文件组，那么在创建数据库对象时如果没有指定将其放在哪一个文件组中，就会将它放在缺省文件组中。如果没有指定缺省文件组，则主文件组为缺省文件组。

! 事务日志文件不属于任何文件组。

## 6.1.2 用 Enterprise Manager 创建数据库

在 Enterprise Manager 中，可以按下列步骤来创建数据库：

- (1) 单击工具栏中的  图标，或在服务器的 **Databases** 文件夹或其下属数据库图标上单击右键，选择 **New Database...** 选项，即会出现如图 6-1 所示的对话框。
- (2) 指定数据库的名称和编码方式，如图 6-1 所示。

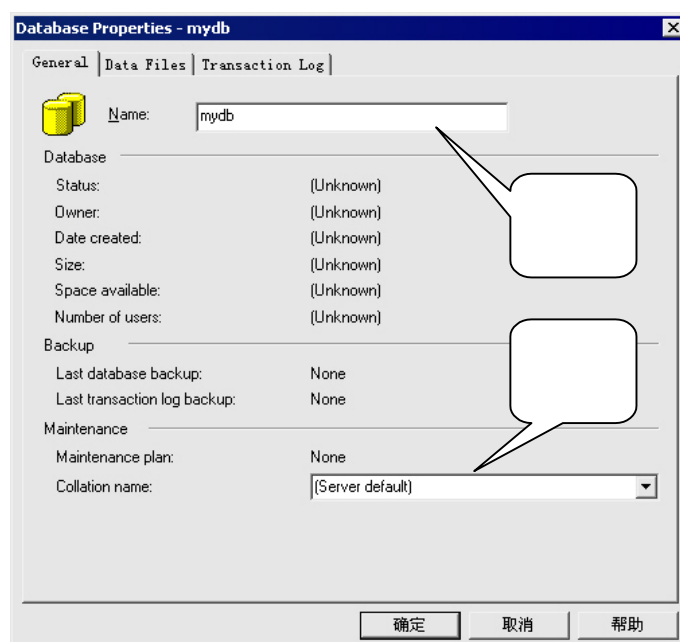


图6-1 创建数据库对话框

- (3) 点击 **Transaction Log** 页框，指定数据库文件的名称、存储位置、初始容量大小和所属文件组，如图 6-2 所示。
- (4) 进行数据库文件大小扩充方式和容量限制设置，如图 6-2 所示。
- (5) 点击 **Transaction Log** 页框，指定事务日志文件的名称、存储位置和初始容量大小，如图 6-3 所示。
- (6) 进行事务日志文件大小扩充方式和容量限制设置，如图 6-3 所示。

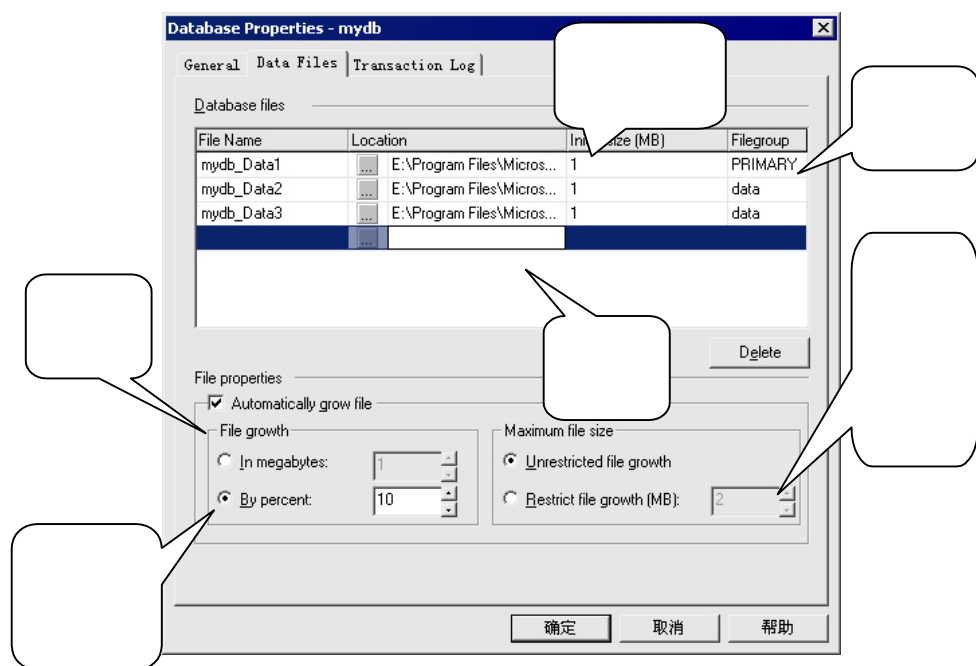


图6-2 创建数据库文件

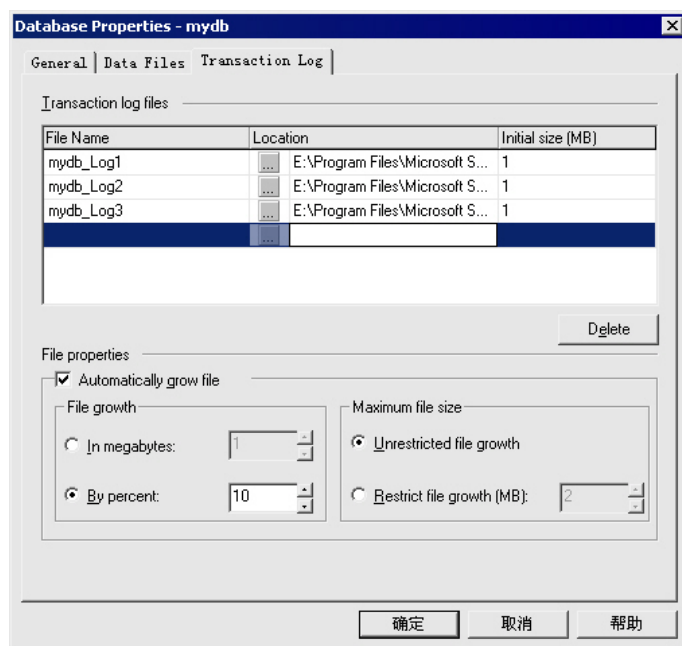


图6-3 创建数据库事务日志文件

(7) 单击“确定”按钮，则建新数据库。

! 数据库的名称最长为 128 个字符，且不区分大小写。

!

在此指定的文件容量是以 MB 为单位的，数据库文件的最小容量为 1MB，最大容量为 1,048,516TB（这么大的硬盘还没出世！）；事务日志文件的最小容量为 1MB。

一个服务器在理论上可以管理 32,767 个数据库。

### 6.1.3 用 CREATE DATABASE 命令创建数据库

CREATE DATABASE 命令语法如下：

**CREATE DATABASE database\_name**

**[ ON [PRIMARY] [ <filespec> [,...n] ] [, <filegroupspec> [,...n] ] ]**  
**[ LOG ON { <filespec> [,...n]} ]**  
**[ COLLATE collation\_name ]**  
**[ FOR LOAD | FOR ATTACH ]**

**<filespec> ::= ( [ NAME = logical\_file\_name, ]**  
**FILENAME = 'os\_file\_name'**  
**[, SIZE = size]**  
**[, MAXSIZE = { max\_size | UNLIMITED } ]**  
**[, FILEGROWTH = growth\_increment] ) [,...n]**

**<filegroupspec> ::= FILEGROUP filegroup\_name <filespec> [,...n]**

各参数说明如下：

- database\_name

数据库名称，不能超过 128 个字符，由于系统会在其后添加 5 个字符的逻辑后缀，因此实际能指定的字符数为 123 个。

- ON

指明数据库文件和文件组的明确定义。

- PRIMARY

指明主数据库文件或主文件组。主文件组的第一个文件被认为是主数据库文件，其中包含了数据库的逻辑启动信息和数据库的系统表。如果没有 PRIMARY 项，则在 CREATE DATABASE 命令中列出的第一个文件将被默认为主文件。

- filespec

文件说明。

- n

占位符，表明可以指定多个类似的对象。

- filegroupspec

文件组说明。

- LOG ON

指明事务日志文件的明确定义。如果没有 LOG ON 选项，则系统会自动产生一个文件名前缀与数据库名相同，容量为所有数据库文件大小 1/4 的事务日志文件。

- COLLATE

指明数据库使用的校验方式。collation\_name 可以是 Windows 的校验方式名称，也可以是 SQL 校验方式名称。如果省略此子句，则数据库使用当前的 SQL Server 设置的校验方式。

- FOR LOAD

此选项是为了与 SQL Server 7.0 以前的版本兼容而设定的。读者可以不用管它。RESTORE 命令可以更好地实现此功能。

- FOR ATTACH

用于附加已经存在的数据库文件到新的数据库中，而不用重新创建数据库文件。使用此命令必须指定主文件。被附加的数据库文件的代码页（Code Page）和排序次序（Sort Order）必须和目前 SQL Server 所使用的一致。建议使用 sp\_attach\_db 系统存储过程来代替此命令（关于 sp\_attach\_db 系统存储过程的用法请参见本章的第 7 节“移动和复制数据库”）。CREATE DATABASE FOR ATTACH 命令只有在指定的文件数目超过 16 个时必须使用。

- NAME

指定文件在 SQL Server 中的逻辑名称。当时用 FOR ATTACH 选项时，就不需要使用 NAME 选项了。

- FILENAME

指定文件在操作系统中存储的路径名和文件名称。

- SIZE

指定数据库的初始容量大小。如果没有指定主文件的大小，则 SQL Server 默认其与模板数据库中的主文件大小一致，其它数据库文件和事务日志文件则默认为 1 MB。指定大小的数字 size 可以使用 KB、MB、GB 和 TB 后缀，默认的后缀是 MB。size 中不能使用小数，其最小值为 512 KB，默认值是 1MB。主文件的 size 不能小于模板数据库中的主文件（关于模板数据库的介绍请参见本章最后一节）。

- MAXSIZE

指定文件的最大容量。如果没有指定 MAXSIZE，则文件可以不断增长直到充满磁盘。

- UNLIMITED

指明文件无容量限制。

- FILEGROWTH

指定文件每次增容时增加的容量大小。增加量可以为确定的以 KB、MB 作后缀的字节数或以%作后缀的被增容文件的百分比来表示。缺省后缀为 MB。如果没有指定 FILEGROWTH，则缺省值为 10%，每次扩容的最小值为 64 KB。

CREATE DATABASE 命令在 SQL Server 中执行时使用模板数据库来初始化新建的数据库（使用 FOR ATTACH 选项时除外）。在模板数据库中的所有用户定义的对象和数据库的设置都会被复制到新数据库中。每个数据库都有一个所有者（Database Owner，简称 DBO），创建数据库的用户被默认为数据库所有者。可以通过 sp\_changedbowner 系统

存储过程来更改数据库所有者。

例 6-1：创建一个与上节中用 Enterprise Manager 创建的数据库类似的数据库。

```
create database mydb
on primary
( name = mydb_data1,
  filename = 'd:\sql data\mydb_data1.mdf',
  size = 1,
  maxsize = unlimited,
  filegrowth = 10% ),
filegroup data2
( name = mydb_data2,
  filename = 'd:\sql data\mydb_data2.ndf',
  size = 2,
  maxsize = 100,
  filegrowth = 1 ),
filegroup data3
( name = mydb_data3,
  filename = 'd:\sql data\mydb_data3.ndf',
  size = 3,
  maxsize = 50,
  filegrowth = 500kb )
log on
( name = 'mydb_log1',
  filename = 'd:\sql data\mydb_log1.ldf',
  size = 1mb,
  maxsize = 25mb,
  filegrowth = 10% ),
( name = 'mydb_log2',
  filename = 'd:\sql data\mydb_log2.ldf',
  size = 1mb,
  maxsize = 10mb,
  filegrowth = 10% ),
( name = 'mydb_log3',
  filename = 'd:\sql data\mydb_log3.ldf',
  size = 1mb,
  maxsize = 5mb,
  filegrowth = 512kb )
```

运行结果如下：

-----

The CREATE DATABASE process is allocating 1.00 MB on disk 'mydb\_Data1'.  
The CREATE DATABASE process is allocating 2.00 MB on disk 'mydb\_Data2'.  
The CREATE DATABASE process is allocating 3.00 MB on disk 'mydb\_Data3'.  
The CREATE DATABASE process is allocating 1.00 MB on disk 'mydb\_Log1'.  
The CREATE DATABASE process is allocating 1.00 MB on disk 'mydb\_Log2'.  
The CREATE DATABASE process is allocating 1.00 MB on disk 'mydb\_Log3'.

## 6.2 数据库属性设置

在创建数据库以后还可以设置其属性，以更改创建时的某些设置和创建时无法设置的属性。用右键单击所要设置属性的数据库，从快捷菜单中选择选项“属性（Properties）”，就会出现如图 6-4 所示的数据库属性设置对话框。

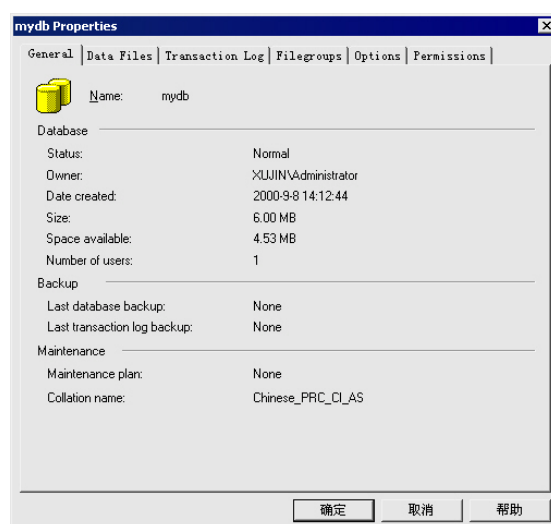


图6-4 数据库属性设置对话框

在“General”页框中，可以看到数据库的状态、所有者、创建时间、容量、备份、维护等属性信息。

在“Data Files”和“Transaction Log”页框中，可以看到类似于图 6-2 和图 6-3 的对话框，此时可以像在创建数据库时那样重新指定数据库文件和事务日志文件的名称、存储位置、初始容量大小等属性。

在“Filegroups”页框中，如图 6-5 所示，可以添加或删除文件组，不过，如果文件组中有文件则不能删除，必须先将文件移出文件组。



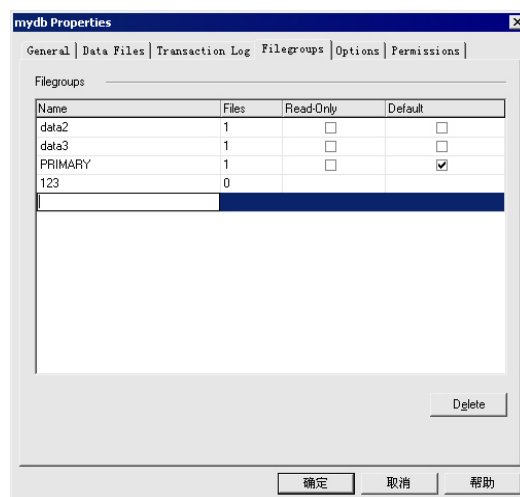


图6-5 文件组设置对话框

在“Options”选项页框中，可以设置数据库的许多属性，如图 6-6 所示。在图 6-6 中选中的选项都是系统默认为 True 的。

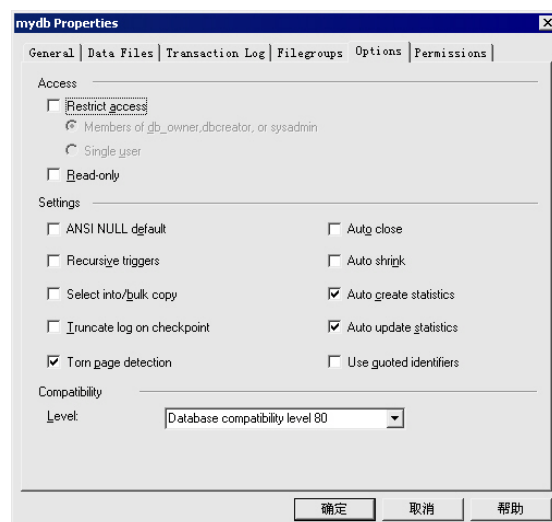


图6-6 数据库选项设置

各选项选中时的含义如下：

- Restrict access

访问限制。

- Members of db\_owner, dbcreator, or sysadmin

只有数据库所有者、数据库创建者和系统管理员才有权使用数据库。

- Single user

数据库在同一时间只能供一个用户使用。

- Read only

数据库只读，即可查看而不可修改。

- **ANSI Null default**

允许在数据库表的列中输入 Null 值。

- **Recursive triggers**

允许触发器递归调用。SQL Server 设定的触发器递归调用的层数上限为 32。

- **Select into / bulk copy**

允许使用 SELECT INTO 或 BCP、WRITETEXT、UPDATETEXT 命令向表中大量插入数据。用上述命令插入数据，在事务日志中不会有记录。在开发数据库时时常将此选项设置为 True。

- **Truncate log on checkpoint**

当执行 CHECKPOINT 命令时，如果事务日志文件超过其大小的 70%，则将其内容清除。在开发数据库时时常将此选项设置为 True。

- **Torn page detection**

自动检测有损坏的页（Torn Page）。页（Page）是数据库内容的基本存储单位。每个页的大小为 8KB。由于 SQL Server 对页的读写单位是大小为 512 字节的扇区（Sector），当每个页的第一个扇区被成功写入后，SQL Server 就认为此页已经被成功写入了。因此，如果写完第一个扇区后发生突发事件，导致写入中断，就会产生有损坏的页。需要通过备份来恢复数据库。

- **Auto close**

当数据库中无用户时，自动关闭此数据库，并将所占资源交还给操作系统。对那些不间断使用的数据库不要使用此选项，否则会额外增加开关数据库的运行开销，得不偿失。

- **Auto shrink**

定期对数据库进行检查，当数据库文件或日志文件的未用空间超过其大小的 25%时，系统将会自动缩减文件使其未用空间等于 25%。当文件大小没有超过其建立时的初始大小时，不会缩减文件。缩减后的文件也必须大于或等于其初始大小。对事务日志文件的缩减只有在对其作备份时或将 Truncate log on checkpoint 选项设为 True 时才能进行。

- **Auto create statistics**

启动数据库列的使用统计，以提供给查询优化器（Query Optimizer）使用。

- **Auto update statistics**

当数据库表更新时，更新所有的统计。

- **Use quoted identifiers**

标识符必须用双引号扩起来，且可以不遵循 Transact-SQL 命名标准。

- **Compatibility Level**

数据库兼容性级别。可以指定与何种版本以前的数据库兼容。

- **其它数据库选项**

还有一些数据库选项没有被显示出来，如：ANSI warning、merge publish、offline、published、subscribed 等，这些选项在一般情况下最好由 SQL Server 自己来控制它，实在需要更改时可以通过 Sp\_dboption 系统存储过程来实现。系统存储过程 Sp\_dboption 用于显示或更改数据库选项。但它不能在系统数据库 master 或 tempdb 数据库中使用。其语法

如下:

```
sp_dboption [[@dbname =] 'database_name']  
            [, [@optname =] 'option_name']  
            [, [@optvalue =] 'value']
```

各参数说明如下:

- [ @dbname =] 'database'

指定所要操作的数据库名。

- [ @optname =] 'option\_name'

指定要显示或修改的数据库选项。可以不输入选项的全名。如果省略此项, 就会列出所有设定为 True 的选项。

- [ @optvalue =] 'value'

指定选项的值。“value” 的值可以为 True、False 或 On、Off。

例 6-2: 显示 mytest 数据库的选项状态。

```
exec sp_dboption 'mytest'
```

运行结果如下:

The following options are set:

-----

select into/bulkcopy

trunc. log on chkpt.

autoshrink

auto create statistics

auto update statistics

例 6-3: 设置 mytest 数据库的 offline 状态为 True, 让一般用户无法使用此数据库。

```
exec sp_dboption 'mytest', 'offline', 'true'
```

运行结果如下:

sp\_dboption command succeeded.



系统管理员可以在模板数据库 model 中执行 Sp\_dboplin 命令来设置数据库选项, 这将改变所有新建数据库的初始选项设置。

在“Permissions”页框中可以设定用户对此数据库的权限, 如图 6-7 所示。有关权限设置的详细介绍请参见“安全性与用户管理”章节。

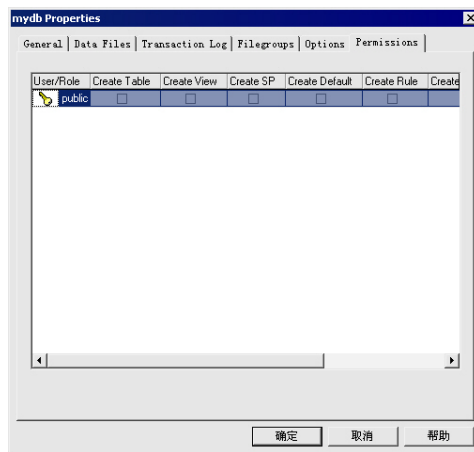


图6-7 数据库权限设置

## 6.3 浏览数据库

### 6.3.1 用 Enterprise Manager 浏览数据库

SQL Server 提供了目录树的浏览方式，使得浏览数据库信息非常方便、快捷。在 Enterprise Manager 中单击要浏览的数据库文件夹，就可在右边的任务板（Task pad）窗口中看到数据库的基本信息（General）、表和索引信息（Tables & Indexes）、数据库文件的配置情况（Space Allocated），如图 6-8 所示。在打开数据库文件夹目录树后，可以选择各种数据库对象进行信息浏览。

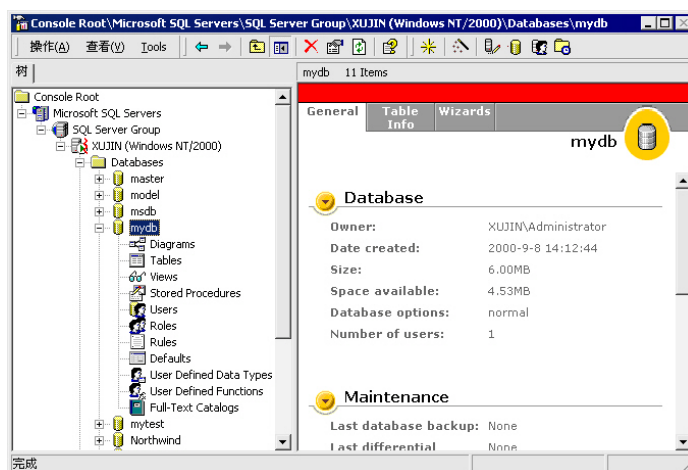


图6-8 浏览数据库

## 6.3.2 用系统存储过程显示数据库信息

SQL Server 提供了许多很有用的系统存储过程，可以用它们来得到许多从 Enterprise Manager 界面中所不易或不能看到的信息（有关存储过程的详细介绍请参见“存储过程”章节。读者如果目前不了解存储过程也不要紧，可以把它当作函数或命令来用）。

### 1. 用系统存储过程显示数据库结构

可以使用系统提供的系统存储过程 Sp\_helpdb 来显示数据库结构。

其语法如下：

**sp\_helpdb** [[@dbname=] 'name']

使用 Sp\_helpdb 系统存储过程可以显示指定数据库的信息。如果不指定[@dbname='name']子句，则会显示在 master.dbo.sysdatabases 表中存储的所有数据库信息。命令执行成功会返回 0，否则返回 1。

例 6-4：显示上一节中创建的 mydb 数据库的信息。

exec sp\_helpdb mydb

运行结果如下：

name	db_size	owner	dbid	created	status	compatibility_level
mydb	9.00MB	XUJIN\Administrator	9	Sep 8 2000	Status=ONLINE,...	80

name	fileid	filename	filegroup	size	maxsize	growth	usage
mydb_Data1	1	d:\SQL Data\mydb_Data1.mdf	PRIMARY	1024KB	Unlimited	10%	data only
mydb_Log1	2	d:\SQL Data\mydb_Log1.ldf	NULL	1024KB	25600KB	10%	log only
mydb_Data2	3	d:\SQL Data\mydb_Data2.ndf	data2	2048KB	102400KB	1024KB	data only
mydb_Data3	4	d:\SQL Data\mydb_Data3.ndf	data3	3072KB	51200KB	512KB	data only
mydb_Log2	5	d:\SQL Data\mydb_Log2.ldf	NULL	1024KB	10240KB	10%	log only
mydb_Log3	6	d:\SQL Data\mydb_Log3.ldf	NULL	1024KB	5120KB	512KB	log only

### 2. 用系统存储过程显示文件信息

可以使用系统提供的系统存储过程 Sp\_helpfile 来显示当前数据库中的文件信息。

其语法如下：

**sp\_helpfile** [[@filename =] 'name']

如果不指定文件名称，则会显示当前数据库中所有的文件信息。命令执行成功会返回 0，否则返回 1。

例 6-5：显示 mydb 数据库中的 mydb\_data1 文件的信息。

```
use mydb
exec sp_helpfile mydb_data1
```

运行结果如下：

name	filename	filegroup	size	maxsize	growth	usage
mydb_Data1	d:\SQLData\mydb_Data1.mdf	PRIMARY	1024KB	Unlimited	10%	data only

### 3. 用系统存储过程显示文件组信息

可以使用系统提供的系统存储过程 Sp\_helpfilegroup 来显示当前数据库中文件组信息。

其语法如下：

**sp\_helpfilegroup** [[@filegroupname =] 'name']

如果不指定文件组名称，则会显示当前数据库中所有的文件组。命令执行成功会返回 0，否则返回 1。

例 6-6：显示 mydb 数据库中的 data2 文件组的信息。

```
use mydb
exec sp_helpfilegroup data2
```

运行结果如下：

groupname	groupid	filecount
data2	2	1

file_in_group	fileid	filename	size	maxsize	growth
mydb_Data2	3	d:\SQLData\mydb_Data2.ndf	2048KB	102400KB	1024KB

例 6-7：显示 mydb 数据库中的所有文件组。

```
use mydb
exec sp_helpfilegroup
```

运行结果如下：

groupname	groupid	filecount
PRIMARY	1	1
data2	2	1
data3	3	1

## 6.4 更改数据库

### 6.4.1 更改文件及其属性

可以在 Enterprise Manager 中利用数据库属性设置更改数据库文件和事务日志文件，也可以用 ALTER DATABASE 命令来更改数据库。ALTER DATABASE 命令可以增加或删除数据库中的文件，也可以修改文件的属性。应注意的是，只有数据库管理员（Database Administration 简称 DBA）或具有 CREATE DATABASE 权限的数据库所有者才有权执行此命令。

ALTER DATABASE 命令的语法如下：

**ALTER DATABASE databasename**

```
{ ADD FILE <filespec> [...n] [TO FILEGROUP filegroup_name]
| ADD LOG FILE <filespec> [...n]
| REMOVE FILE logical_file_name [WITH DELETE]
| ADD FILEGROUP filegroup_name
| REMOVE FILEGROUP filegroup_name
| MODIFY FILE <filespec>
| MODIFY NAME = new_dbname
| MODIFY FILEGROUP filegroup_name
  {filegroup_property | NAME = new_filegroup_name }
| SET < optionspec > [ ,...n ] [ WITH < termination > ]
| COLLATE < collation_name > }
```

```
<filespec> ::= ( NAME = logical_file_name
                [ , NEWNAME = new_logical_name ]
                [ , FILENAME = 'os_file_name' ]
                [ , SIZE = size ]
                [ , MAXSIZE = { max_size | UNLIMITED } ]
                [ , FILEGROWTH = growth_increment ] )
```

```
< optionspec > ::= < state_option >
                  | < cursor_option >
                  | < auto_option >
                  | < sql_option >
                  | < recovery_option >
```

各参数说明如下：

- ADD FILE

指定要增加的数据库文件。

- **TO FILEGROUP**

指定要增加文件到哪个文件组。

- **ADD LOG FILE**

指定要增加的事务日志文件。

- **REMOVE FILE**

从数据库系统表中删除指定文件的定义，并且删除其物理文件。文件只有为空时才能被删除。

- **ADD FILEGROUP**

指定要增加的文件组。

- **REMOVE FILEGROUP**

从数据库中删除指定文件组的定义，并且删除其包括的所有数据库文件。文件组只有为空时才能被删除。

- **MODIFY FILE**

修改指定文件的文件名、容量大小、最大容量、文件增容方式等属性。但一次只能修改一个文件的一个属性。使用此选项时应注意，在文件格式 `filespec` 中必须用 **NAME** 明确指定文件的名称；如果文件大小是已经确定了的，那么新定义的 **size** 必须比当前的文件容量大；**FILENAME** 只能指定在 `tempdb database` 中存在的文件，并且新的文件名只有在 `SQL Server` 重新启动后才发生作用。

- **MODIFY FILEGROUP filegroup\_name filegroup\_property**

修改文件组属性 `filegroup_property`。`filegroup_property` 可取值为：

- ◆**READONLY**：指定文件组为只读。主文件组（Primary File Group）不能指定为只读。只有对数据库有独占访问权限的用户才可以将一个文件组标志为只读。
- ◆**READWRITE**：使文件组为可读写。只有对数据库有独占访问权限的用户才可以将一个文件组标志为可读写。
- ◆**DEFAULT**：指定文件组为默认文件组。一个数据库中只能有一个默认文件组。

- **SET**

设置数据库属性。

- **<state\_option>**

控制用户对数据库访问的属性选项，如 `SINGLE_USER` | `RESTRICTED_USER` | `MULTI_USER`、`OFFLINE` | `ONLINE`、`READ_ONLY` | `READ_WRITE` 等。

- **<cursor\_option>**

控制游标的属性选项，如 `CURSOR_CLOSE_ON_COMMIT ON` | `OFF`、`CURSOR_DEFAULTLOCAL` | `GLOBAL` 等。

- **<auto\_option>**

控制数据库的自动属性选项，如 `AUTO_CLOSE ON` | `OFF`、`AUTO_CREATE_STATISTICS ON` | `OFF`、`AUTO_SHRINK ON` | `OFF`、`AUTO_UPDATE_STATISTICS ON` | `OFF` 等。

- **<sql\_option>**

控制 ANSI 一致性的属性选项，如 `ANSI_NULL_DEFAULT ON` | `OFF`、`ANSI_NULLS ON` | `OFF`、`ANSI_PADDING ON` | `OFF`、`RECURSIVE_TRIGGERS ON` | `OFF` 等。



- <recovery\_options>

控制数据库恢复的选项，如 RECOVERY FULL | BULK\_LOGGED | SIMPLE、TORN\_PAGE\_DETECTION ON | OFF 等。

其余的参数，如：NAME、FILENAME、SIZE、MAXSIZE、UNLIMITED、FILEGROWTH 等与 CREATE DATABASE 命令中的含义相同。个别没讲到的参数将在后面的章节中介绍。

例 6-8：添加一个包含两个文件的文件组到 mydb 数据库中去。并将此文件组指定为默认文件组。

```
alter database mydb
add filegroup data4
/* 要首先定义文件组，然后才能添加文件到文件组中 */
alter database mydb
add file
( name = mydb_data4,
  filename = 'd:\sql data\mydb_data4.ndf',
  size = 2mb,
  maxsize = 100mb,
  filegrowth = 5%),
( name = mydb_data5,
  filename = 'd:\sql data\mydb_data5.ndf',
  size = 2mb,
  maxsize = 50mb,
  filegrowth = 5%)
to filegroup data4
alter database mydb
modify filegroup data4 default
```

运行结果如下：

Extending database by 2.00 mb on disk 'mydb\_data4'.

Extending database by 2.00 mb on disk 'mydb\_data5'.

The filegroup property 'default' has been set.

例 6-9：添加两个事务日志文件到 mydb 数据库中去。

```
alter database mydb
add log file
( name = mydb_log4,
  filename = 'd:\sql data\mydb_log4.ndf',
  size = 1mb,
  maxsize = 50mb,
```

```
filegrowth = 5%),  
( name = mydb_log5,  
  filename = 'd:\sql data\mydb_log5.ndf',  
  size = 1mb,  
  maxsize = 50mb,  
  filegrowth = 5%)go
```

运行结果如下：

Extending database by 1.00 MB on disk 'mydb\_Log4'.

Extending database by 1.00 MB on disk 'mydb\_Log5'.

例 6-10：修改 mydb 数据库中的 mydb\_Data5 文件增容方式为一次增加 2MB。

```
alter database mydb  
modify file  
( name = mydb_log5,  
  filegrowth = 2mb)
```

运行结果如下：

The command(s) completed successfully.

例 6-11：删除 mydb 数据库中的事务日志文件 mydb\_Log5。

```
alter database mydb  
remove file mydb_log5
```

运行结果如下：

The file 'mydb\_Log5' has been removed.

## 6.4.2 更改数据库名

重命名数据库需要使用系统存储过程 Sp\_renamedb，其语法如下：

**sp\_renamedb [@old\_name =] 'old\_name', [@new\_name =] 'new\_name'**


例 6-12：更改 mydb 数据库的名称为 “My DB”。

```
exec sp_renamedb 'mydb', 'My DB'
```

运行结果如下：


The database name ' My DB ' has been set.

！  
在使用此命令前，应将要改名的数据库的访问选项设为 Single user mode，并关闭该数据库。在更改数据库名称后，在 Enterprise Manager 中所看到的仍然是原来的数据库名称，

只有选择“Databases”目录文件夹，再单击工具栏中的  按钮，刷新界面或在 SQL Server 重新启动后才会看到更改后的数据库名。另外还应注意的是，数据库名不区分大小写。

## 6.5 删除数据库

### 6.5.1 用 Enterprise Manager 删除数据库

在 Enterprise Manager 中在所要删除的数据库上单击右键，从快捷菜单中选择“删除（Delete）”选项即可删除数据库，也可以选择数据库文件夹或图标后从工具栏中选择图标  来删除数据库。系统会提示确认是否要删除数据库，如图 6-9 所示。

删除数据库一定要慎重，因为删除数据库后，与此数据库关联的数据库文件和事务日志文件都会被删除，存储在系统数据库中的关于该数据库的所有信息也会被删除。

！ 当数据库处于以下状态时不能被删除：数据库正在使用；数据库正在被恢复；数据库包含用于复制的已经出版的对象。

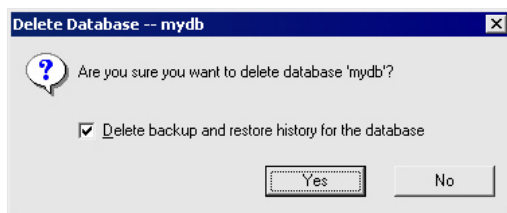


图6-9 确认删除数据库对话框

### 6.5.2 用 DROP DATABASE 命令删除数据库

DROP DATABASE 命令可以从 SQL Server 中一次删除一个或几个数据库。数据库所有者 DBO 和数据库管理员 DBA 才有权执行此命令。其语法如下：

**DROP DATABASE database\_name [...n]**

例 6-13：删除数据库 abc。

drop database abc

运行结果如下：

Deleting database file 'E:\Program Files\Microsoft SQL Server\MSSQL\data\abc\_Log.LDF'.

Deleting database file 'E:\Program Files\Microsoft SQL Server\MSSQL\data\abc\_Data.MDF'.

## 6.6 压缩数据库

数据库在使用一段时间后，时常会出现因数据删除而造成数据库中空闲空间太多的情况，这时就需要减少分配给数据库文件和事务日志文件的磁盘空间，以免浪费磁盘空间。当数据库中没有数据时，可以修改数据库文件属性直接改变其占用空间，但当数据库中有数据时，这样做会破坏数据库中的数据，因此需要使用压缩的方式来缩减数据库空间。可以在数据库属性选项中选择“Auto shrink”选项，让系统自动压缩数据库，也可以用人工的方法来压缩。人工压缩数据库有以下两种方式：

### 6.6.1 用 Enterprise Manager 压缩数据库

在 Enterprise Manager 中在所要压缩的数据库上单击右键，从快捷菜单中的“所有任务（All Tasks）”中选择“Shrink Database（压缩数据库）”选项，就会出现如图 6-10 所示的对话框。可以在图 6-10 所示的对话框中选择数据库的压缩方式，也可以选择使用压缩计划或压缩单个文件。

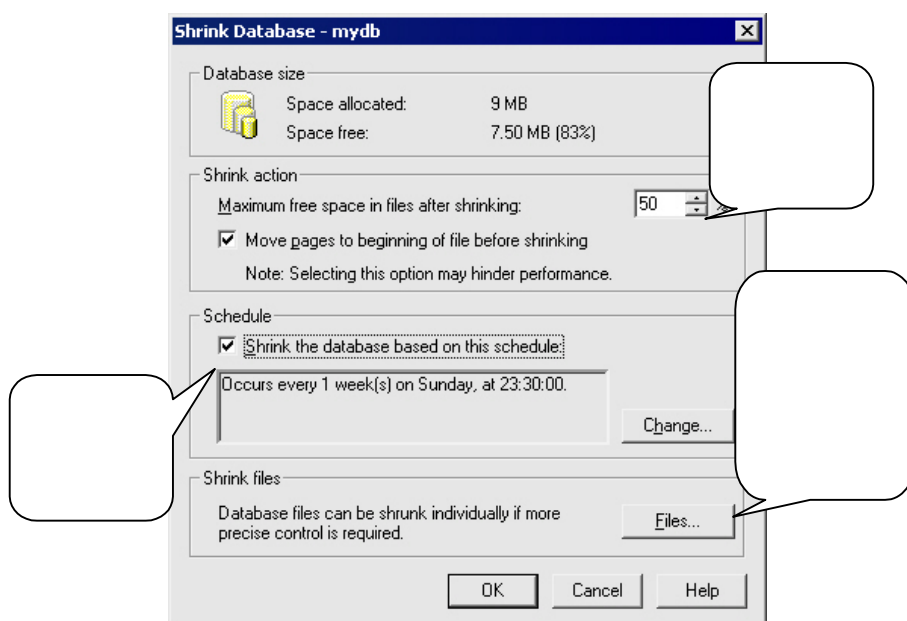


图6-10 压缩数据库对话框

单击图 6-10 中的“Files”按钮，会出现如图 6-11 所示的压缩数据库文件对话框，可以针对每个数据库文件进行不同的压缩设置。

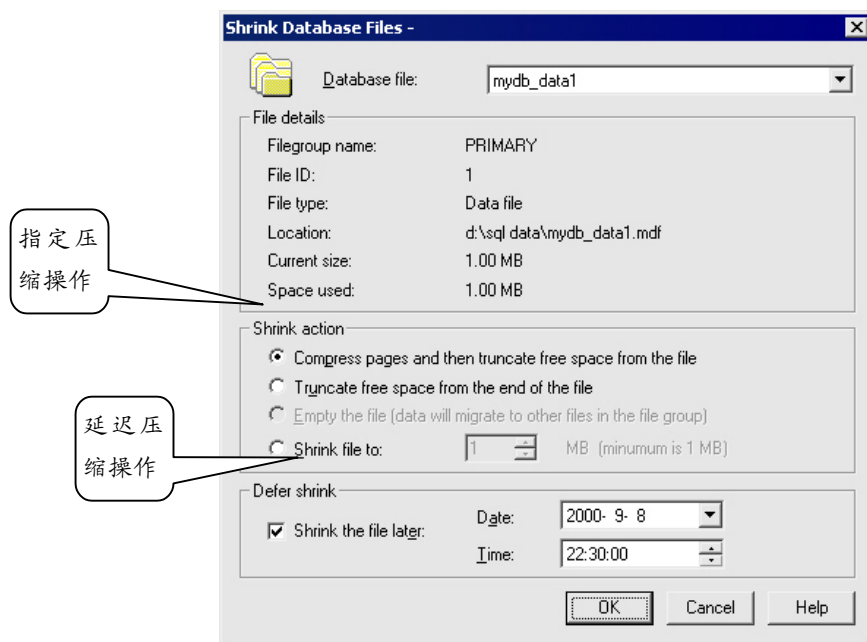


图6-11 压缩数据库文件对话框

单击图 6-10 中的“Change”按钮，会出现如图 6-12 所示的压缩计划编辑对话框，可以指定压缩计划的执行方式。单击图 6-12 中的“Change”按钮，会出现如图 6-13 所示的循环工作计划编辑对话框，可以编辑计划执行的周期或时间点。设置完成后单击“OK”按钮就开始压缩数据库，在压缩结束后会显示一个压缩情况信息框。

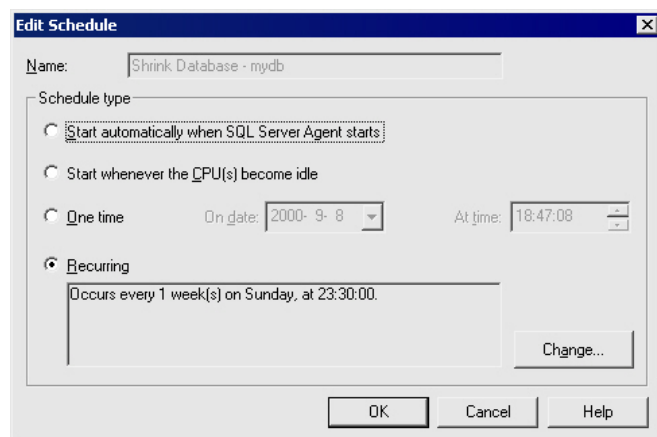


图6-12 压缩计划编辑对话框

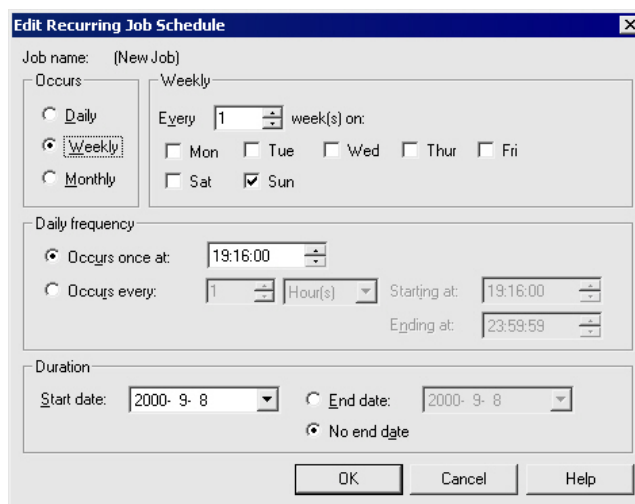


图6-13 循环工作计划编辑对话框

## 6.6.2 用 Transact-SQL 命令压缩数据库

可以使用 DBCC SHRINKDATABASE 和 DBCC SHRINKFILE 命令来压缩数据库。其中 DBCC SHRINKDATABASE 命令对数据库进行压缩，DBCC SHRINKFILE 命令对数据库中指定的文件进行压缩。

### 1. DBCC SHRINKDATABASE

DBCC SHRINKDATABASE 命令语法如下：

**DBCC SHRINKDATABASE (database\_name [, target\_percent]  
[, {NOTRUNCATE | TRUNCATEONLY}] )**

各参数说明如下：

- **target\_percent**

指定将数据库压缩后，未使用的空间占数据库大小的百分之几。如果指定的百分比过大，超过了压缩前未使用空间所占的比例，则数据库不会被压缩。并且压缩后的数据库不能比数据库初始设定的容量小。

- **NOTRUNCATE**

将数据库缩减后剩余的空间保留在数据库中，不返还给操作系统。如果不选择此选项，则剩余的空间返还给操作系统。

- **TRUNCATEONLY**

将数据库缩减后剩余的空间返还给操作系统。使用此命令时 SQL Server 将文件缩减到最后一个文件分配区域，但不移动任何数据文件。选择此项后，target\_percent 选项就无效了。

例 6-14：压缩数据库 mytest 的未使用空间为数据库大小的 20%。

```
dbcc shrinkdatabase (mytest, 20)
```

运行结果如下：

DBCC execution completed. If DBCC printed error messages, contact your system administrator.

## 2. DBCC SHRINKFILE

DBCC SHRINKFILE 命令压缩当前数据库中的文件。其语法如下：

```
DBCC SHRINKFILE ( {file_name | file_id }  
                  { [, target_size] |  
                  [, {EMPTYFILE | NOTRUNCATE | TRUNCATEONLY} ] } )
```

各参数说明如下：

- file\_id

指定要压缩的文件的鉴别号（Identification number，即 ID）。文件的 ID 号可以通过 FILE\_ID（）函数或如本章前面所讲述的 Sp\_helpdb 系统存储过程来得到。

- target\_size

指定文件压缩后的大小。以 MB 为单位。如果不指定此选项，SQL Server 就会尽最大可能地缩减文件。

- EMPTYFILE

指明此文件不再使用，将移动所有在此文件中的数据到同一文件组中的其它文件中。执行带此参数的命令后，此文件就可以用 ALTER DATABASE 命令来删除了。

其余参数 NOTRUNCATE 和 TRUNCATEONLY 与 DBCC SHRINKDATABASE 命令中的含义相同。

例 6-15：压缩数据库 mydb 中的数据库文件 mydb\_data2 的大小到 1MB。

```
use mydb  
dbcc shrinkfile (mydb_data2, 1)
```

运行结果如下：

DbId	FileId	CurrentSize	MinimumSize	UsedPages	EstimatedPages
9	3	128	128	0	0

(1 row(s) affected)

DBCC execution completed. If DBCC printed error messages, contact your system administrator.

## 6.7 移动数据库

在 SQL Server 中可以使用拆分（Detach）和附加（Attach）的方法来移动数据库。拆分数据库是从服务器中移去逻辑数据库，但不会将操作系统中的数据库文件删除。附加数据库将会创建一个新的数据库，并复制存储在已有的数据库文件和事务日志文件中的数据。使用系统存储过程 Sp\_detach\_db 来拆分数据库，用系统存储过程 Sp\_attach\_db 来附加数据库。

Sp\_detach\_db 系统存储过程的语法如下：

```
sp_detach_db [@dbname=] 'database_name'
            [, [@skipchecks=] 'skipchecks']
```

其中[@skipchecks=] 'skipchecks'子句中 Skipchecks 的值为 True 或 False。当 Skipchecks 的值为 True 时,指定在执行此过程之前不需要对数据库中的所有表执行 UPDATE STATISTICS 命令;为 False 时,则需要执行 UPDATE STATISTICS 命令。

Sp\_attach\_db 系统存储过程的语法如下:

```
sp_attach_db [@dbname=] 'dbname',
            [@filename1=] 'filename_n' [...16]
```

其中“filename\_n”包括文件的路径和物理名称。最多可指定 16 个文件。文件中必须包含主数据库文件。如果需要附加的文件超过了 16 个,就必须使用带 FOR ATTACH 子句的 CREATE DATABASE 命令来代替。

! Sp\_attach\_db 系统存储过程只能作用于那些已经用 Sp\_detach\_db 系统存储过程从服务器中拆分出来的数据库。

例 6-16: 移动数据库 mytest 到 E:\SQL Data 目录下。

(1) 在 SQL Server Query Analyzer 中运行系统存储过程 Sp\_helpdb, 得到 mytest 数据库所包含的文件名称、数量、类型、存放位置等信息。命令语句如下:

```
exec sp_helpdb mytest
```

运行结果如下:

name	db_size	owner	dbid	created	status	compatibility_level
mytest	6.00MB	XUJIN\Administrator	10	Jun 8 2000	Status=ONLINE,...	80

name	fileid	filename	filegroup	size	maxsize	growth	usage
mytest_Data	1	D:\SQL Data\mytest_Data.MDF	PRIMARY	3072KB	Unlimited	10%	data only
mytest_Log	2	D:\SQL Data\mytest_Log.LDF	NULL	3072KB	Unlimited	10%	log only

(2) 在 SQL Server Query Analyzer 中运行 Sp\_detach\_db 系统存储过程, 拆分 mytest 数据库。命令语句如下:

```
exec sp_detach_db mytest, true
```

运行结果如下:

Successfully detached database 'mytest1'.

DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(3) 在操作系统的文件夹窗口中用剪切或粘贴的方式直接将与数据库相关的操作系统文件移动到 E:\ SQL Data 2000 Server 目录下。即将 D:\SQL Data\mytest\_



**Data.MDF** 文件和 **D:\SQLData\mytest\_Log.LDF** 文件移到 **E:\ SQL Data 2000 Server** 目录下。

(4) 在 **SQL Server Query Analyzer** 中运行 **Sp\_attach\_db** 系统存储过程，附加文件到 **mytest** 数据库。命令语句如下：

```
exec sp_attach_db @dbname = 'mytest',  
@filename1 = 'e:\sql data\mytest_data.mdf',  
@filename2 = 'e:\sql data\mytest_log.ldf'
```

运行结果如下：

Successfully attached database 'mytest'.

至此已完成了数据库的移动工作，可在 **Enterprise Manager** 中查看 **mytest** 数据库的信息，也可以在 **SQL Server Query Analyzer** 中运行系统存储过程 **Sp\_helpdb** 来查看移动是否成功。

## 6.8 数据库维护计划

数据库维护可以使数据库保持运行在最佳状态。创建数据库维护计划可以让 **SQL Server** 有效地自动维护数据库，为管理员节省不少时间，也可以防止延误数据库的维护工作。创建数据库维护计划的过程如下：

(1) 在 **Enterprise Manager** 中选择要创建维护计划的数据库，单击右键，从快捷菜单中选择“所有任务 (All Tasks)”子菜单中的“**Maintenance Plan (维护计划)**”菜单项，则出现如图 6-14 所示的欢迎界面。



图6-14 数据库维护计划向导

(2) 单击“下一步”按钮，出现如图 6-15 所示的选择数据库对话框。可以根据需要选择所要维护的数据库。

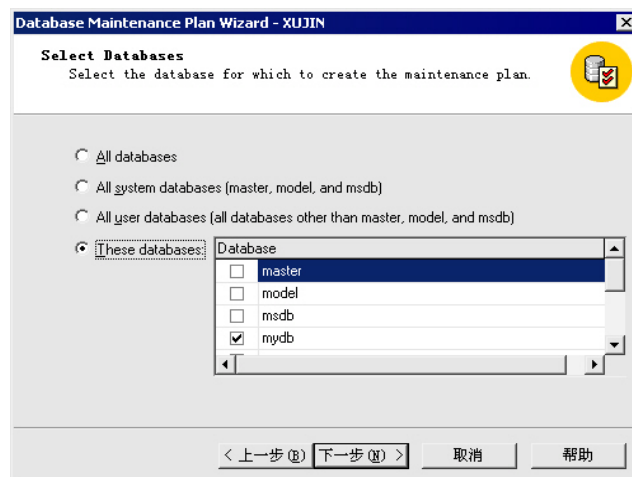


图6-15 选择要维护的数据库对话框

- (3) 单击“下一步”按钮，出现如图 6-16 所示的更新数据最优化信息对话框。可以根据需要选择是否要对数据库的数据页或索引页做重整工作。单击“Change”按钮，出现如图 6-13 所示的编辑循环工作计划对话框。
- (4) 在图 6-16 中单击“下一步”按钮，出现如图 6-17 所示的数据库完整性检查对话框。可以选择是否要对数据库做完整性测试，以便查找是否有因软件或硬件错误所造成的数据库错误。

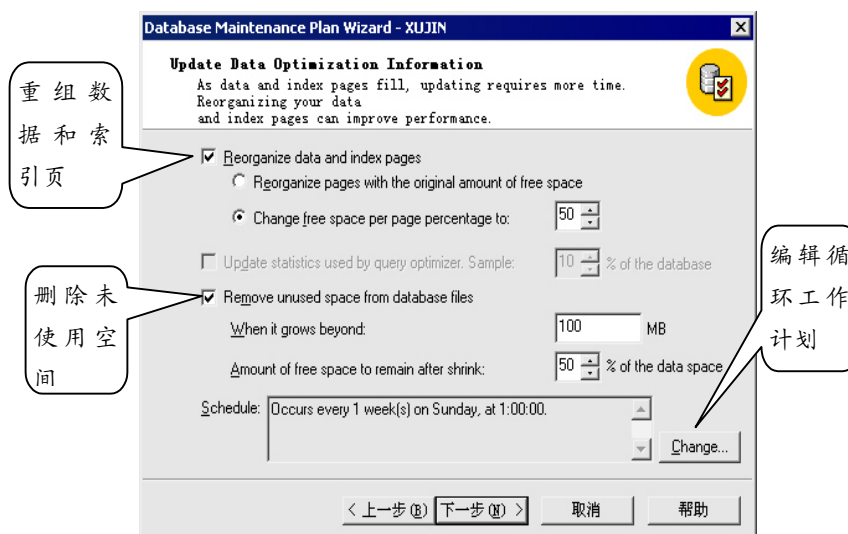


图6-16 更新数据最优化信息对话框

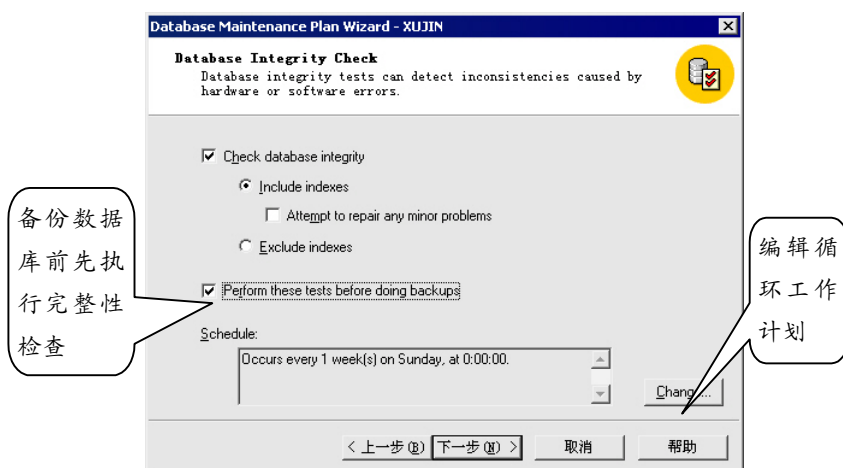


图6-17 数据库完整性检查对话框

(5) 单击“下一步”按钮，出现如图 6-18 所示的指定数据库备份计划对话框。可以选择是否将数据库备份到磁带或硬盘。一般都选择将数据库备份到硬盘。

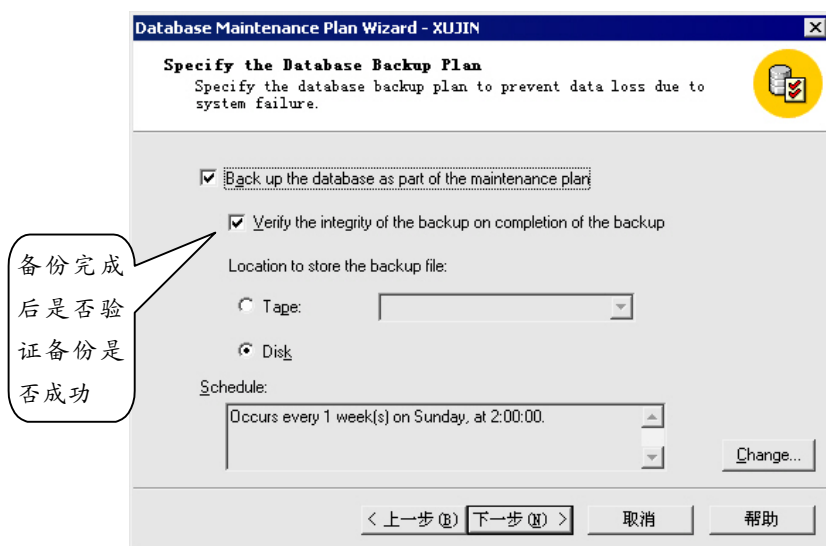


图6-18 指定数据库备份计划对话框

(6) 单击“下一步”按钮，出现如图 6-19 所示的指定数据库备份文件存放目录对话框。可以选择将数据库备份文件存放到指定的硬盘目录，还可以给每个数据库创建一个子目录。

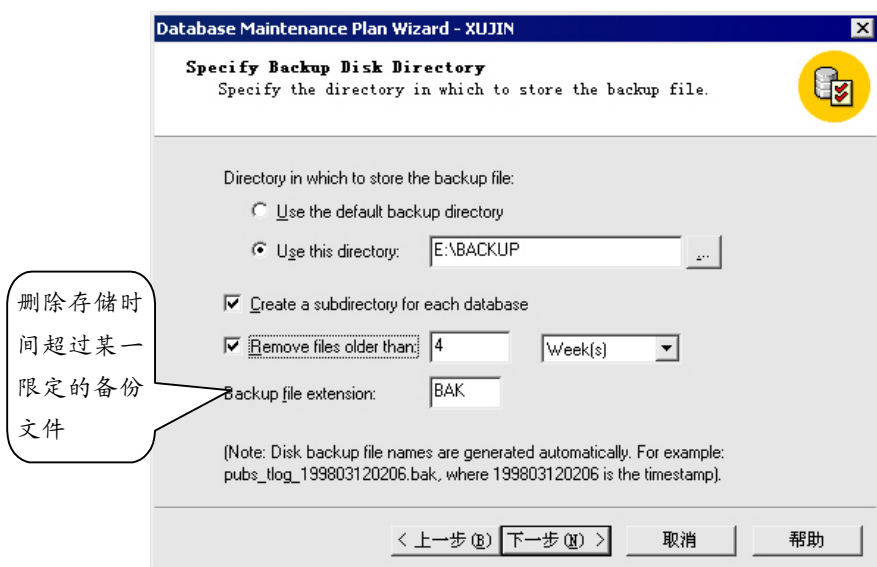


图6-19 指定数据库备份文件存放目录对话框

- (7) 单击“下一步”按钮，出现如图 6-20 所示的指定事务日志文件备份计划对话框。可以选择是否将事务日志备份加入到数据库维护计划中。
- (8) 单击“下一步”按钮，出现如图 6-21 所示的指定事务日志文件备份存放目录对话框。可以选择是否将事务日志备份存放的相关选项。

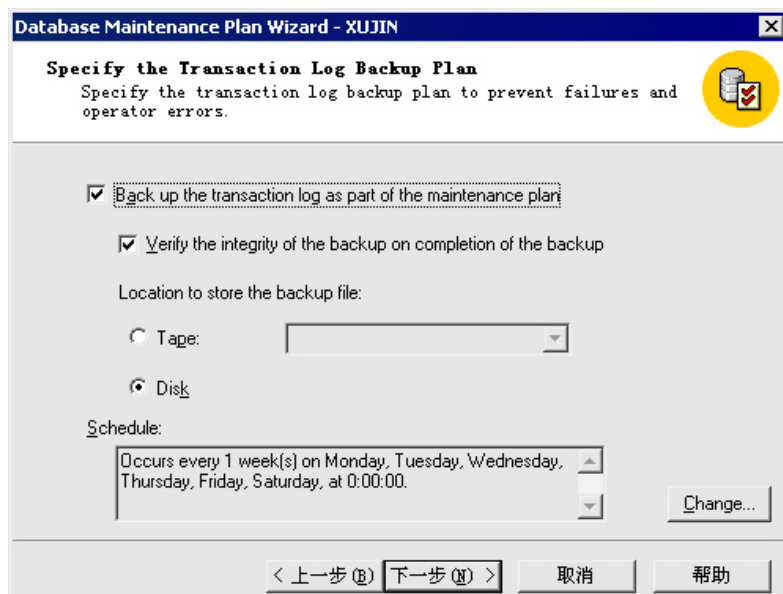


图6-20 指定事务日志文件备份计划对话框

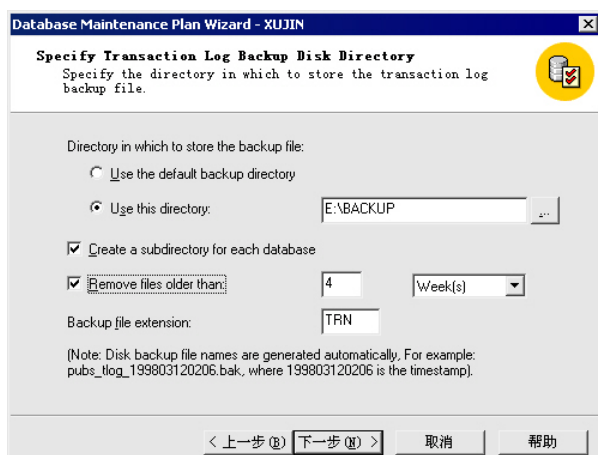


图6-21 指定事务日志文件备份存放目录对话框

(9) 单击“下一步”按钮，出现如图 6-22 所示的产生维护计划报告对话框。可以选择报告的存放目录等选项。

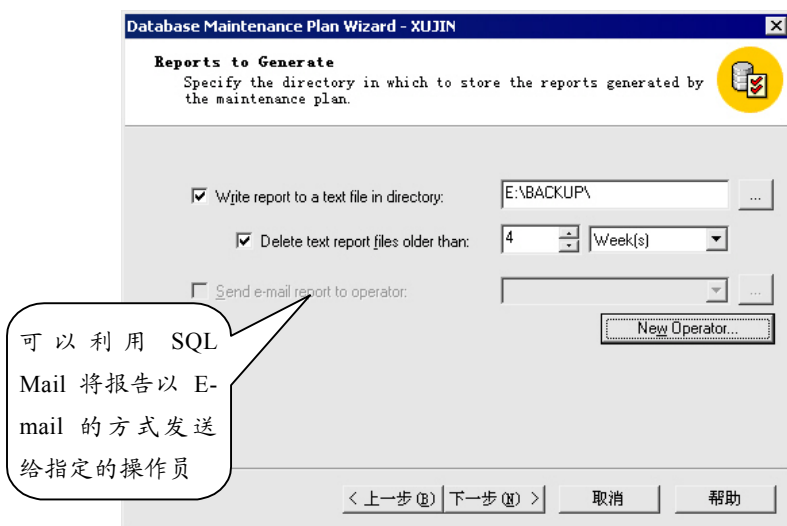


图6-22 产生维护计划报告对话框

(10) 单击“下一步”按钮，出现如图 6-23 所示的指定维护计划执行历史记录存放位置对话框。可以选择将其存放在本地服务器或远程服务器上。

(11) 单击“下一步”按钮，出现如图 6-24 所示的最终界面，指定维护计划名称对话框。输入名称后，单击“完成”按钮，完成维护计划的设置。

在设置好维护计划后，可以在 Enterprise Manager 中查看、修改它。在企业管理器中服务器文件夹下的“Management”子目录下选择“Database Maintenance Plans”就可看到所设置的计划。如图 6-25 所示。

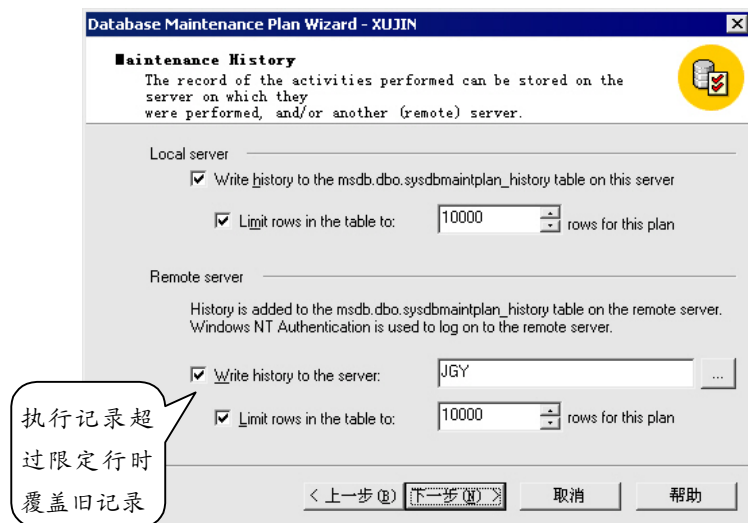


图6-23 指定维护计划执行历史记录存放位置对话框

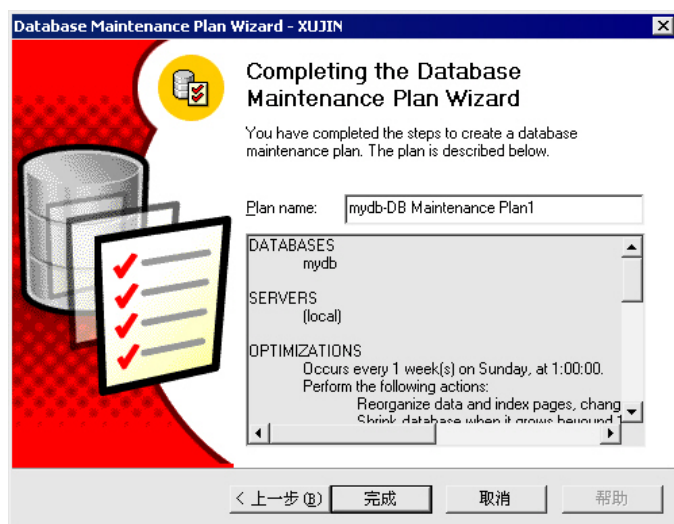


图6-24 指定维护计划名称对话框

在图 6-25 窗口中选择要修改的数据库维护计划名称，单击右键，从快捷菜单中选择“属性（Properties）”选项，则出现如图 6-26 所示的数据库维护计划编辑对话框。可以编辑在计划生成向导中所设置的几乎所有信息。

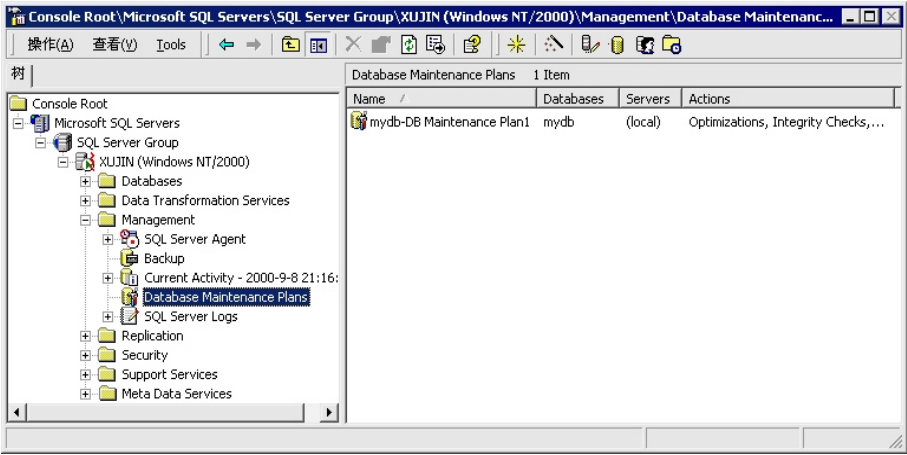


图6-25 维护计划对象

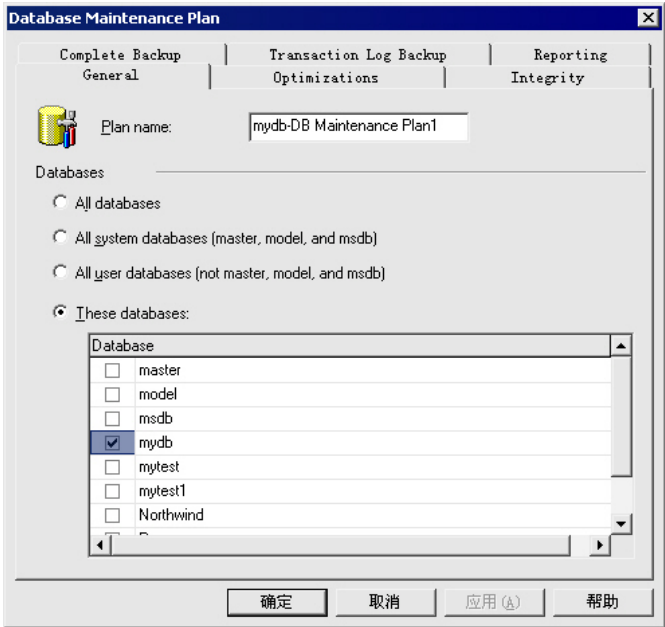


图6-26 数据库维护计划编辑

## 6.9 系统数据库

SQL Server 2000 有四个系统数据库（System Databases）：Master、Model、Msdb、Tempdb。这些系统数据库的文件存储在 Microsoft SQL Server（默认安装目录）的 MSSQL 子目录的 Data 文件夹中。各系统数据库所包含的文件如表 6-1 所示。

表6-1 系统数据库的文件

数据库文件	物理文件名	默认大小（典型安装）
-------	-------	------------

master primary data	Master.mdf	7.5 MB
master log	Mastlog.ldf	1.0 MB
model primary data	Model.mdf	0.75 MB
model log	Modellog.ldf	0.75 MB
msdb primary data	Msdbdata.mdf	3.5 MB
msdb log	Msdblog.ldf	0.75 MB
tempdb primary data	Tempdb.mdf	8.0 MB
tempdb log	Templog.ldf	0.5 MB

如果在企业管理器中看不到系统数据库，则可用鼠标右键单击服务器，从快捷菜单中选择“Edit SQL Server Registration Properties”，会出现如图 5-5 所示的编辑服务器注册属性对话框，从中选择“Show system databases and system objects”选项，再刷新企业管理器界面即可。

## 6.9.1 Master 数据库

Master 数据库记录了一个 SQL Server 系统的所有系统信息。这些系统信息主要有：

- 所有的登录信息；
- 系统设置信息；
- SQL Server 初始化信息；
- 系统中其它系统数据库和用户数据库的相关信息，包括其主文件的存放位置等。

## 6.9.2 Model 数据库

Model 数据库是所有用户数据库和 Tempdb 数据库的创建模板。当创建数据库时，系统会将 Model 数据库中的内容复制到新建的数据库中去。由此可见，利用 Model 数据库的模板特性，通过更改 Model 数据库的设置，并将时常使用的数据库对象复制到 Model 数据库中，可以大大简化数据库及其对象的创建、设置工作，为用户节省大量的时间。通常，可以将以下内容添加到 Model 数据库中：

- 数据库的最小容量；
- 数据库选项设置；
- 经常使用的数据库对象，如用户自定义的数据类型、函数、规则、缺省值等。

## 6.9.3 Msdb 数据库

SQL Server、SQL Server Enterprise Manager 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。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。

## 6.9.4 Tempdb 数据库

Tempdb 数据库用作系统的临时存储空间，其主要作用有：



- 存储用户建立的临时表和临时存储过程;
- 存储用户说明的全局变量值;
- 为数据排序创建临时表;
- 存储用户利用游标说明所筛选出来的数据。

在 Tempdb 数据库中所做的存在不会被记录,因而在 Tempdb 数据库中的表上进行数据操作比在其它数据库中要快得多。

! 当退出 SQL Server 时,用户在 Tempdb 数据库中建立的所有对象都将被删除。每次 SQL Server 启动时,Tempdb 数据库都将被重建,回复到系统设定的初始状态。因此,千万不要将 Tempdb 数据库作为数据的最终存放处。

## 6.10 实例数据库

SQL Server 2000 有两个实例数据库 (Sample Databases): pubs 和 Northwind。与系统数据库一样,实例数据库的文件也存储在 Microsoft SQL Server (默认安装目录)的 MSSQL 子目录的 Data 文件夹中。实例数据库所包含的文件如表 6-2 所示:

表6-2 实例数据库的文件

数据库文件	物理文件名	默认大小 (典型安装)
pubs primary data	Pubs.mdf	1.75 MB
pubs log	Pubs_log.ldf	0.75 MB
Northwind primary data	Northwnd.mdf	3.25 MB
Northwind log	Northwnd.ldf	1.0 MB

pubs 和 Northwind 数据库可以作为 SQL Server 的学习工具,SQL Server Books Online 中的实例基本上都是基于这两个数据库来讲解的。其中 pubs 实例数据库存储了一个虚构的图书出版公司的基本情况。Northwind 实例数据库包含了一个公司的销售数据。此公司名称为 Northwind 商人 (Northwind Traders),是一个虚构的公司,从事食品的进出口业务。

如果改变了实例数据库的内容或不小心删除了实例数据库,想将其回复到初始状态,可以用 SQL Server Analyzer 运行“安装目录\MSSQL\Install”中的 Instpubs.sql 或 Instnwnd.sql 程序来重建 pubs 或 Northwind 数据库。

## 6.11 本章小结

本章介绍了数据库的创建与管理知识,以及系统数据库和实例数据库的基本情况。强大的数据库管理功能是 SQL Server 的特点,掌握本章内容是对数据库管理员的基本要求。