

第 3 章

数据库的基本操作

数据库安装完成后,可在 SQL Server 2008 中建立图书馆管理与借阅网站系统的数据库 Library_DB。本章主要说明在 SQL Server 中如何对数据库和文件组进行创建和管理。

3.1 文件和文件组

3.1.1 文件

每个 SQL Server 2008 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要的信息。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。

SQL Server 2008 数据库具有三种类型的文件,它们分别是主要文件、次要文件和事务日志。

(1) 主要数据文件(建议文件扩展名为 .mdf): 主要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。每个数据库有一个主要数据文件。

(2) 次要数据文件(建议文件扩展名为 .ndf): 次要数据文件是可选的,由用户定义并存储用户数据。通过将每个文件放在不同的磁盘驱动器上,次要文件可用于将数据分散到多个磁盘上。另外,如果数据库超过了单个 Windows 文件的最大容量,可以使用次要数据文件,这样数据库就能继续增长。

(3) 事务日志文件(建议文件扩展名为 .ldf): 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。

例如,可以创建一个简单的数据库 BookList,其中包括一个包含所有数据和对象的主要文件和一个包含事务日志信息的日志文件。也可以创建一个更复杂的数据库 Library,其中包括一个主要文件和 5 个次要文件。数据库中的数据和对象分散在所有 6 个文件中,而 4 个日志文件包含事务日志信息。

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。

“新建数据库”窗口,如图 3-2 所示,在“常规”属性页中输入想要建立的数据库的名称如 Library_DB,单击“确定”按钮建立数据库。

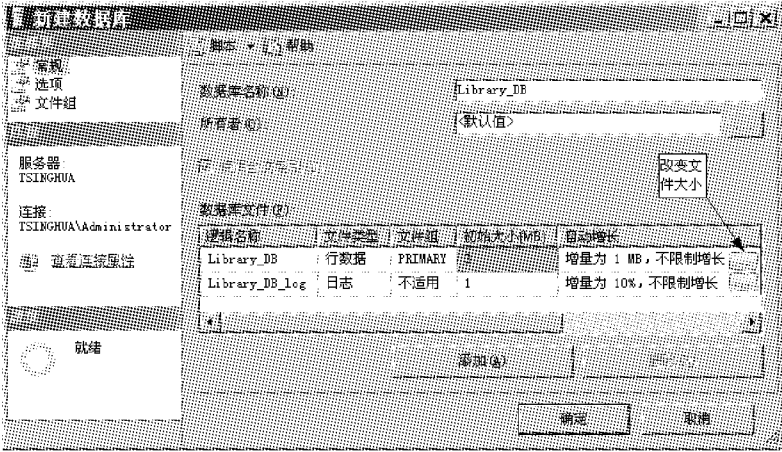


图 3-2 “新建数据库”窗口

用户也可以通过选择图 3-2 中的“数据库文件”栏目,对数据库文件的默认属性进行修改。如图 3-3 所示,可以通过设置“最大文件大小”,将文件增长限制为 150MB,以防止磁盘被数据文件写满。

同理,用户也可以在图 3-2 中通过对“日志”文件类型进行操作,以对数据库的事务日志文件的默认属性进行修改,如图 3-4 所示。

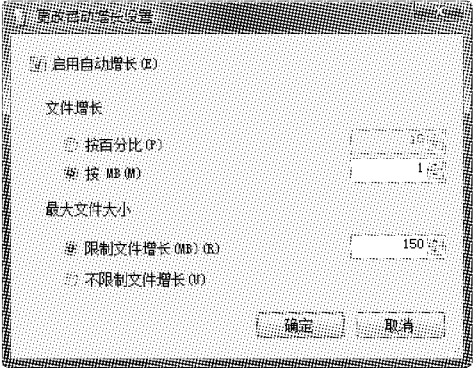


图 3-3 限制文件增长

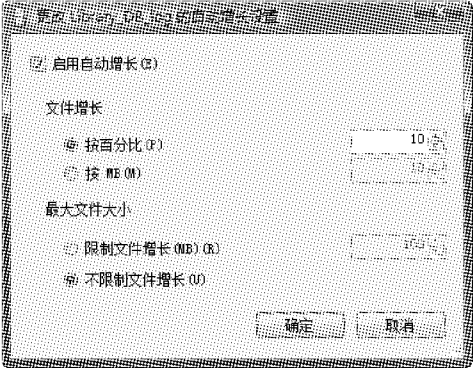


图 3-4 设置“事务日志”属性

创建好 Library_DB 数据库后,在图 3-1 窗口中,展开 SQL Server 组下的“数据库”,用户可以看到新建立的数据库,如图 3-5 所示。

3.2.2 在查询分析器窗口中用 T-SQL 命令创建数据库

用 T-SQL 创建数据库的命令如下。

```
CREATE DATABASE database_name
```



图 3-5 新建的 Library_DB 数据库

```
[ON
    [PRIMARY ] [<filespec> [, ...n]
    [, <filegroup> [ , ...n]]
[LOG ON {<filespec> [ , ...n]]}]
]
[COLLATE collation_name]
[WITH<external_access_option>]
]
[;]
```

To attach a database

```
CREATE DATABASE database_name
    ON<filespec> [ , ...n ]
    FOR {ATTACH [WITH<service_broker_option>]
        |ATTACH_REBUILD_LOG}
[;]
```

<filespec> ::=

```
{
(
    NAME=logical_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|GB|TB|% ]]
    ) [ ,...n ]
}

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

<external_access_option>::=
{
    DB_CHAINING {ON|OFF}
    |TRUSTWORTHY {ON|OFF}
}

<service_broker_option>::=
{
    ENABLE_BROKER
    |NEW_BROKER
    |ERROR_BROKER_CONVERSATIONS
}

Create a database snapshot
CREATE DATABASE database_snapshot_name
    ON
        (
            NAME=logical_file_name,
            FILENAME='os_file_name'
        ) [ ,...n]
    AS SNAPSHOT OF source_database_name
[;]

```

其中重要参数含义如下。

(1) database_name: 新数据库的名称。数据库名称在 SQL Server 的实例中必须唯一,并且必须符合标识符规则。除非没有为日志文件指定逻辑名称,否则 database_name 最多可以包含 128 个字符。如果未指定逻辑日志文件名称,则 SQL Server 将通过向 database_name 追加后缀来为日志生成 logical_file_name 和 os_file_name。这会将 database_name 限制为 123 个字符,从而使生成的逻辑文件名称不超过 128 个字符。

(2) ON: 指定显式定义用来存储数据库数据部分的磁盘文件(数据文件)。当后面是以逗号分隔的、用于定义主文件组的数据文件的<filespec>项列表时,需要使用 ON。主文件组的文件列表可跟以逗号分隔的、用于定义用户文件组及其文件的<filegroup>项列表(可选)。

(3) PRIMARY: 指定关联的<filespec>列表定义主文件。在主文件组的<filespec>

项中指定的第一个文件将成为主文件。一个数据库只能有一个主文件。有关详细信息，请参阅物理数据库文件和文件组。

(4) LOG ON: 指定显式定义用来存储数据库日志的磁盘文件(日志文件)。LOG ON 后跟以逗号分隔的用于定义日志文件的<filespec>项列表。如果没有指定 LOG ON,将自动创建一个日志文件,其大小为该数据库的所有数据文件大小总和的 25% 或 512KB,取两者之中的较大者。不能对数据库快照指定 LOG ON。

(5) COLLATE collation_name: 指定数据库的默认排序规则。排序规则名称既可以是 Windows 排序规则名称,也可以是 SQL 排序规则名称。如果没有指定排序规则,则将 SQL Server 实例的默认排序规则分配为数据库的排序规则。不能对数据库快照指定排序规则名称。

(6) FOR ATTACH: 指定通过附加一组现有的操作系统文件来创建数据库。必须有一个指定主文件的<filespec>项。至于其他<filespec>项,只需要指定与第一次创建数据库或上一次附加数据库时路径不同的文件的那些项即可。必须有一个<filespec>项指定这些文件。

(7) FOR ATTACH_REBUILD_LOG: 指定通过附加一组现有的操作系统文件来创建数据库。该选项只限于读/写数据库。如果缺少一个或多个事务日志文件,将重新生成日志文件。必须有一个指定主文件的<filespec>项。

(8) <filespec>: 控制文件属性。

(9) NAME: logical_file_name: 指定文件的逻辑名称。指定 FILENAME 时,需要使用 NAME,除非指定 FOR ATTACH 子句之一。

(10) logical_file_name: 引用文件时 SQL Server 中使用的逻辑名称。logical_file_name 必须在数据库中唯一,必须符合标识符规则。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。

(11) FILENAME 'os_file_name': 指定操作系统(物理)文件名称。它是创建文件时由操作系统使用的路径和文件名。文件必须驻留在下列一种设备中: 安装 SQL Server 的本地服务器、存储区域网络[*SAN*]或基于 iSCSI 的网络。执行 CREATE DATABASE 语句前,指定路径必须存在。

(12) SIZE size: 指定文件的大小。如果没有为主文件提供 size,则数据库引擎将使用 model 数据库中的主文件的大小。如果指定了辅助数据文件或日志文件,但未指定该文件的 size,则数据库引擎将以 1MB 作为该文件的大小。为主文件指定的大小至少应与 model 数据库的主文件大小相同。

(13) MAXSIZE max_size: 指定文件可增大到的最大容量。可以使用 KB、MB、GB 和 TB 后缀。默认为 MB。指定一个整数,不包含小数位。如果未指定 max_size,则文件将一直增大,直至磁盘已满。

(14) UNLIMITED: 指定文件将增长到磁盘已满。在 SQL Server 2008 中,指定为不限制增长的日志文件的容量最大为 2TB,而数据文件的容量最大为 16TB。

(15) FILEGROWTH growth_increment: 指定文件的自动增量。文件的容量 FILEGROWTH 设置不能超过 MAXSIZE 设置。将 os_file_name 指定为 UNC 路径时,

不能指定 FILEGROWTH。

(16) growth_increment: 每次需要新空间时为文件添加的空间量。该值可以 MB、KB、GB、TB 或百分比(%)为单位指定。如果未在数字后面指定 MB、KB 或 %, 则默认值为 MB。如果指定 %, 则增量大小为发生增长时文件大小的指定百分比。指定的大小舍入为最接近的 64KB 的倍数。值为 0 时表明自动增长被设置为关闭, 不允许增加空间。

(17) <filegroup>: 控制文件组属性。不能对数据库快照指定文件组。

(18) FILEGROUP filegroup_name: 文件组的逻辑名称。

(19) filegroup_name: filegroup_name 必须在数据库中唯一, 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。名称必须符合标识符规则。

(20) DEFAULT: 指定命名文件组为数据库中的默认文件组。

(21) <external_access_option>: 控制外部与数据库之间的双向访问。

(22) DB_CHAINING{ON|OFF}: 当指定为 ON 时, 数据库可以为跨数据库所有权链接的源或目标。当为 OFF 时, 数据库不能参与跨数据库所有权链接。默认值为 OFF。

(23) database_snapshot_name: 新数据库快照的名称。数据库快照名称必须在 SQL Server 的实例中唯一, 并且必须符合标识符规则。database_snapshot_name 最多可以包含 128 个字符。

(24) ON(NAME=logical_file_name, FILENAME='os_file_name')[,...n]: 若要创建数据库快照, 请在源数据库中指定文件列表。若要使快照工作, 必须分别指定所有数据文件。但是, 日志文件不允许用于数据库快照。

(25) AS SNAPSHOT OF source_database_name: 指定要创建的数据库为 source_database_name 指定的源数据库的数据库快照。快照和源数据库必须位于同一实例中。

此处为了保护开发案例中的 Library_DB 数据库, 同时便于演示数据库中的一些操作, 我们将建立一个额外的数据库 My_DB。

如下示例是在 C 盘根目录下创建数据库名为 My_DB 的数据库, 其主文件大小为 10MB, 最大长度为 30MB, 日志文件大小为 10MB。命令如下。

```
CREATE DATABASE My_DB
ON
PRIMARY (NAME=My_DBData,
FILENAME='C:\My_DB.mdf',
SIZE=10MB,
MAXSIZE=30MB,
FILEGROWTH=2MB)
LOG ON
(NAME=My_DBLog,
FILENAME='C:\My_DB.ldf',
SIZE=10MB,
MAXSIZE=20MB,
FILEGROWTH=25%)
```

COLLATE Chinese_PRC_CI_AS

用户可以在图 3-6 中,选择“新建查询”选项,在界面中输入上述命令即可,如图 3-7 所示。

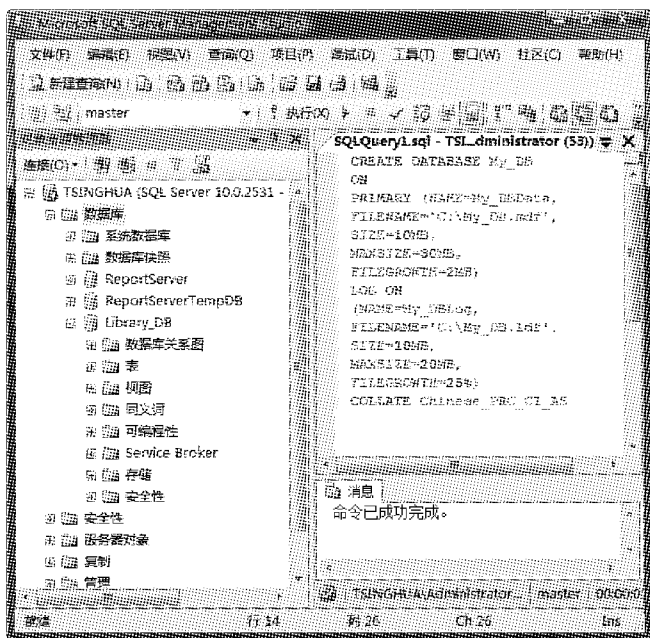


图 3-6 用 T-SQL 命令创建数据库

3.2.3 事务日志

每个 SQL Server 2008 数据库都具有事务日志,用于记录所有事务以及每个事务对数据库所做的修改。事务日志是数据库的一个重要组件,如果系统出现故障,它将成为最新数据的唯一源。删除或移动事务日志以前,必须完全了解此操作带来的后果。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 语句,或者数据库引擎检测到错误(例如失去与客户端的通信),就使用日志记录回滚未完成的事务所做的修改。

(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,数据库可能处于这样的状态,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的事务所做的修改。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。有关详细信息,请参阅事务复制的工作机制。

(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。

SQL Server 使用各数据库的事务日志来恢复事务。事务日志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。事务日志记录每个事务的开始。它记录了在每个事务期间,对数据的更改及撤销所做更改(以后如有必要)所需要的足够信息。对于一些大的操作(如 CREATE INDEX),事务日志则记录该操作发生的事实。随着数据库中发生被记录的操作,日志会不断地增长。记录日志的过程如下。

(1) 应用程序发送数据修改。

(2) 在执行修改时,受影响的数据页面从磁盘加载到缓冲区高速缓冲(Buffer Cache)中,假定这些页面尚未因先前的查询加载到缓冲区高速缓冲中。

(3) 每一条数据修改语句都随着其执行记录在事务日志中。在修改实际写入数据库之前,这些修改总是首先记录到事务日志中,并把事务日志写入到磁盘中。这种类型的日志称作先写(Write-ahead)日志。

(4) 基于递归原则,检查点进程把所有已经完成的事务写入磁盘上的数据库中。如果系统失败的话,自动恢复进程使用事务日志前滚所有已经提交的事务,回滚任何不完整的事务。在自动恢复过程中,日志中的事务标记用于确定事务的起始点和结束点。当 BEGIN TRANSACTION 标记具备相应的 COMMIT TRANSACTION 标记时,认为这个事务是完整的。当发生检查点时,数据页面被写入磁盘。

3.2.4 查看数据库信息

对于已有数据库,我们可以分别利用 Microsoft SQL Server Management Studio 管理界面和 T-SQL 语句来查看数据库信息。

1. 用 Microsoft SQL Server Management Studio 以图形化界面查看数据库信息

(1) 打开 Microsoft SQL Server Management Studio 管理器,在“对象资源管理器”中,选中需要查看信息的数据库,右击,在弹出的快捷菜单中选择“属性”命令,如图 3-7 所示。

(2) 在“数据库属性”对话框中,用户可以选择查看数据库的各种属性,如图 3-8 所示。

2. 在查询分析器窗口中用 T-SQL 命令查看数据库信息

在 T-SQL 中,存在了多种查看数据库信息的语句。最常用的为使用函数 DATABASEPROPERTYEX 或系统存储过程 sp_helpdb 来显示有关数据库和数据库参数的信息。使用函数 DATABASEPROPERTYEX 的语法为

```
Select DATABASEPROPERTYEX(database,property)
```

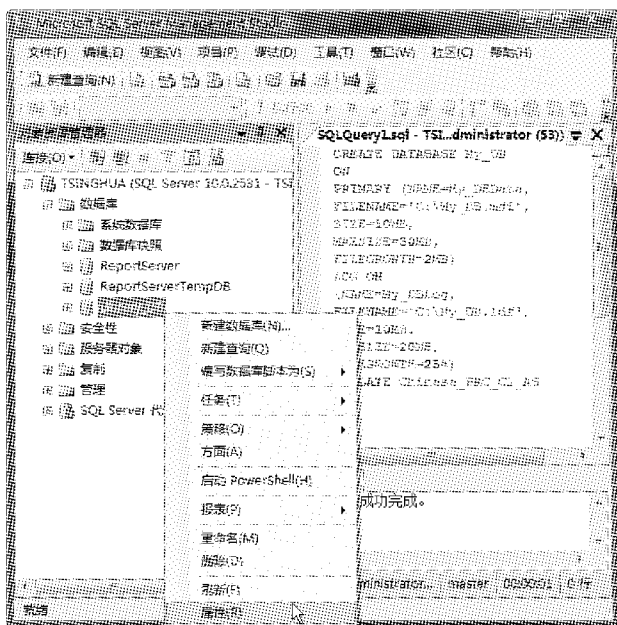


图 3-7 查看数据库属性



图 3-8 “数据库属性”对话框

其中各项的含义如下。

(1) Database: 表示要为其返回命名属性信息的数据库的名称。database 的数据类型为 nvarchar(128)。

(2) Property: 表示要返回的数据库属性的名称的表达式。property 的数据类型为 varchar(128), 可以是下列值之一。返回类型为 sql_variant。

如表 3-1 所示显示了各属性值的基本数据类型。