

第3章

SQL Server 数据库结构和管理

SQL Server 能够支持多个数据库。在一个服务器上最多可以创建 32767 个数据库。创建数据库的用户将成为该数据库的拥有者。数据库中的数据及各逻辑对象存储在操作系统文件或文件组中。对于用户来说创建一个数据库并不难，但对创建前数据库的物理存储规划和逻辑存储规划往往重视不够，因此使得创建的数据库在实际使用中不是很理想。

本章将花比较大的篇幅对数据库的物理存储和逻辑存储进行介绍，比如数据文件的使用分配、事务日志文件的结构、数据库的逻辑组织等，目的是希望读者通过学习了解数据库的这些存储原理，更合理地规划所要创建的数据库（比如一个数据库是用一个操作系统文件存储好，还是把它分成多个操作系统文件存储比较好），从而创建的数据库更接近实际，在实际中更好用。本章还将给出大量创建和管理数据库的实例和配置数据库的实例。

3.1 数据库物理存储结构

数据库管理系统体系结构最底层的管理器从原理上讲都可以统称为磁盘空间管理器，所以，数据库的物理存储是指数据库的数据以什么方式存储在计算机磁盘上，又是怎么在磁盘空间管理器管理下运行的。

3.1.1 数据库文件和文件组

数据库在磁盘上是以文件为单位存储的，由数据库文件和事务日志文件组成，一个数据库至少应该包含一个数据库文件和一个事务日志文件，有时可能还包括次要文件和多个事务日志文件。

- 主数据库文件（Primary Database File）：数据库的起点，可以指向数据库中文件的其他部分。每个数据库都有一个主数据库文件。文件扩展名一般是.mdf。
- 次数据库文件（Secondary Database File）：有的数据库可能没有次数据文件，而有的数据库则有多个次数据文件。文件扩展名多是.ndf。
- 事务日志文件：日志文件包含恢复数据库所需的所有日志信息。每个数据库必须至少有一个日志文件，但可以不止一个。推荐的文件扩展名是.ldf。

图 3-1 所示为 SQL Server 2000 在默认实例上按默认存储位置创建的数据库文件示例，它有一个主数据库文件、两个次数据库文件和两个日志文件。

其他版本存储子目录有所变化，如 SQL Server 2012 版本默认实例存储位置是 C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA

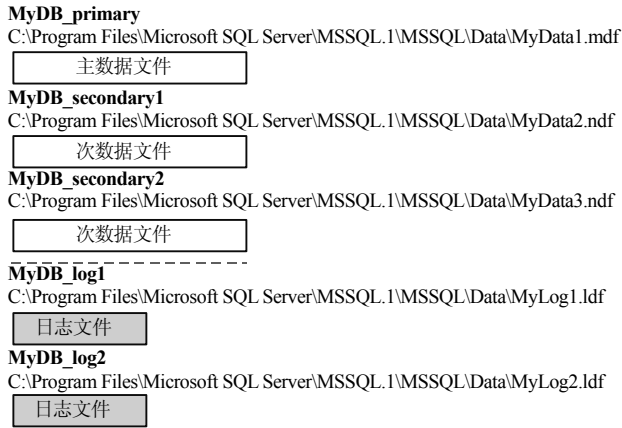


图 3-1 创建的数据库文件存储

为了便于分配和管理，SQL Server 允许将多个文件归纳为同一组，并赋予此组一个名称，这就是文件组。文件组能够控制各文件的存放位置，其中的文件常建立在不同的硬盘驱动器上，这样可以减轻单个磁盘驱动器的存储负载，提高数据库的存储效率，从而达到提高系统性能的目的。

在存储数据时，SQL Server 采用按比例填充的策略使用文件组内每个文件所提供的存储空间。例如，如果一个服务器上有 4 个可供数据库使用的硬盘，提供给数据库的最大存储空间分别为 100 MB、200 MB、300 MB 和 100 MB，我们可以在前面 3 个硬盘上建立一个数据文件组，包含 3 个文件，每个硬盘上分配 1 个文件，在第 4 个硬盘上建立数据库的日志文件，当发生数据库的读写操作时，日志数据写入第 4 个硬盘，而数据库数据写入前 3 个硬盘。在写入数据时，SQL Server 根据文件组内每个数据文件中剩余的空间大小按比例分配写入其中的数据量，即 1:2:3。这样不但保证文件组内每个文件的空間基本上同时用完，而且将一次磁盘操作同时分配给多个磁盘控制器，可以减轻每个磁盘的负载，从而提高写入速度。

在 SQL Server 中建立数据文件和文件组时，应注意以下两点：

- ① 每个数据文件或文件组只能属于一个数据库，每个数据文件也只能成为一个文件组的成员。也就是说，数据文件不能跨文件组使用，数据文件和文件组不能跨数据库使用。
- ② 日志文件是独立的，它不能作为其他数据文件组的成员。即数据库内的数据和日志不能存入相同的文件或文件组。

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）。

① 主文件组：包含主数据文件和所有未被包含在其他文件组里的文件。在创建数据库时，如果未指定其他数据文件所属文件组，这些文件将归属于主文件组。数据库的系统表都包含在主文件组中，所以，当主文件组的空间用完后，将无法向系统表中添加新的目录信息。

② 次文件组：也称为用户自定义文件组，包括所有使用数据库创建语句（CREATE DATABASES）或数据库修改语句（ALTER DATABASES）时使用 FILEGROUP 关键字指定的文件。

任何时候，只能有一个文件组是默认文件组。默认情况下，主文件组被认为是默认文件组。

使用数据文件和文件组应注意以下几点。

- ① 创建数据库时，允许数据文件能够自动增长，但要设置一个上限，否则有可能充满磁盘。
- ② 主文件组要足够大以容纳所有的系统表，否则新的信息就无法添加到系统表中，数据库也就无法追加修改。
- ③ 建议把频繁查询的文件和频繁修改的文件分放在不同的文件组中。
- ④ 把索引、大型的文本文件和图像文件放到专门的文件组中。

3.1.2 数据文件的使用分配

1. 基本知识

数据库管理系统体系结构底层的这些管理器，从原理上讲都可以统称为磁盘空间管理器，磁盘空间管理器支持作为数据单元的页的概念，并且提供分配、回收页和读写页的命令。通常以磁盘块的大小作为页的大小，并且将页以磁盘块的方式存储起来，以便在一次磁盘 I/O 中就能完成一页的读写。

为一连串的页分配连续的磁盘块来存放那些需要频繁地按顺序访问的数据是非常有用的。对发挥顺序访问磁盘块的优势来说，这种能力是必需的。如果有必要，这种能力也可以由磁盘空间管理器提供。

总之，磁盘空间管理器隐藏了底层硬件和操作系统的细节，并允许数据库管理系统的高层软件把数据看成是页的集合。因此，在 SQL Server 的体系结构中有专门的页管理器和文本管理器。

在 SQL Server 中，数据文件存储的基本单位是页，页的大小是 8 KB。这就意味着 SQL Server 数据库每兆字节有 128 页。每页的开始部分是 96 字节的页首，用于存储系统信息，如页的类型、页的可用空间量、拥有页的对象 ID 等。根据页面所存储的不同信息，可以将它划分为 8 种类型。

表 3-1 所示为 SQL Server 数据库的数据文件中的 8 种页类型。

表 3-1 SQL Server 数据库的数据文件中的 8 种页类型

页类型	内容
数据页面	存储数据行中除 TEXT、nTEXT 和 IMAGE 列数据以外的数据
文本/图像页面	存储数据行中的 TEXT、nTEXT 和 IMAGE 列数据
索引页面	存储索引项
全局分配映射页面	存储数据文件的区域分配信息
页的可用空间信息页面	存储数据文件中可用的空闲页面信息
索引分配映射页面	存储表或索引所使用的区域信息
大容量更改信息页面	存储有关自上次执行 BACKUP LOG 语句后大容量操作所修改的扩展盘区的信息
差异更改信息页面	存储有关自上次执行 BACKUP DATABASE 语句后更改的扩展盘区的信息

2. 数据页面存储格式

(1) 数据页面

在 SQL Server 中，数据文件的页按顺序编号，这个编号称为页码。文件首页的页码是 0。众多数据页构成一个数据文件，每个数据文件都有一个文件 ID。在数据库中唯一标识的一页需要同时使用文件 ID 和页码。

数据页中数据由数据行组成，数据行中包含除 TEXT、NTEXT 和 IMAGE 数据外的所有数据，TEXT、NTEXT 和 IMAGE 数据存储在单独的页中。在数据页上，数据行紧接着页首按顺序放置。在页尾有一个行偏移表。页上的每一行在行偏移表中都有一个条目，每个条目记录对应行的第 1 字节与页首的距离。行偏移表中的条目序列与页中行的序列相反。数据页面的存储结构如图 3-2 所示。

页首占用每个数据页的前 96 字节，剩余的 8096 字节用于数据和行偏移数组。

(2) 数据行

紧跟着页头的就是存储表的真正数据行区域。单个数据行的最大长度是 8060 字节。

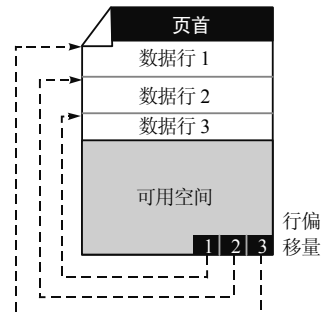


图 3-2 数据页面的存储结构

数据行不能跨页存储（文本和图像例外）。页内数据行的多少依赖于表的结构和要存储的数据。如果一个表的所有列都是定长的，那么该表在每一页上存储相同数目的行。如果一个表中有变长列，那么该表总是在每一页上存储尽可能多的行。数据行越短，每一页存储的行数就越多。

(3) 行偏移表

当单行数据长度为最大 8060 字节时，行偏移表占用 $8096 - 8060 = 36$ 字节。

但实际中一个数据行大多不是 8060 字节，往往比这个数字小，所以数据行占用的总字节数目和行偏移表占用的总字节数是系统动态调整的，数据行字节越少，行偏移表字节越多，反之，数据行字节越多，行偏移表字节越少，但不能少于 36 字节。

每两个字节构成一个条目块，每个条目表示页中相关数据行第 1 字节相对页首的偏移量。

注意：行偏移数组表示的是页中数据行的逻辑顺序，不是物理顺序。真正的物理顺序与聚集索引有关。

(4) 页面链接

每个表或索引视图的数据行一般都分开存储在多个 8 KB 数据页中。如上所述，每个数据页都有一个 96 字节的页头，其中包含拥有该页的表的标志符（ID）这样的系统信息，也包含指向下一页及前面用过的页的指针，如图 3-3 所示。

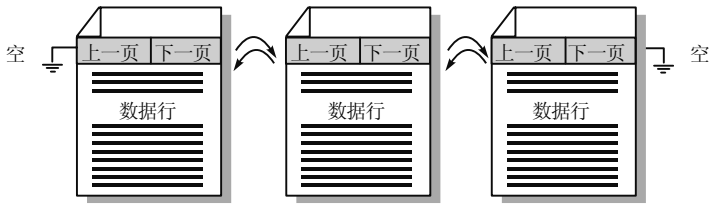


图 3-3 页面链接

3. SQL Server 空间使用分配

SQL Server 数据库是存储表和索引的，向表或索引分配空间的基本单位为区域，一个区域长度为 8 个连续的页面，也就是 64 KB。区域分为以下两种类型。

① 统一区域：区域中的 8 个页面只能存储同一种数据库对象。

② 混合区域：区域中不同页面可以存储不同的数据库对象。1 个混合区域最多可以存储 8 种数据库对象。

SQL Server 对空间使用分配时，当数据库对象中的数据较少时，可以分配混合区域，供数据库中不同类型的数据库对象使用。但当同种数据库对象所占用的空间达到 8 个页面时，可以再将它们转换为统一区域。

SQL Server 使用以下两种类型的分配映射页面记录区域的分配使用情况。

① 全局分配映射（Global Allocation Map, GAM）页面：用来记录哪些区域已经作为何种类型的区域分配使用。该页面中的每一位记录一个区域的分配情况，当位值为 1 时，说明区域为空闲区域，当位值为 0 时，表示区域已经被分配使用。由于每个页面大小为 8 KB，所以一个 GAM 页面能够覆盖 64 000 个区域，即 4 GB。

② 共享全局分配映射（Shared Global Allocation Map, SGAM）页面：用来记录有空闲页面的混合区域，而且至少有一页未被使用。每个 SGAM 页面覆盖 64 000 个区域。当位值为 1 时，说明区域

为混合区域，并且其中有空闲页面；当位值为 0 时，说明相应的盘区未被用做混合区域，或者它是一个没有空闲页可分配的混合区域。

表 3-2 说明了区域可能的位值设置情况。

SQL Server 在分配统一区域时，首先在 GAM 中查找到一个位值为 1 的位，然后将其设置为 0；而在分配混合区域时，首先在 GAM 中查找到一个位值为 1 的位，并将其值设置为 0，再将 SGAM 页面中对应位值设置为 1。当释放一个区域时，SQL Server 将 GAM 和 SGAM 页面中的对应位分别设置为 1 和 0。所以如果 SQL Server 需要找到一个新的、完全未被使用的区域，就可以使用 GAM 页面中相应位值是 1 的所有区域；如果 SQL Server 需要找到一个可用的空间的混合区域，那么 GAM 页面中对应位值是 0，而且 SGAM 页面中对应位值是 1 的任何区域就都满足条件。

在任何 SQL Server 数据库文件中，页面 0（即第 1 页）永远是文件头页面，而且每个文件只能有一个文件头页面存在，页面 1（即第 2 页）是 PFS（Page Free Space）页，即页面自由空间页面，第一个 GAM 页面总是页 2（即第 3 页），之后，每隔 64 000 个区域就建立另一个 GAM 或 SGAM 页面。

4. 索引分配映射管理

索引分配映射（IAM）页面管理堆或索引所分配区域的使用情况，它是根据每一个对象的需要来分配的。每个 IAM 页面的页面头记录该 IAM 页面所映射区域范围的起始区域，其映射区中的每一位说明一个区域的使用状态，其中第 1 位代表 IAM 页面所映射区域范围内的第 1 个区域，第 2 位代表第 2 个区域，等等。当映射区中某位为 0 时，说明该位所映射的区域仍未分配给拥有该 IAM 页面的对象使用；当其值为 1 时，说明该位所映射区域已经分配给拥有该 IAM 页面的对象使用。

每个堆或索引可以有多个 IAM 页面记录分配给该对象区域的使用情况，堆或索引在每个分配有区域的数据文件中至少有一个 IAM 页面。这些 IAM 页面在数据文件中没有固定的位置，它们根据需要进行分配，并随机定位。一个对象的所有 IAM 页面组成一个链表，其第一个 IAM 页面的位置记录在 sysindexes 系统表的 FirstIAM 列内。所以，根据 FirstIAM 列和 IAM 页面链表就可以查找到一个对象的所有 IAM 页面。SQL Server 确定某个页面属于某个表的唯一方法就是检查该表的 IAM 页。

5. 自由页面及页面空间管理

一旦一个区域分配给某个对象，就可以向这些区域的页中插入数据。如果数据被插入到 B 树，那么新数据的位置基于它在 B 树内的顺序，如果数据被插入到堆中，新数据就可以插入到任何有空闲区域的地方。在数据库文件中，SQL Server 使用 PFS（Page Free Space）页面记录每个单独的页是否已经被分配，以及页面中的空间使用情况，即全部空闲、1%~50%满、51%~80%满、81%~95%满，还是 96%~100%满。当 SQL Server 需要分配新的页面，或者查找到有自由空间页面时，它使用 PFS 页面中的这些信息。数据库文件的第 2 页（页面 1）是第一个 PFS 页，其后每经过 8088 页就又是一个 PFS 页。

SQL Server 使用 IAM 页面和 PFS 页面在已经分配的区域为新插入的数据行查找空闲的页面空间，如果在这些区域内不能查找到足够的空间，它将分配给一个新的区域给数据库对象。

3.1.3 事务日志文件结构

1. 事务日志

每个 SQL Server 数据库都有事务日志，用以记录所有事务和每个事务对数据库所做的修改，目的是一旦数据库出现故障，可以迅速执行前滚、重做等操作，从而保证数据库数据的一致性。为了获得

表 3-2 区域在 GAM 和 SGAM 页面中的位值设置

区域使用情况	GAM 位值	SGAM 位值
空闲，未分配使用	1	0
统一区域或其中页面全部用完的混合区域	0	0
混合区域并且其中有空闲页面	0	1

最大的吞吐效果，日志管理器在内存中建立了两个或更多的快速缓冲区，在检索数据时，它将数据读入该缓冲区，而在修改数据时，它并不是直接修改磁盘中的数据，而是先在缓冲区中建立修改数据副本，之后在页面刷新时再将它们写入磁盘。当出现当前日志缓冲区爆满的时候，备用的快速缓冲区就立即启用。每当对缓冲区中的数据页面进行修改时，SQL Server 自动在日志缓存中构造该操作的日志记录。每个日志记录由日志序列号所标识（Log Sequence Number, LSN），一个新的日志记录的日志序列号均大于其前面记录的日志序列号，新的日志记录被写在日志的逻辑尾部。

日志记录所记录的操作类型如下：

- 每个事务的开始和结束。
- 数据的插入、修改和删除操作等。
- 事务操作的对象。
- 修改前数据的旧值，修改后数据的新值。
- 区域的分配和释放。
- 表和索引的创建和删除。

在数据库修复阶段所需要的事务日志称做事务日志的有效部分，事务日志有效部分中起始记录的日志序列号叫做最小恢复日志序列号（MinLSN）。MinLSN 之前的日志记录对数据库修复操作没有任何用途，但是在数据库日志备份和恢复操作时仍需要事务日志有效部分之前的日志记录，使用它们能够前滚数据库故障点之前的修改内容。

日志记录删除操作称为截断数据库日志。如果一直不截断一个数据库的事务日志，其日志记录最终可能消耗完整个日志文件空间。默认时，在日志备份后，SQL Server 会自动删除日志中的不活动部分。

2. 事务日志的物理存储

一个数据库事务日志可以对应一个或多个物理文件，SQL Server 在内部又将每个物理日志文件分成许多个虚拟日志文件。虚拟日志文件没有固定大小，且物理日志文件所包含的虚拟日志文件数不固定。在创建或扩展日志文件时，SQL Server 动态地选择虚拟日志文件的大小。虚拟日志文件的大小或数量不能由管理员配置或设置，而由 SQL Server 代码动态确定。

事务日志是回绕的日志文件。例如，假设有一个数据库，它包含一个分成 4 个虚拟日志文件的物理日志文件。当创建数据库时，逻辑日志文件从物理日志文件的始端开始。在逻辑日志的末端添加新的日志记录，逻辑日志就向物理日志末端增长。截断操作发生时，删除最小恢复日志序号（MinLSN）之前的虚拟日志内的记录，这部分日志记录所占用的空间即可被重复使用。这一过程如图 3-4 所示。

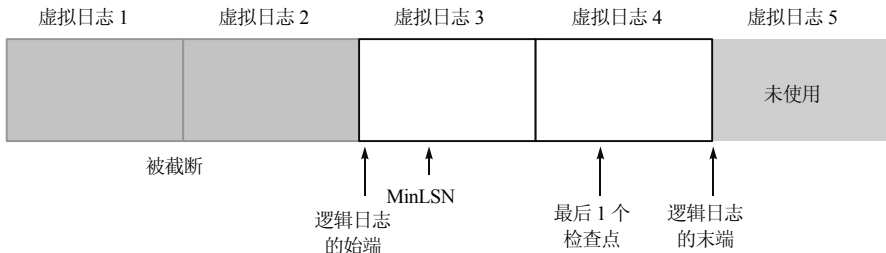


图 3-4 日志记录顺序填充过程

当逻辑日志的末端到达物理日志文件的末端时，新的日志记录绕回物理日志文件的始端，如图 3-5 所示。这个循环不断重复，只要逻辑日志的末端不到达逻辑日志的始端。如果经常截断旧的日志记录，使得总能为下一个检查点创建的所有新日志记录保留足够的空间，那么日志永远不会填满。然而，如果逻辑日志的末端真的到达了逻辑日志的始端，SQL Server 将执行下面两方面的操作：

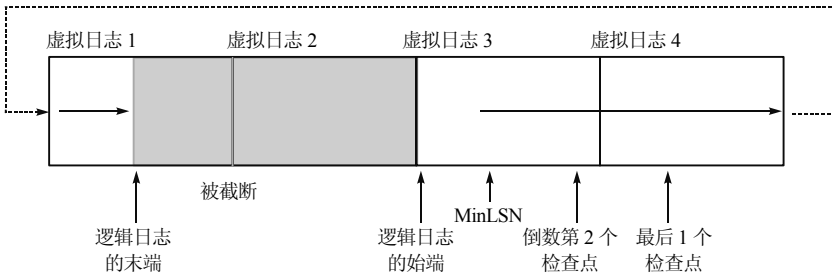


图 3-5 日志记录循环填充过程

① 如果对日志启用了自动增长且磁盘上有可用空间，文件就按指定的数量扩展，新的日志记录则添加到扩展的日志文件中。

② 如果没有启用自动增长，或者保存日志文件的磁盘上的可用空间比指定的日志文件增长数量少，系统将产生 1105 个错误。

3.2 数据库的逻辑组织

一个数据库服务器上可以有多个数据库，这些数据库可以按功能进行划分，最终存储管理相应的数据。每个数据库又有各种不同的逻辑组件。

3.2.1 数据库构架

SQL Server 数据存储在数据库中。在数据库中，数据被组织到用户可以看见的逻辑组件中。数据库存储按物理方式在磁盘上作为两个或更多的文件实现。文件的物理实现在很大程度上是透明的。一般只有数据库管理员需要处理物理实现。两者的关系如图 3-6 所示。

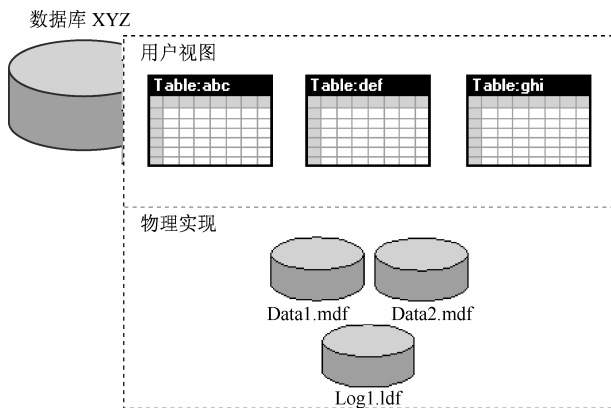


图 3-6 数据库物理实现和用户视图的关系

用户使用数据库时，使用的主要是逻辑组件，如表、视图、过程和用户。

每个 SQL Server 实例有 4 个系统数据库 (master、model、tempdb 和 msdb)，以及 1 个或多个用户数据库。有些单位只使用 1 个用户数据库来存储其所有数据；有些单位则为本单位的每一个组都设立不同的数据库，而且有时一个数据库只能由一个应用程序使用。例如，一个单位可以有销售数据库、工资单数据库、雇员数据库等。应用程序有时只使用一个数据库，有时则可以访问几个数据库，其结构如图 3-7 所示。

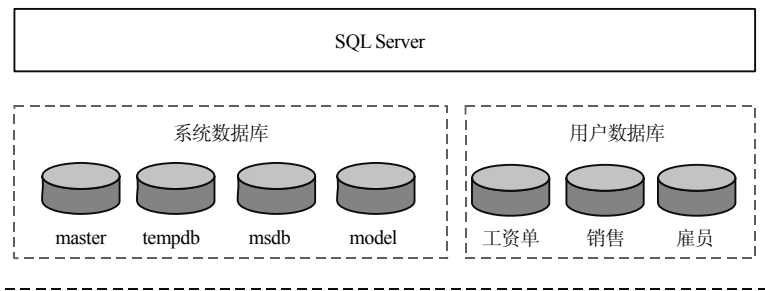


图 3-7 SQL Server 数据库构架

当连接到 SQL Server 实例时，用户的连接会与服务器上的某个具体数据库相关联，这个数据库就称为当前数据库。一般情况下，系统管理员通常会将用户的连接与默认数据库相连。但用户可以使用数据库 API 内的连接选项来指定另一个数据库。可以使用 Transact-SQL USE database_name 语句，或使用可更改当前数据库上下文的 API 函数，由一个数据库切换到另一个数据库。

3.2.2 系统数据库

在 SQL Server 中，除了用户自己定义的数据库之外，还有 4 个系统数据库，如表 3-3 所示。注意不要在 master 数据库中创建任何用户对象。表中列出了其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 3-3 系统数据库对照表

数据库文件	物理文件名	默认大小（典型安装）
master 主数据	Master.mdf	4.87 MB
master 日志	Mastlog.ldf	1.75 MB
tempdb 主数据	Tempdb.mdf	8.00 MB
tempdb 日志	Templog.ldf	0.76 MB
model 主数据	Model.mdf	4.06 MB
model 日志	Modellog.ldf	1.25 MB
msdb 主数据	Msdldata.mdf	16.60 MB
msdb 日志	Msdblog.ldf	4.56 MB

(1) master 数据库

master 数据库由系统表组成，记录了安装及随后创建的所有数据库的信息，包括数据库所用磁盘空间、文件分配、空间使用率、系统级的配置设置、登录账户密码、存储位置等。

master 数据库是系统的关键，不允许任何人对它进行修改。必须经常保留一份它当前的备份。

(2) tempdb 数据库

tempdb 数据库记录了用户创建的所有临时表、临时数据和临时的存储过程。该数据库是一个全局资源，允许所有可以连接上的 SQL 服务器访问。

注意，每次 SQL Server 重新启动，该数据库被重新创建而不是恢复，所以以前用户创建的任何临时对象（表、数据、存储过程等）都将丢失。也就是说，每次 SQL Server 重启，tempdb 数据库都是空的。

(3) model 数据库

model 数据库是建立新数据库的模板。每当创建一个新的数据库时（比如用对象资源管理器去创建，或用 CREATE DATABASE 创建），SQL Server 就会根据 model 数据库的内容来形成新数据库结构的基础，把后面初始化为空，以准备存放数据，同时将系统表复制到刚创建的数据库中去。

严格禁止删除 model 数据库，否则 SQL Server 系统将无法使用。

(4) msdb 数据库

msdb 数据库是由 SQL Server Agent 服务使用的数据库。由于 SQL Server Agent 主要执行一些事先安排好的任务，所以该数据库多用于进行复制、作业调度以及管理报警等活动。

如果不使用代理服务功能，我们可以忽略这个数据库。

3.2.3 用户数据库

在 SQL Server 中，一个用户数据库由用户定义的用来永久存储像表和索引这样的数据库对象的磁盘空间构成，这些空间被分配在一个或多个操作系统文件上。

用户数据库和系统数据库一样，也被划分成许多逻辑页（每个逻辑页的大小是 8 KB），在每个数据库文件中，页从 0 到 X 连续编号，上限值 X 是由文件的大小决定的。

通过指定数据库 ID、文件 ID 和页号，可以引用任何一页。

当扩大文件时，新空间被追加到文件的末尾。

使用 CREATE DATABASE 语句创建一个新的用户数据库，该数据库就被赋予了一个唯一的数据库 ID，或者说 dbid，同时在 master 数据库中的 sysdatabases 表中就会插入一个新建，如图 3-8 所示。（注意，sysdatabases 表只在 master 数据库中。）

	name	dbid	sid	mode	status	status2	crdate
1	master	1	0x01	0	65544	1090520064	2003-04-08 09
2	tempdb	2	0x01	0	8	1090519040	2010-07-31 07
3	model	3	0x01	0	65536	1090519040	2003-04-08 09
4	msdb	4	0x01	0	65544	1627390976	2005-10-14 01
5	教学管理	5	0x0105000000000005150000000DE8E41C9A7CD636235F63...	0	65536	1090519040	2010-07-30 09
6	教学练习	6	0x0105000000000005150000000DE8E41C9A7CD636235F63...	0	65536	1627390976	2010-07-30 09

图 3-8 用户数据库在 sysdatabases 表中的记录

3.3 数据库创建与管理

3.3.1 创建数据库

SQL Server 能够支持多个数据库。在一个服务器上最多可以创建 32 767 个数据库。创建数据库的用户就成为该数据库的所有者。每个数据库都必须包含一个也只能包含一个主数据文件，必要时可以拥有多个次文件；每个数据库至少有一个日志文件，也可以有多个日志文件。可以把各个数据库文件组织成不同的文件组。

每个数据库由以下几个部分的数据库对象组成：关系图、表、视图、存储过程、用户、角色、规则、默认、用户自定义数据类型和用户自定义函数。

1. 准备创建数据库

① 确定数据库的名称、所有者（创建数据库的用户）。

② 确定存储该数据库的数据文件的初始大小及文件空间增长方式、日志、备份和系统存储参数等配置。

下面以创建教学管理数据库为例说明。

数据库：教学管理。

数据文件逻辑名称：教学管理_data。

日志文件逻辑名称：教学管理_log。

数据文件存储：d:\server\MSSQL\data\教学管理_data.mdf，初始大小 2 MB，最大空间 20 MB，增加量 2 MB，主文件中包含数据库的系统表。

日志文件存储：d:\server\MSSQL\data\教学管理_data.ldf，初始大小 2 MB，最大空间 20 MB，增加量 2 MB。

备份设备名称: backup。

备份文件: 教学管理_backup.dat。

2. 创建数据库实例

创建数据库的方法有使用对象资源管理器和使用 Transact-SQL 命令两种。

方法一: 使用对象资源管理器创建数据库

【例 3-1】创建“教学管理”数据库。

① 在对象资源管理器中, 在数据库文件夹下的“数据库”图标上单击鼠标右键, 选择“新建数据库”选项, 就会出现如图 3-9 所示的下拉菜单。

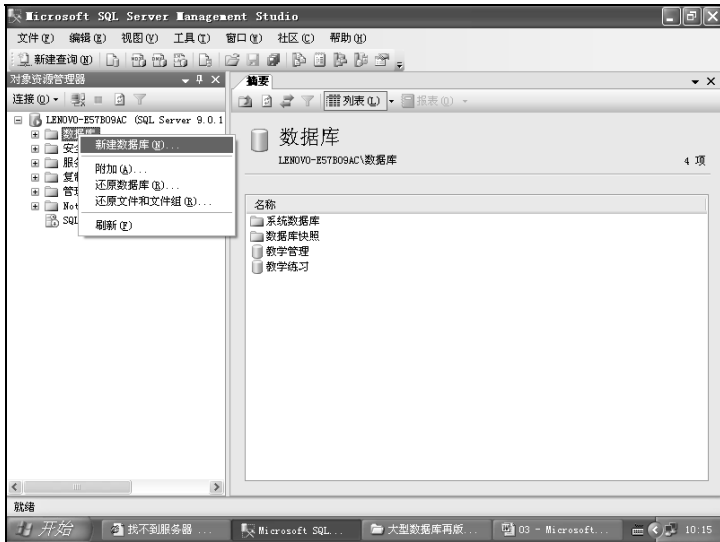



图 3-9 “新建数据库”快捷菜单

② 在“常规”选项卡中, 要求用户输入数据库名称。输入新建数据库名称“教学管理”, 如图 3-10 所示。



图 3-10 “数据库属性”对话框

③ 单击数据文件的自动增长  按钮,如图 3-11 所示。

- “按 MB”:表示指定数据文件按固定步长增长,并要求指定一个值。
- “按百分比”:表示指定数据文件按当前大小的百分比增长,并要求指定一个值。
- “不限制文件增长”:表示允许文件按需求增长。
- “限制文件增长(MB)”,表示允许文件增长到指定的最大值。

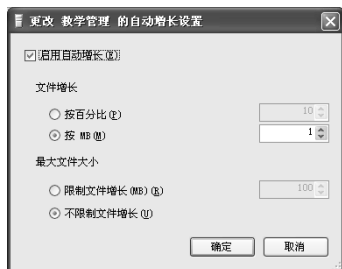


图 3-11 “数据文件”选项卡

建议指定文件最大允许增长的大小,这样做可以防止文件在添加数据时无限增大,以至于用尽磁盘空间。

④ 单击“事务日志”选项卡,从中设置事务日志文件信息,与图 3-11 类似。

⑤ 单击“确定”按钮,则开始创建新的数据库。

方法二:使用 SQL 命令创建数据库

语法格式:

```
CREATE DATABASE 数据库名
    [ON [PRIMARY] [<NAME='数据库逻辑名称'> [,...n] [,< 文件名> [,...n]] ]
    [LOG ON {<LOG_NAME > [,...n]}]
    [FOR RESTORE]
    FILENAME='数据库物理文件名'
        [, SIZE=size]
        [, MAXSIZE={max_size|UNLIMITED}]
        [, FILEGROWTH=growth_increment] ) [,...n]
```

参数说明:

- PRIMARY 是一个关键字,指定主数据库文件,若未给出这个关键字,则默认文件序列第一个是主数据文件。
- LOG ON 指明事务日志文件的明确定义。
- NAME=‘数据库逻辑名称’是在 SQL Server 系统中使用的名称,是数据库在 SQL Server 中的标识符。
- FILENAME=‘数据库物理文件名’指定数据库所在文件的操作系统文件名称和路径,该操作系统文件名与 NAME 的逻辑名称一一对应。
- SIZE 指定数据库的初始容量大小。
- MAXSIZE 指定操作系统文件可以增长到的最大尺寸。
- FILEGROWTH 指定文件每次增加容量的大小,当指定数据为 0 时,表示文件不增长。

【例 3-2】指定文件组,创建一个多数据文件和日志文件的数据库。

该数据库名称为“教学练习”。其中,数据文件“教学练习_data”分配 20 MB,属于文件组 PRIMARY;“教学练习_data1”和“教学练习_data2”各分配 10 MB,属于文件组 stugroup。有 1 个 10 MB 的事务日志文件。

创建命令的程序清单如下:

```
CREATE DATABASE 教学练习      --数据库名称: 教学练习
ON PRIMARY                  --下面主文件属于主文件组
(NAME= 教学练习_data,      --对应数据库第一个逻辑文件名
FILENAME='d:\server\mssql\data\教学练习_data.mdf',
```

```

--对应的主文件名和存储位置
SIZE=20, --初始分配的空间
MAXSIZE=150, --指出最大空间为 150MB
FILEGROWTH=10%), --空间增长按 10%增长
FILEGROUP stugroup --指定新的文件组, 下面文件属于 stugroup 文件组
(NAME= 教学练习_data1, --对应数据库第二个逻辑文件名
FILENAME='d:\server\mssql\data\教学练习_data1.ndf',
--对应的第一个次文件名和存储位置
SIZE=10, --初始分配的空间
MAXSIZE=100, --指出最大空间为 100MB
FILEGROWTH=1), --空间增长按 1MB 增长
(NAME= 教学练习_data2, --对应数据库第三个逻辑文件名
FILENAME='d:\server\mssql\data\教学练习_data2.ndf',
--对应的第二个次文件名和存储位置
SIZE=10, --初始分配的空间
MAXSIZE=100, --指出最大空间为 100 MB
FILEGROWTH=1) --空间增长按 1 MB 增长
LOG ON --日志文件
(NAME= 教学练习_log,
FILENAME='d:\server\mssql\data\教学练习_log.ldf',
SIZE=10,
MAXSIZE=50,
FILEGROWTH=1)
```

输出结果如下:

命令已成功完成。

说明:

- ① 执行命令后, 如果语句执行正确, 则出现如上所述的数据库创建成功的信息。
- ② 如果命令未成功执行, 则出现出错信息, 比如, 出现如下提示信息:

设备激活错误。

物理文件名'd:\server\mssql\data\教学练习_data.mdf'可能有误, 创建数据库失败。
未能创建所列出的某些文件名, 请检查前面的错误信息。

上述命令失败的主要原因是存储物理文件的目录“d:\server\mssql\data”不存在, 解决的办法是建立该文件夹。

3.3.2 管理数据库

1. 修改数据库

数据库创建后, 主数据文件和日志文件的物理地址就不允许被改变和删除了。但数据文件和日志文件的大小、增长方式等属性可以改变, 可以增加或删除次数据文件、次日志文件、文件组。

修改数据库的方法有使用 SQL 命令和使用对象资源管理器两种。

方法一: 使用对象资源管理器

【例 3-3】修改“教学练习”数据库。

① 在对象资源管理器的“教学练习”数据库节点上单击右键, 在出现的快捷菜单中选择“属性”, 再单击“文件”标签, 如图 3-12 所示。

② 当数据文件的容量不够存储数据时, 可以考虑增加数据文件。单击页面下面的“添加”按钮,



直接在文件名一列的新行处输入要添加的数据文件逻辑名称“教学练习_add”，单击自动增长  按钮，如图 3-12 所示。分配空间 5，选择“限制文件增长”；再单击路径  按钮，确定物理文件的存储位置。



图 3-12 “数据库属性”对话框

③ 选中要删除的数据文件“教学练习_data2”，单击页面下方的“删除”按钮，出现“确定删除数据文件吗？”询问对话框，单击“确定”按钮。

在“教学练习属性”对话框中选择文件类型“日志”，也可以修改日志文件的属性。

方法二：使用 Transact-SQL 命令

语法格式：

```
ALTER DATABASE 数据库名
    { ADD FILE < filespec > [ ,...n ] [ TO FILEGROUP 文件组名 ]
    | ADD LOG FILE < filespec > [ ,...n ]
    | REMOVE FILE 数据库逻辑名
    | ADD FILEGROUP 文件组名
    | REMOVE FILEGROUP 文件组名
    | MODIFY FILE < filespec >
    | MODIFY NAME=数据库名
    |MODIFY FILEGROUP 文件组名{NAME=新文件组名 } }
```

参数说明：

① filespec 参数定义如下。

```
( NAME =数据库逻辑名
    [ , NEWNAME =新数据库逻辑名]
    [ , FILENAME = '数据库物理文件名' ]
    [ , SIZE = size ]
    [ , MAXSIZE = { max_size | UNLIMITED } ]
    [ , FILEGROWTH = growth_increment ] )
```

② ADD FILE 指定要添加的文件。

③ TO FILEGROUP 指定要将指定文件添加到文件组。

- ④ ADD LOG FILE 指定要将日志文件添加到的指定数据库。
- ⑤ REMOVE FILE 从数据库系统表中删除文件描述并删除物理文件。只有在文件为空时才能删除。
- ⑥ ADD FILEGROUP 指定要添加的文件组。
- ⑦ REMOVE FILEGROUP 从数据库中删除文件组并删除该文件组中的所有文件。只有在文件组为空时才能删除。
- ⑧ MODIFY FILE 指定要更改的指定文件，更改选项包括 FILENAME、SIZE、FILEGROWTH 和 MAXSIZE。一次只能更改这些属性中的一种。必须在<filespec>中指定数据库逻辑名，以标识要更改的文件。如果指定了 SIZE，那么新大小必须比文件当前大小要大。
- ⑨ SIZE 指定数据库的初始容量大小。
- ⑩ MAXSIZE 指定操作系统文件可以增长到的最大尺寸。
- ⑪ FILEGROWTH 指定文件每次增加容量的大小，当指定数据为 0 时，表示文件不增长。

【例 3-4】修改“教学练习”数据库。

主数据文件的最大文件大小为文件增长不受限制。增加次数据文件教学练习_add，分配空间 5MB，不允许自动增长空间。删除教学练习_data1 次数据文件。

修改教学练习数据库的程序清单如下：

```
ALTER DATABASE 教学练习          --修改教学练习数据库
MODIFY FILE (                    --修改数据文件教学练习_data
    NAME='教学练习_data',
    MAXSIZE=UNLIMITED
)
GO
ALTER DATABASE 教学练习
ADD FILE (                        --增加数据文件教学练习_add
    NAME='教学练习_add',
    FILENAME='d:\server\mssql\data\教学练习_add.mdf',
    SIZE=5,
    FILEGROWTH=0                  --不允许自动增长
)
GO
ALTER DATABASE 教学练习
REMOVE FILE 教学练习_data2      --删除次数据文件教学练习_data2
GO
ALTER DATABASE 教学练习
ADD LOG FILE (                   --增加日志文件教学练习_addlog
    NAME='教学练习_addlog',
    FILENAME='d:\server\mssql\data\教学练习_addlog.ldf',
    SIZE=1,
    MaxSIZE=10,
    FILEGROWTH=1
)
GO
```

输出结果如下：

文件'教学练习_data2' 已删除。

2. 收缩数据库

SQL Server 提供收缩过于庞大的数据库的手段，以收回未使用的数据页面。可以用手动的方法单

独收缩某一数据库文件，也可以收缩整个文件组的长度，还可以设置数据库在达到一定大小前自动执行收缩操作，自动收缩操作是在后台运行的，不会影响当前前台的任何活动。

注意：不能将数据库收缩到小于创建的长度。日志文件不可以被收缩。

收缩数据库的方法有使用对象资源管理器和使用 Transact-SQL 命令两种。

使用 SQL 命令收缩数据库的**语法格式**如下：

```
DBCC SHRINKDATABASE
    (database_name [,target_percent]
    [{,}{NOTRUNCATE | TRUNCATEONLY}]
    )
```

参数说明：

- target_percent: 表示当数据库收缩后还剩下的自由空间。
- NOTRUNCATE: 被释放的文件空间还保持在数据库文件的范围内，否则释放的空间被系统收回。
- TRUNCATEONLY: 将所有未使用的数据空间释放给操作系统使用。使用该关键字，将忽略 target_percent 限制。

【例 3-5】收缩教学练习数据库文件，使使用空间为原来的 40%。

方法一：使用对象资源管理器

① 在对象资源管理器的“教学练习”数据库结点上单击右键，在出现的快捷菜单中选择“所有任务”→“收缩数据库”，出现“收缩数据库”对话框，如图 3-13 所示。

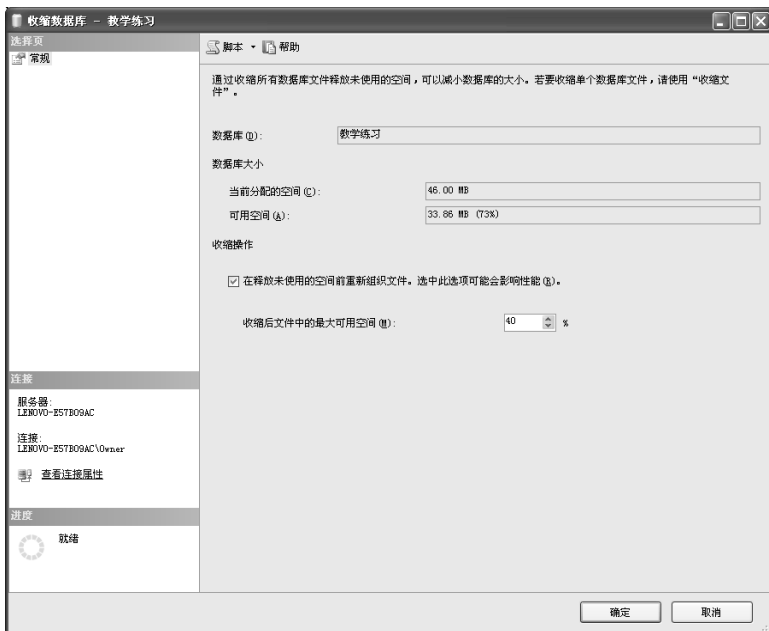


图 3-13 “收缩数据库”对话框

② “收缩后文件中的最大可用空间”表示用于设置压缩后文件的最大空闲空间。我们在此输入 40。

③ 在对象资源管理器的“教学练习”数据库结点上单击右键，在出现的快捷菜单中选择“所有任务”→“收缩文件”，同样出现“收缩文件”对话框。

④ 因为一个数据库往往不是一个数据文件，所以单击“文件”按钮可以选择数据库的某个数据文件单独进行收缩。

⑤ 单击“确定”按钮。

方法二：使用 Transact-SQL 命令

语句如下：

```
DBCC SHRINKDATABASE (教学练习, 40)
GO
```

返回结果：

```
DBCC 执行完毕。
```

3. 查看数据库信息

(1) 查看数据库定义信息

程序如下：

```
sp_helpdb --返回所有定义的数据库信息
sp_helpdb 教学练习 --返回指定数据库的定义信息
```

结果如图 3-14 所示。

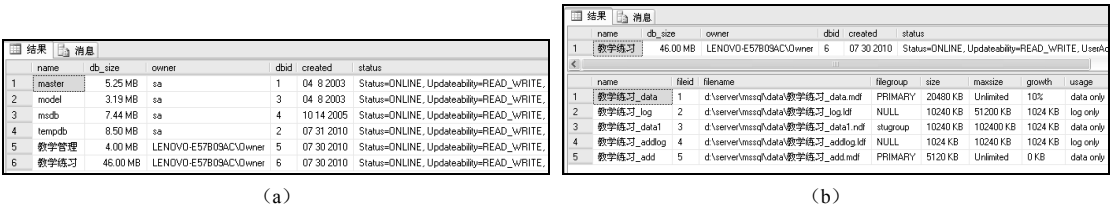


图 3-14 查看数据库定义信息

图 3-14 (a) 所示为数据库的信息，图 3-14 (b) 所示为库内文件的信息。

(2) 查看数据空间使用状况

程序如下：

```
USE 教学练习
GO
sp_spaceused --返回教学练习数据库数据文件使用的空间
GO
DBCC SQLPERF (LOGSPACE) --返回所有数据库的日志文件使用的空间
```

4. 删除数据库

【例 3-6】 删除数据库教学练习（注：因为后面还要使用该数据库，故先不要真正删除）。

方法一：使用对象资源管理器

操作步骤如下：

- ① 在对象资源管理器的“教学练习”数据库节点上单击右键，在出现的快捷菜单中选择“删除”。
- ② 在弹出的对话框中选择“是”按钮。

方法二：使用 Transact-SQL 命令

利用 DROP 语句删除数据库。DROP 语句可以从 SQL Server 中一次删除一个或多个数据库。其语句如下：

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

例如，删除创建的数据库“教学练习”的语句如下：

```
DROP DATABASE 教学练习 --教学练习中所包含的文件都被删除
```


实验与思考

目的和任务

- (1) 熟悉 SQL Server 数据库存储原理，并能进行数据库在不同存储设备上的存储规划。
- (2) 掌握 SQL Server 数据库的创建方法。
- (3) 掌握查看、修改数据库属性的方法。
- (4) 掌握数据库删除的方法。

实验内容

- (1) 用对象资源管理器创建数据库。

数据库名: xmgl。

数据文件 1 的逻辑名为 xmgl1, 物理名为 xmgl1.mdf, 存放在“D:\xmgl”目录下, 初始大小为 1 MB, 增长方式为自动增长, 每次增加 1 MB。

数据文件 2 的逻辑名为 xmgl2, 物理名为 xmgl2.ndf, 存放在与主数据文件相同的目录下; 文件大小为 3 MB; 增长方式为自动增长, 每次增加 10%。

日志文件 1 的逻辑名为 xmrz1, 物理名为 xmrz1.ldf, 存放在“D:\xmrz”目录下, 初始大小为 1 MB, 增长方式为自动增长, 每次增加 10%。

- (2) 用对象资源管理器查看和修改数据库, 将 xmgl1 的增长每次增加 1MB 修改为 10%。
- (3) 用 SQL 命令创建数据库, 数据库要求同上。
- (4) 用 SQL 命令修改数据库。

在 xmgl 中增加一个名为“xmgl3”的数据文件, 在“F:\xmgl”目录下增加“xmgl3.ndf”文件, 初始大小为 3 MB, 增长方式为自动增长, 每次增加 15%。删除“xmgl2”。

将数据库“xmgl”中的文件 xmgl2 和刚增加的 xmgl3 删掉。

- (5) 删除数据库。

读者自己可以创建另外一个数据库, 用企业管理器和 SQL 命令“DROP database 数据库名”分别练习。

问题思考

- (1) 数据库在磁盘上的文件组织分几种类型? 其中哪个是必需的?
- (2) 创建的数据库文件默认存放在磁盘的什么位置?
- (3) 使用 SQL 命令创建数据库时, 语句中的每个参数是否都必须给出?