

经典 SQL. 2000 教程

实验一

1.1 SQL Server 2000 安装简介

SQL Server 2000 是微软公司最新版的大型数据库服务器，其性能指标在各方面都有赶超 Oracle 数据库的趋势。在经历了 SQL Server 6.5 和 7.0 两个版本的尝试后，微软公司终于开始向大规模的业务领域进发了。

1.1.1 硬件和操作系统要求

一、硬件要求（最低要求）

计算机：Pentium 166 MHz 或更高；

内存：至少 64MB，建议 128 MB 或更多；

硬盘空间：需要约 500MB 的程序空间，以及预留 500M 的数据空间；

显示器：需要设置成 800x600 模式，才能使用其图形分析工具。

二、操作系统要求

下面说明 SQL Server 2000 各种版本对操作系统的要求：

企业版

- ① Microsoft Windows NT Server 4.0
- ② Microsoft Windows NT Server 4.0 企业版
- ③ Windows 2000 Server
- ④ Windows 2000 Advanced Server
- ⑤ Windows 2000 Data Center Server

（以上所有版本均需安装 IE5.0 以上版本浏览器）

标准版

- ① Microsoft Windows NT Server 4.0

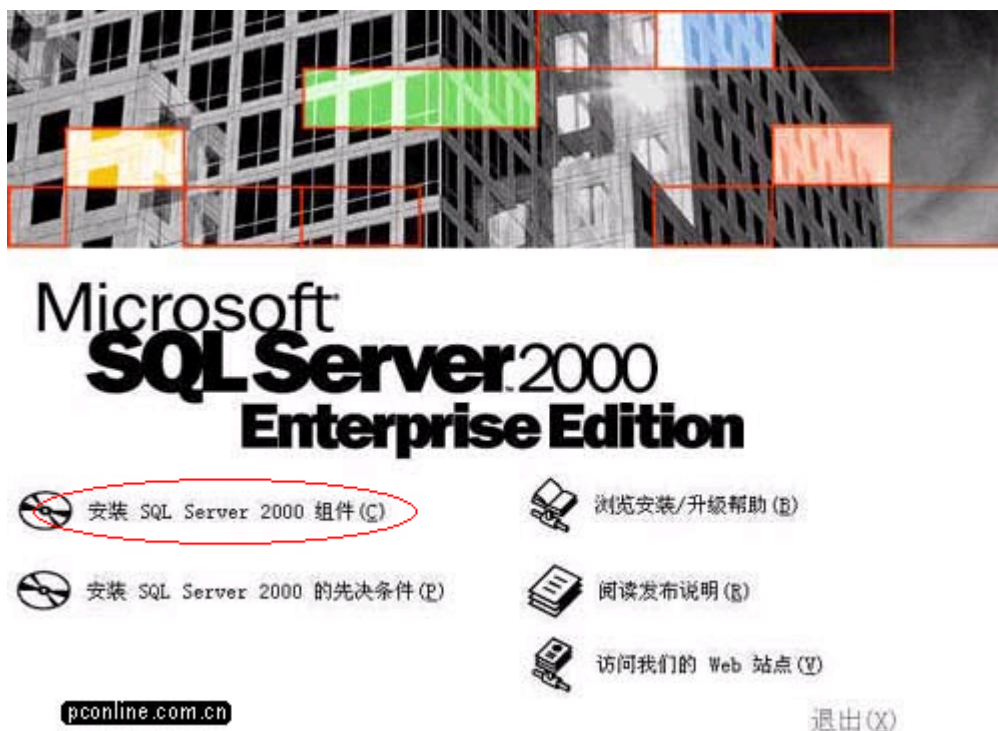
- ② Windows 2000 Server
- ③ Microsoft Windows NT Server 企业版
- ④ Windows 2000 Advanced Server
- ⑤ Windows 2000 Data Center Server
- 个人版
- ① Microsoft Windows Me
- ② Windows 98
- ③ Windows NT Workstation 4.0
- ④ Windows 2000 Professional
- ⑤ Microsoft Windows NT Server 4.0
- ⑥ Windows 2000 Server
- ⑦ 所有更高级的 Windows 操作系统

注：SQL Server 2000 的某些功能要求在Microsoft Windows 2000 Server以上的版本才能运行。因此大家安装Windows Server 2000(建议为Advanced版本)，可以学习和使用到SQL Server 2000 的更多功能，以及享受更好的性能。

1.1.2 详细的安装过程

下面以 Windows 2000 Server 操作系统作为示例，详细介绍安装 SQL Server 2000 企业版的过程。

将企业版安装光盘插入光驱后，出现以下提示画面，在下面窗口中选择“安装 SQL Server 2000 组件”。



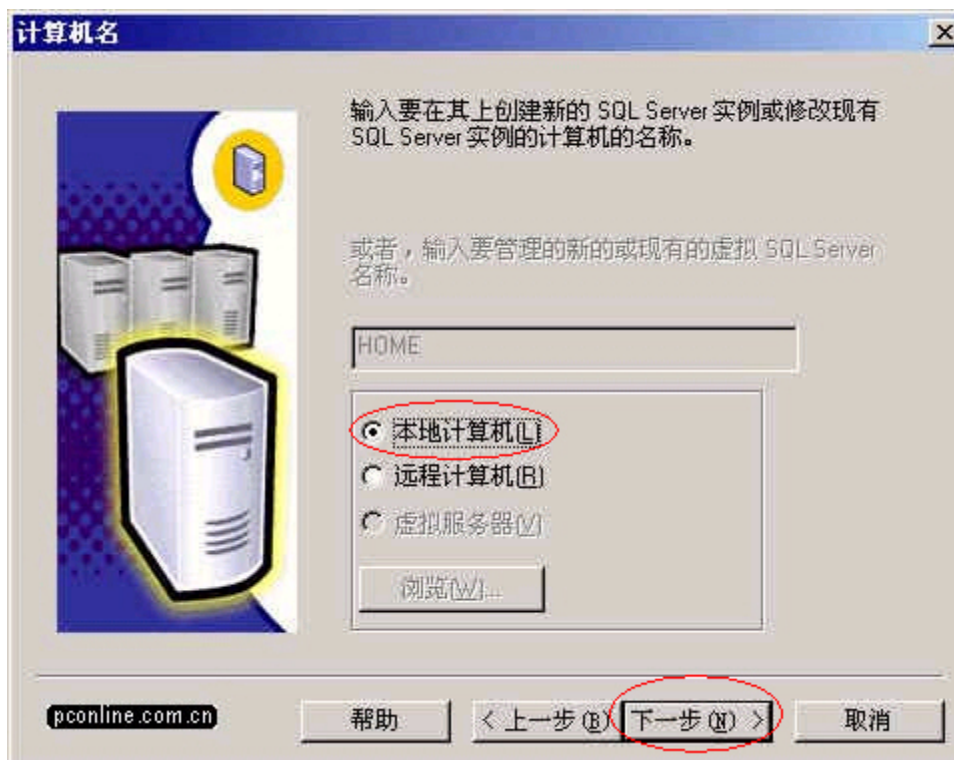
在下面窗口中，选择 “安装数据库服务器”。



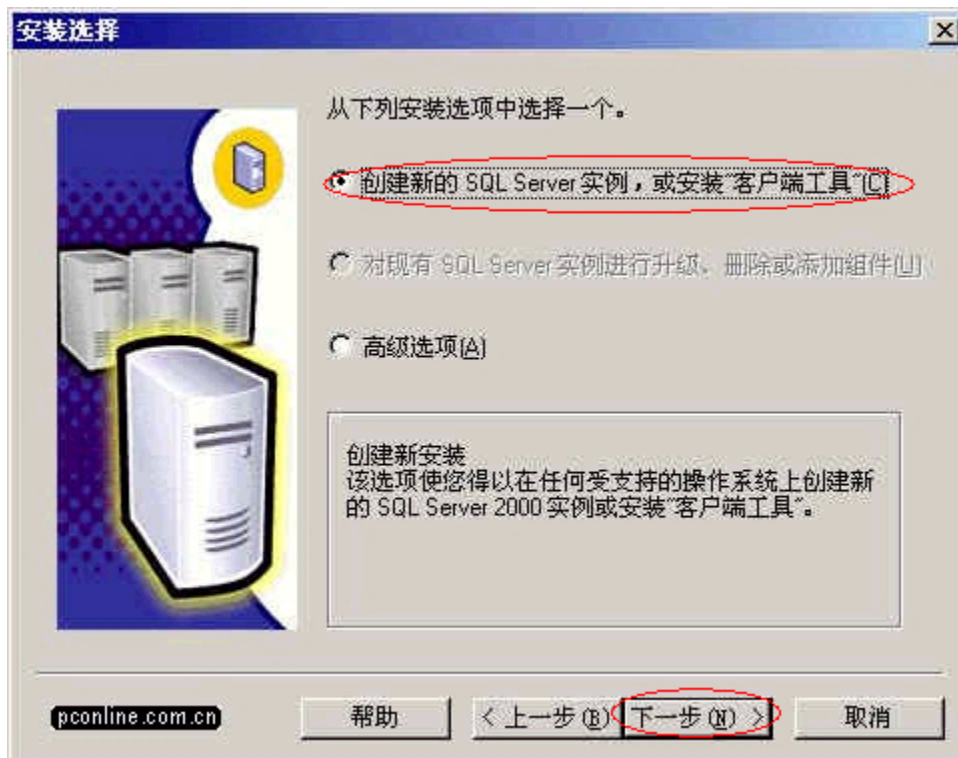
在下面窗口中，选择 “下一步”。



在下面窗口中，选择“本地计算机”，再选择“下一步”。



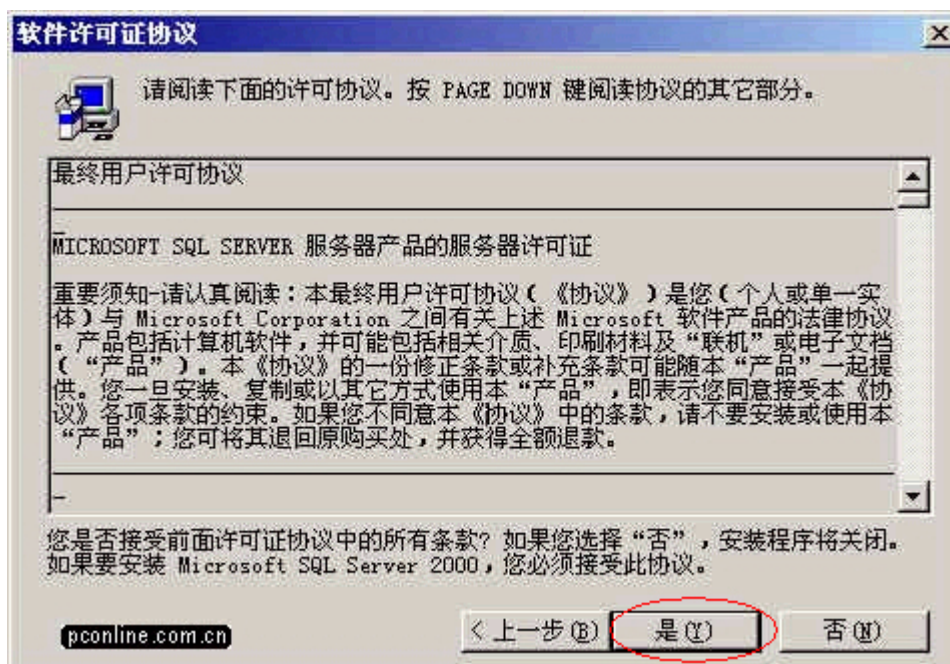
在下面窗口中，选择“创建新的 SQL Server 实例...”，然后选择“下一步”。



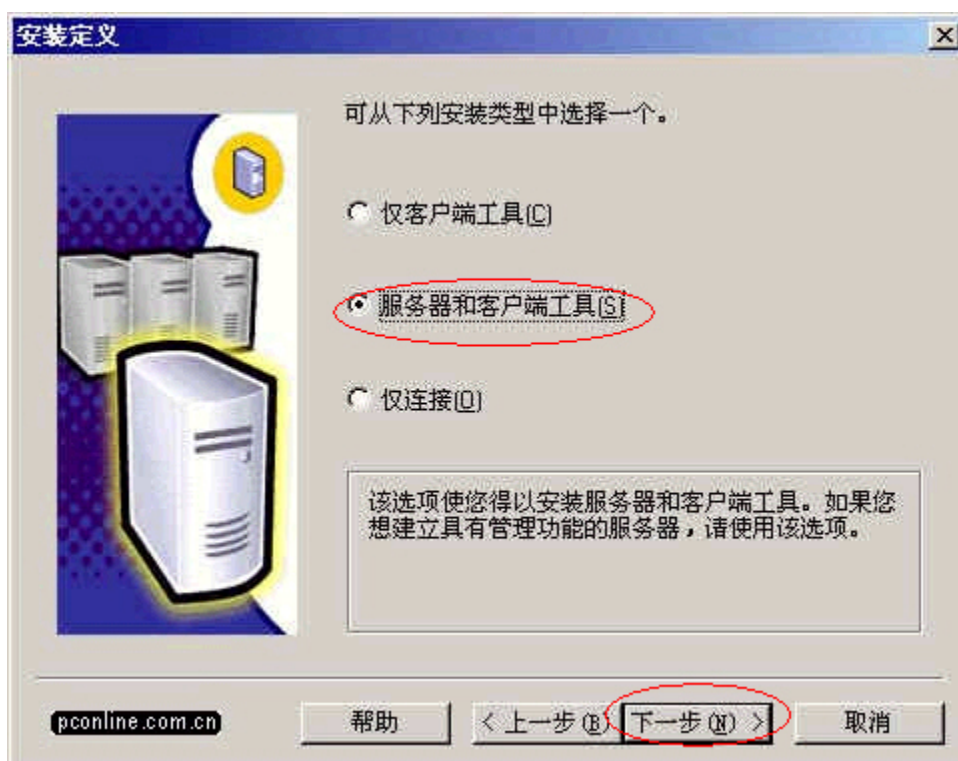
在下面窗口中，输入用户信息，然后选择“下一步”。



在下面窗口中，接受软件许可证协议。

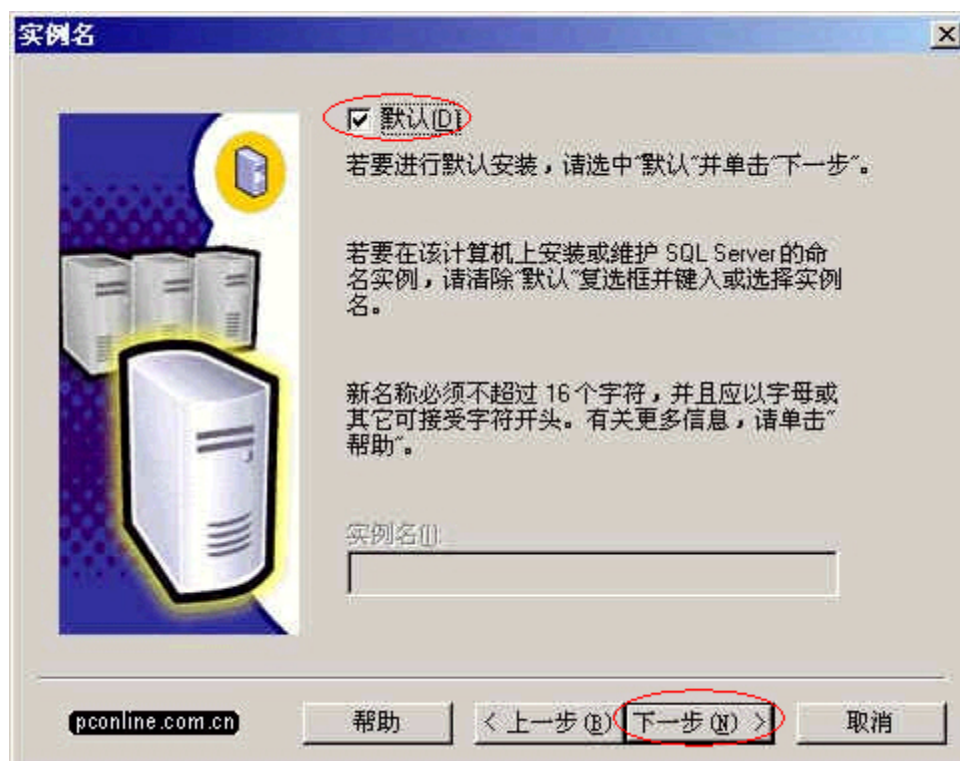


在下面窗口中，选择“服务器和客户端工具”选项进行安装。我们需要将服务器和客户端同时安装，这样在同一台机器上，我们可以完成相关的所有操作，对于我们学习 SQL Server 很有用处。如果你已经在其它机器上安装了 SQL Server，则可以只安装客户端工具，用于对其它机器上 SQL Server 的存取。

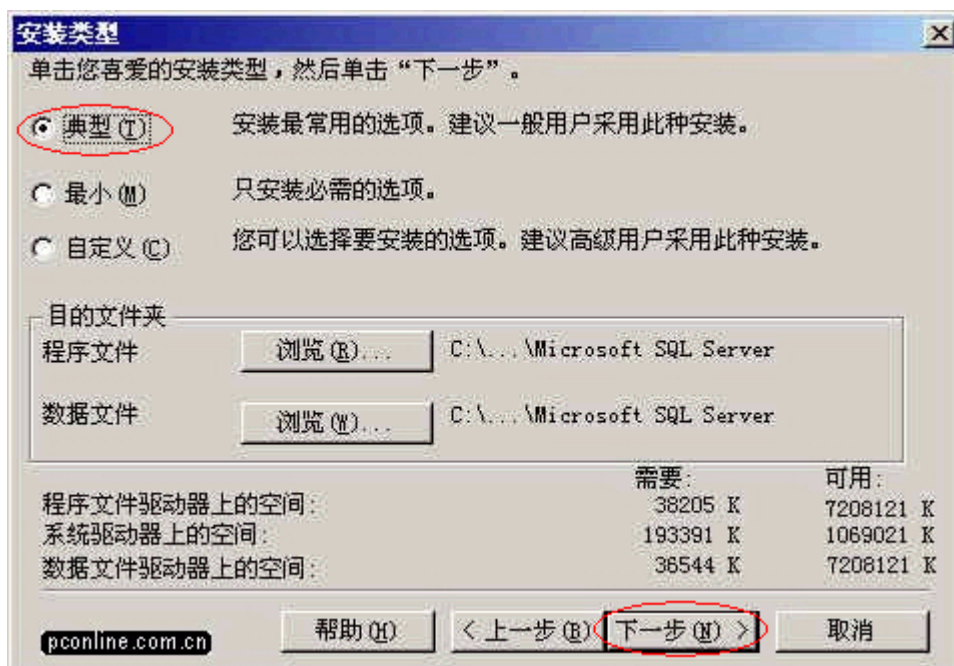


在下面窗口中，选择“默认”的实例名称。这时本 SQL Server 的名称将和 Windows 2000 服务器的名称相同。例如 Windows 服务器名称是“LWHNOTE”，则 SQL Server 的名字也

是“LWHNOTE”。SQL Server 2000 可以在同一台服务器上安装多个实例，也就是你可以重复安装几次。这时您就需要选择不同的实例名称了。建议将实例名限制在 10 个字符之内。实例名会出现在各种 SQL Server 和系统工具的用户界面中，因此，名称越短越容易读取。另外，实例名称不能是“Default”或“MSSQLServer”以及 SQL Server 的保留关键字等。



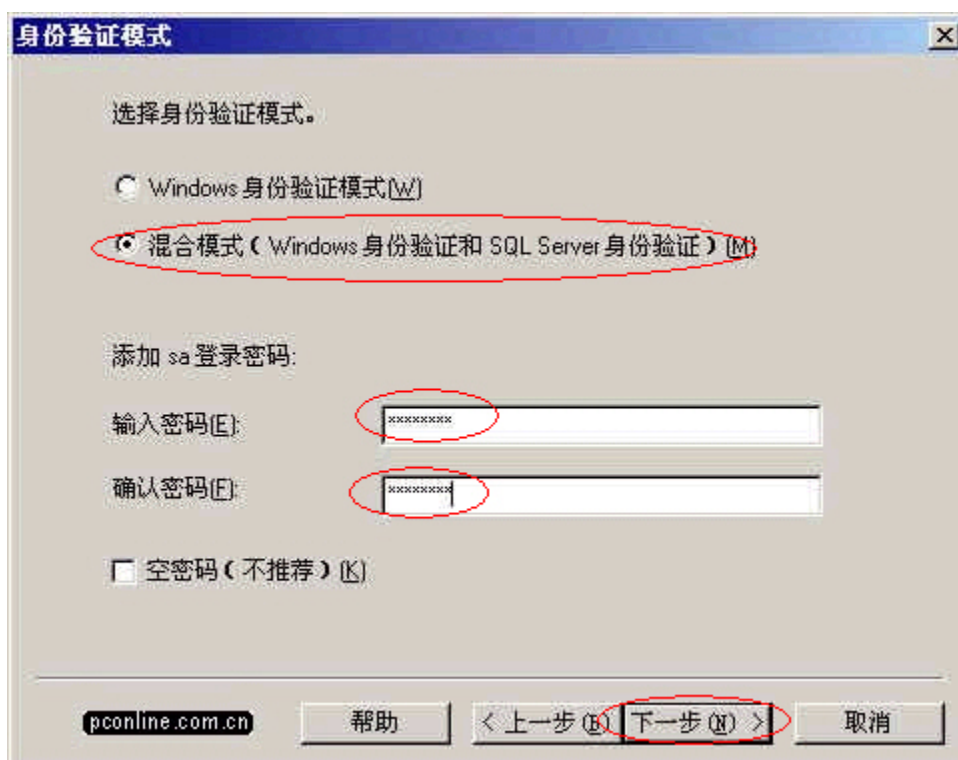
在下面窗口中，选择“典型”安装选项，并指定“目的文件夹”。程序和数据文件的默认安装位置都是“C:\Program Files\Microsoft SQL Server\”。注意，如果您的数据库数据有 10 万条以上的话，请预留至少 1G 的存储空间，以应付需求庞大的日志空间和索引空间。



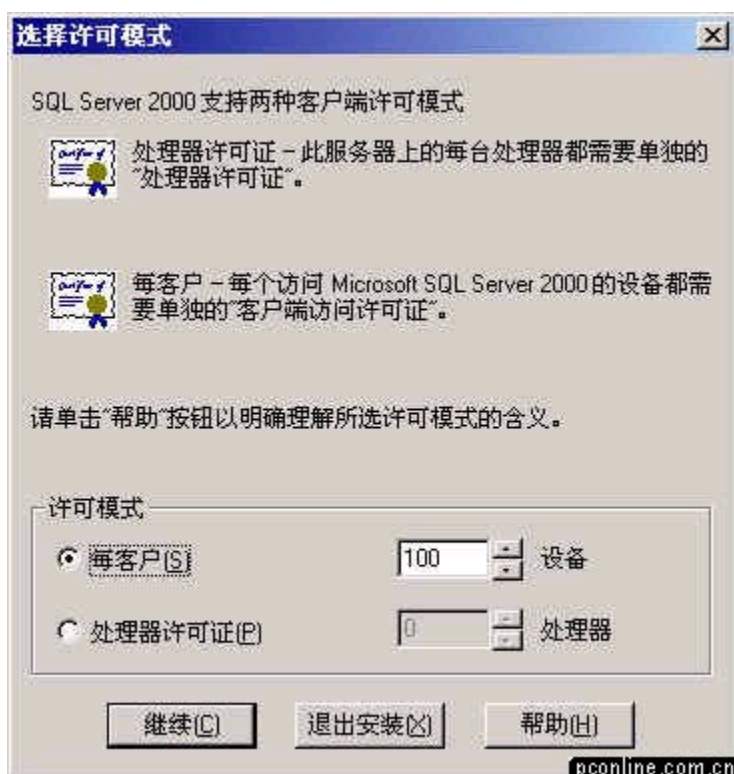
在“服务账号”窗口中，请选择“对每个服务使用统一账户...”的选项。在“服务设置”处，选择“使用本地系统账户”。如果需要“使用域用户账户”的话，请将该用户添加至 Windows Server 的本机管理员组中。

在“身份验证模式”窗口中，请选择“混合模式...”选项，并设置管理员“sa”账号的密码。如果您的目的只是为了学习的话，可以将该密码设置为空，以方便登录。如果是真正的应用系统，则千万需要设置和保管好该密码！如果需要更高的安全性，则可以选择

“Windows 身份验证模式”，这时就只有 Windows Server 的本地用户和域用户才能使用 SQL Server 了。



在“选择许可模式”窗口中，根据您购买的类型和数量输入（0 表示没有数量限制）。“每客户”表示同一时间最多允许的连接数，“处理器许可证”表示该服务器最多能安装多少个 CPU。笔者这里选择了“每客户”并输入了 100 作为示例。



然后就是约 10 分钟左右的安装时间，安装完毕后，出现该界面，并新增了以下的菜单。如果您是个初学者，就可以先打开“联机丛书”，做一个简单的学习了！



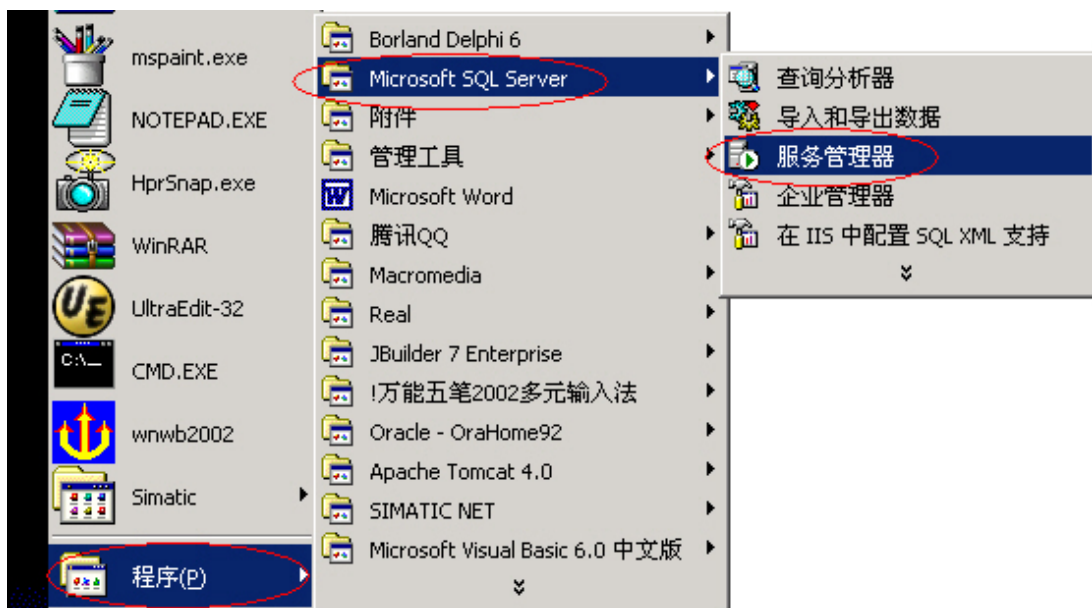
1.2 启动和停止 SQL Server

应用程序在使用 SQL Server 数据库之前，必须连接到一个 SQL Server 实例，如：

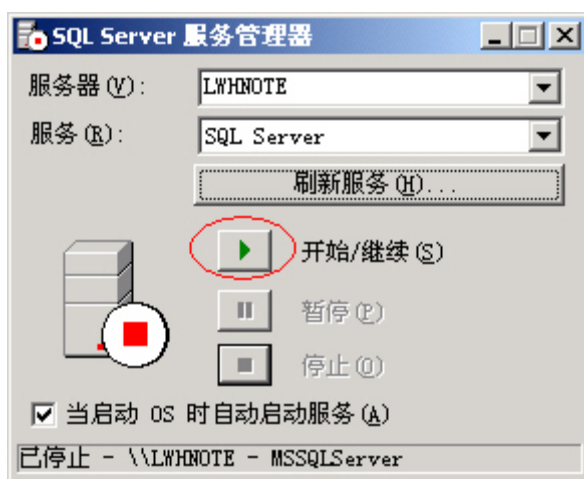
LWHNOTE。

一、用服务管理器启动和停止 SQL Server

选择菜单“程序|Microsoft SQL Server|服务管理器”。



弹出“SQL Server 服务管理器”，选择服务器（如：LWHNOTE）和服务选项（如：SQL Server），若没有启动，按下 ，即可启动 SQL Server，若已经启动，按下 ，即可停止 SQL Server。



启动 SQL Server 后，在任务栏的右边会显示正在运行的 SQL Server，即：



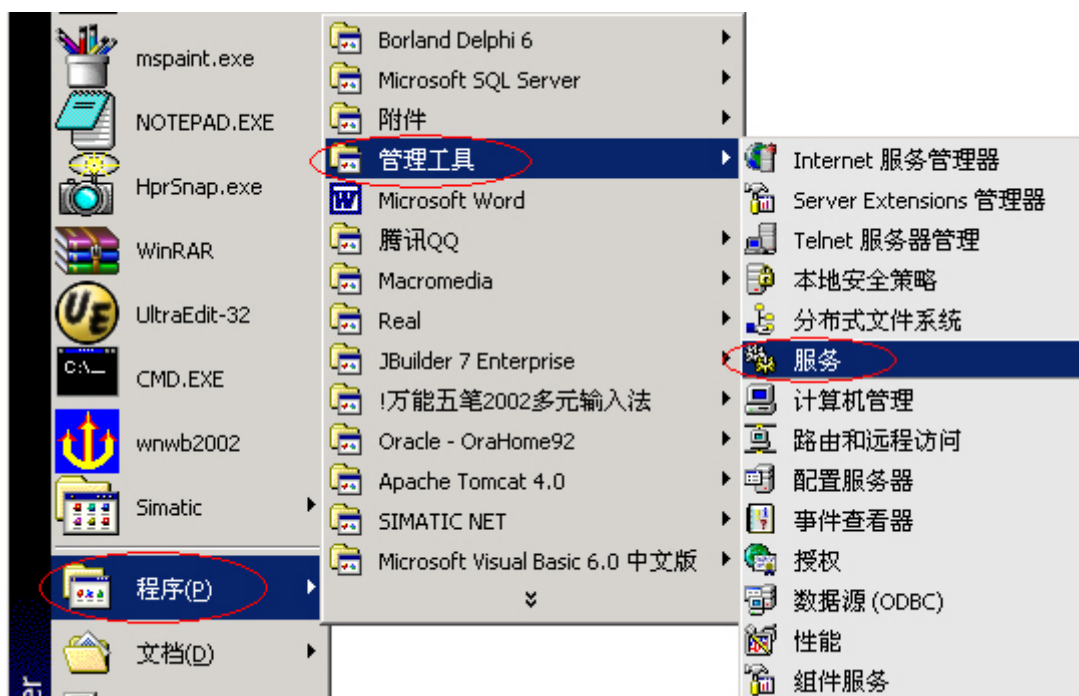
二、自动启动 SQL Server

使用手动启动弹出“SQL Server 服务管理器”后，选择复选框“当启动 OS 时自动启动服务”后，下次开机时系统即可自动启动 SQL Server。

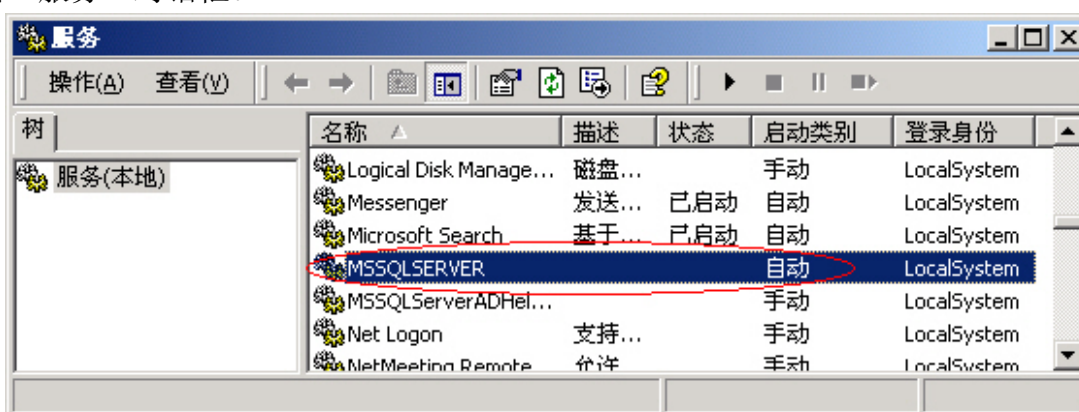


三、用操作系统控制面板中的服务管理程序启动和停止 SQL Server

选择菜单“程序|管理工具|服务”。

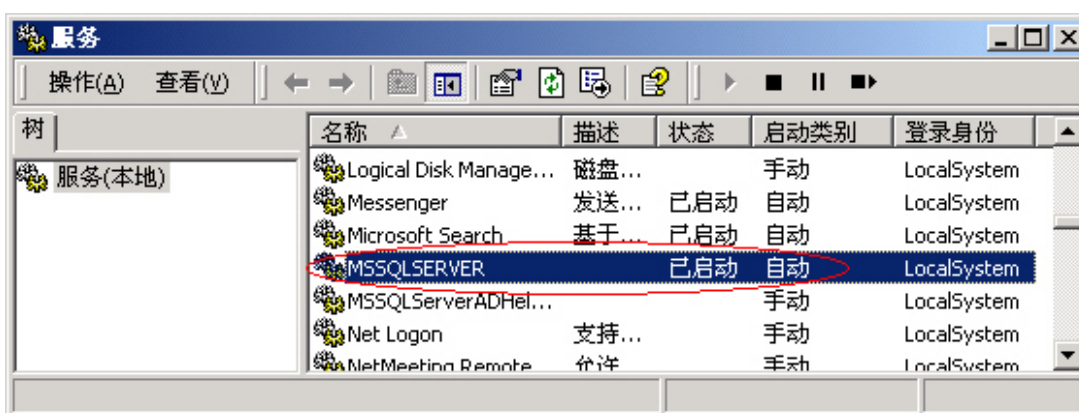


弹出“服务”对话框。



鼠标右击要启动的服务器名称，即：MSSQLSERVER，弹出菜单后，选择“启动”菜单项，

就可启动 SQL Server。



类似的方法，可以停止 SQL Server。

四、用命令方式启动和停止 SQL Server

用 net start mssqlserver 启动 SQL Server。



用 net stop mssqlserver 启动 SQL Server。



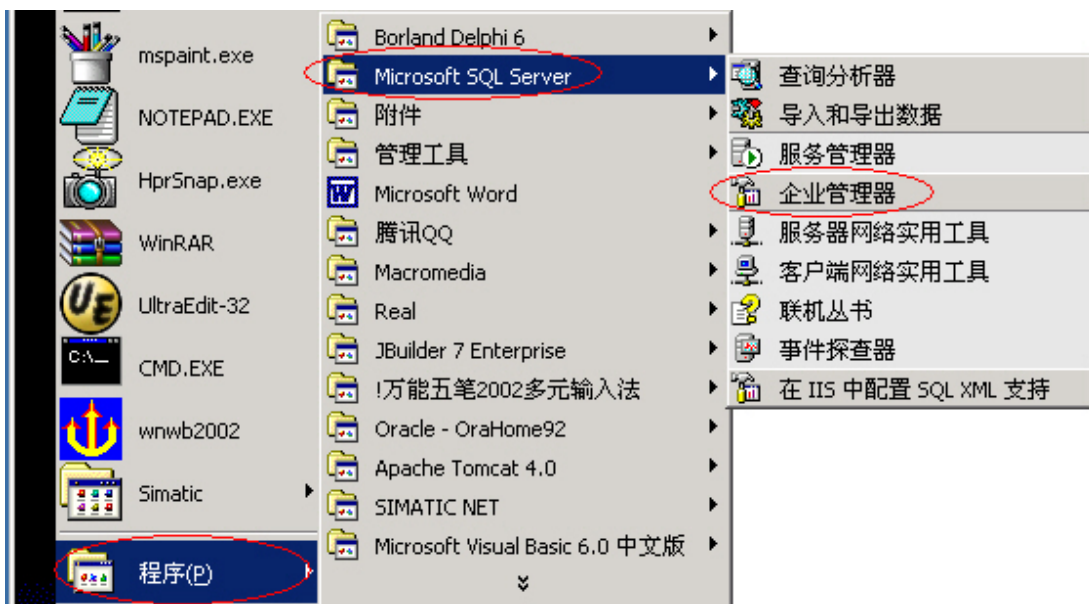
五、用企业管理器启动和停止 SQL Server

参见 1.3:使用 SQL Server 企业管理器。

1.3 使用企业管理器

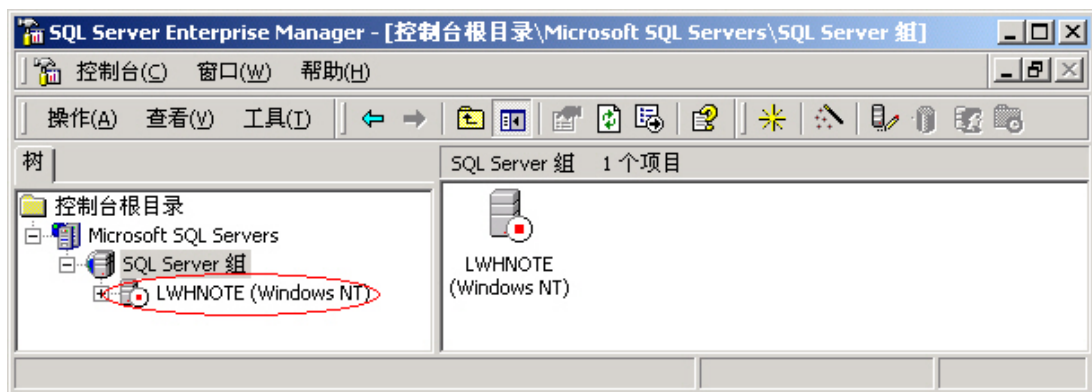
一、启动企业管理器

在 SQL Server 中，企业管理器(Enterprise Manager)是最重要最常用的管理工具。

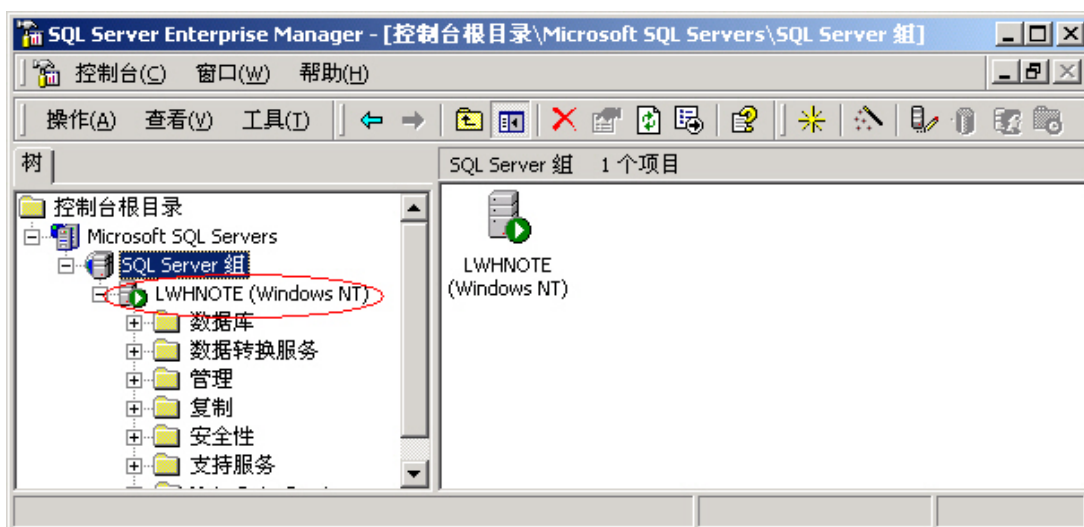


二、启动和停止 SQL Server

在企业管理器中所要启动的服务器上（如：LWHNOTE）鼠标右击，从快捷菜单中选择“连接”。



即可启动并连接。

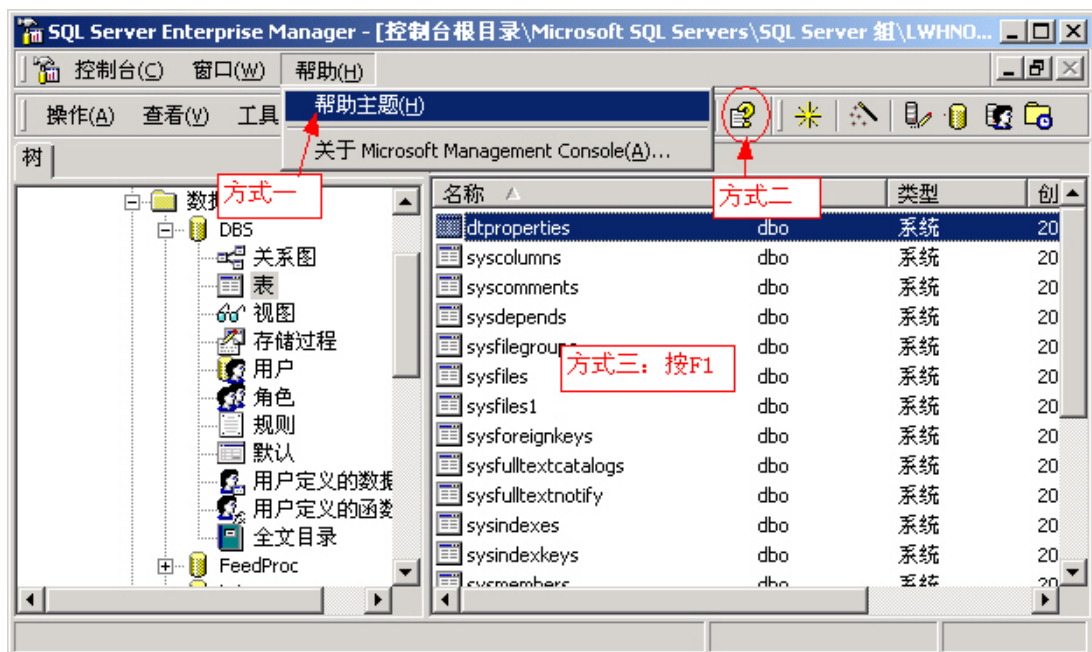


类似的方法，可以停止 SQL Server。

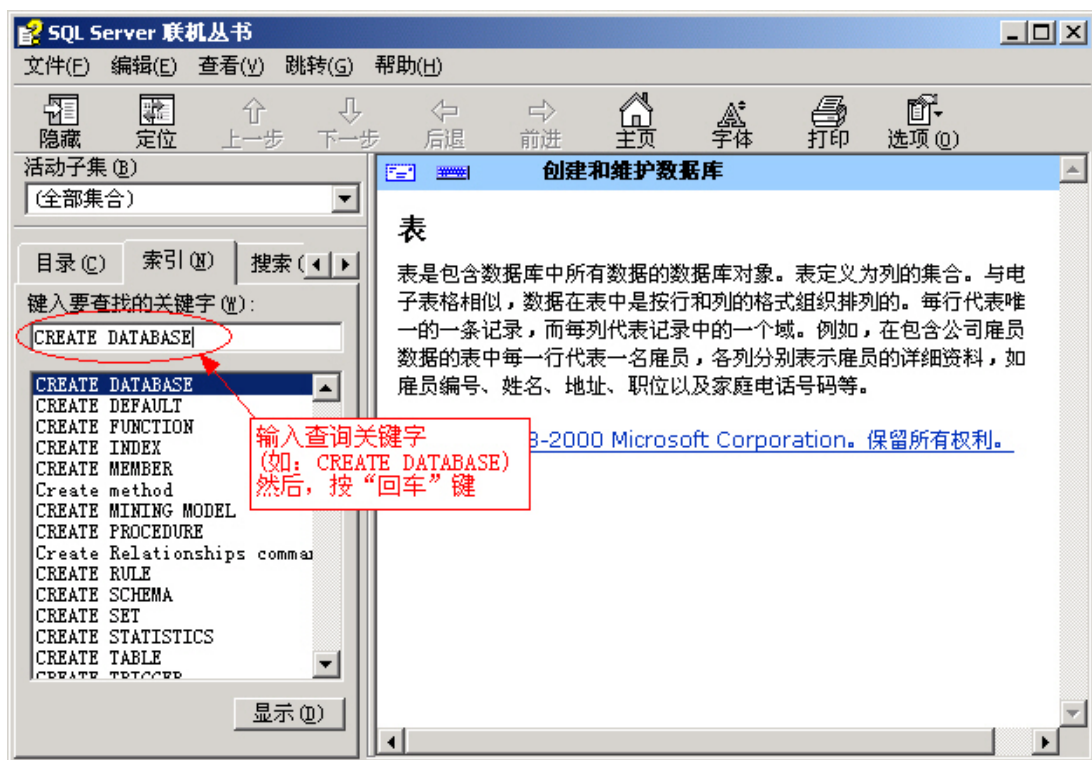
说明：企业管理器是一个对初学者来说最重要的管理工具，在以后的内容会经常使用。

三、使用帮助

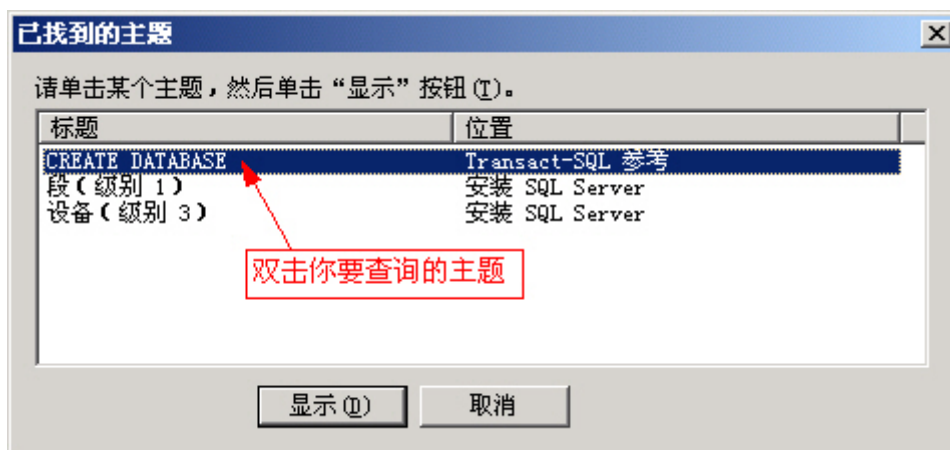
有三种方式可以启动帮助系统：使用帮助菜单、帮助按钮或按“F1”键。



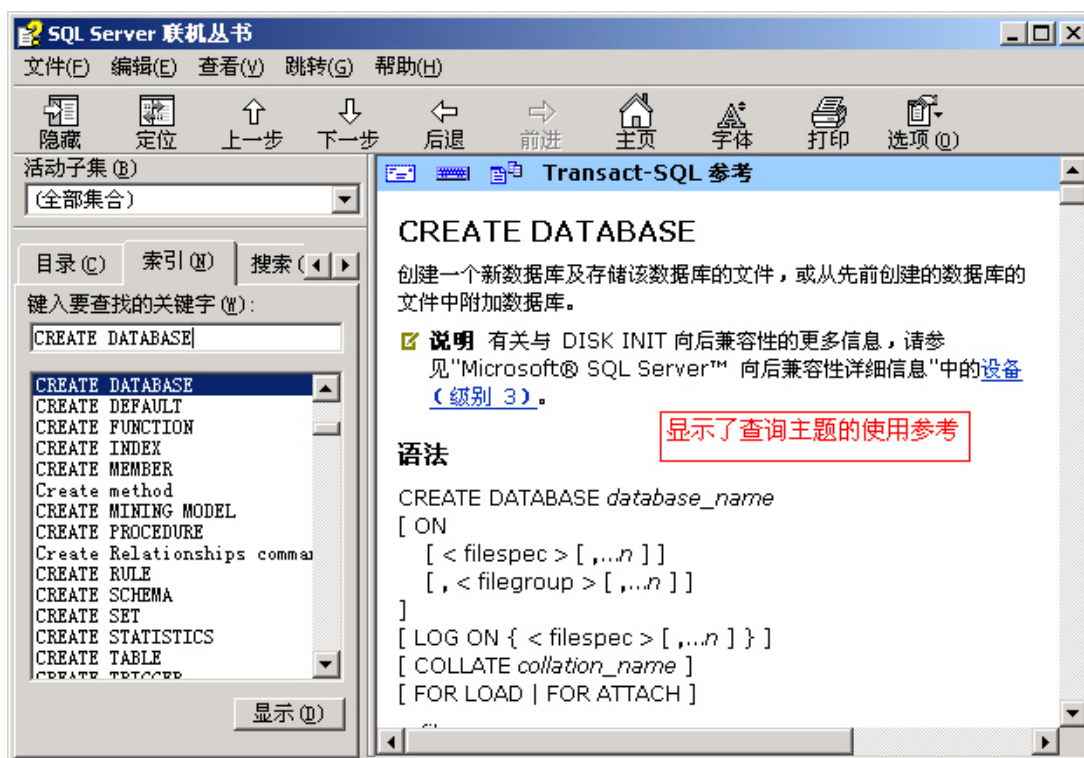
按如下方式输入要查询的关键字，然后按“回车”键



显示与查询关键字相关的查询主题，双击查询主题



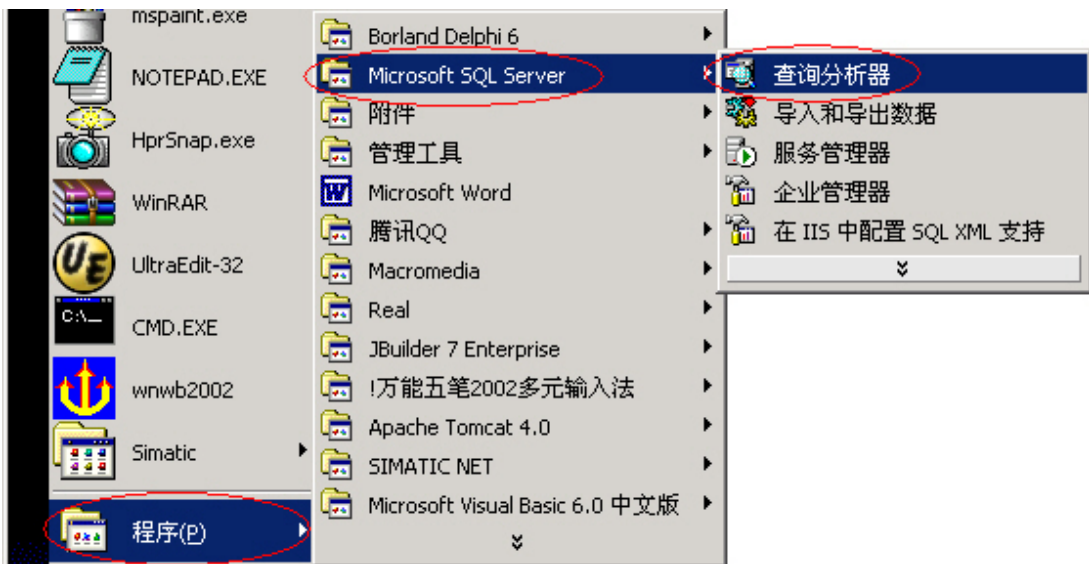
即可在窗口的右边显示相关查询主题的使用参考



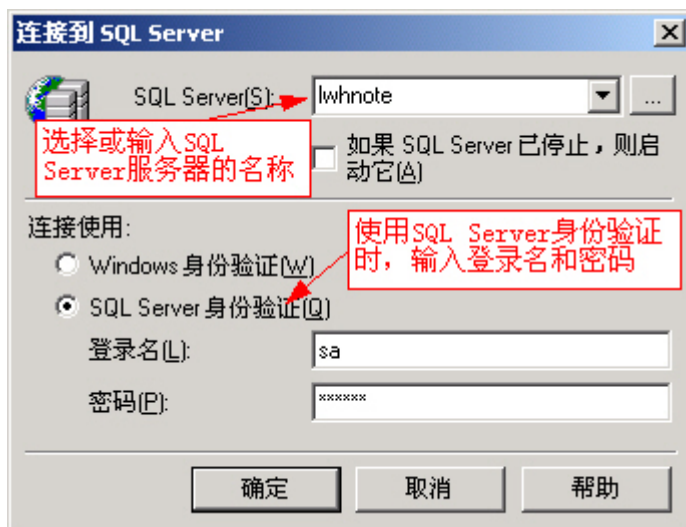
1.4 使用查询分析器

一、启动查询分析器

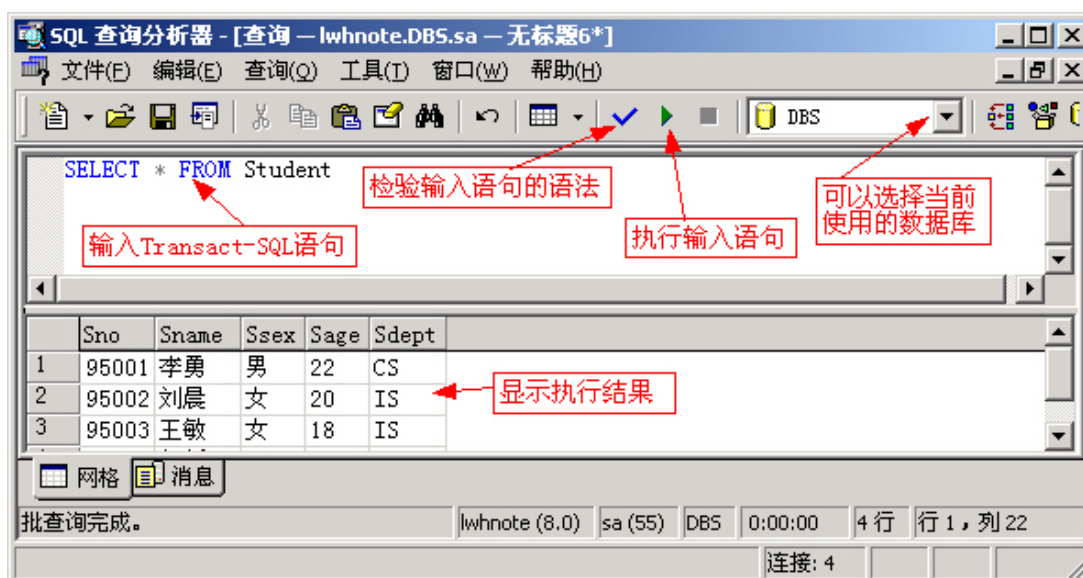
在 SQL Server 中，查询分析器也是最常用的管理工具之一。



选择“”菜单后，弹出“连接到 SQL Server 服务器”对话框。



用户输入 SQL Server 服务器的名称和身份验证后，即可进入查询分析器。注：身份验证可以使用“Windows 身份验证”或“SQL Server 身份验证”。



注：以后很多命令操作均可在查询分析器中完成，如：各种查询命令实验、SQL文件编写、存储过程和触发器文件的编写等。

1.5 数据库管理

1.5.1 创建数据库

帮助索引：<创建数据库-SQL Server|创建数据库>

一、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务日志文件。

A. 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**，用来存储主文件没有存储的其它数据。

B. 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。

C. 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和次文件组。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。

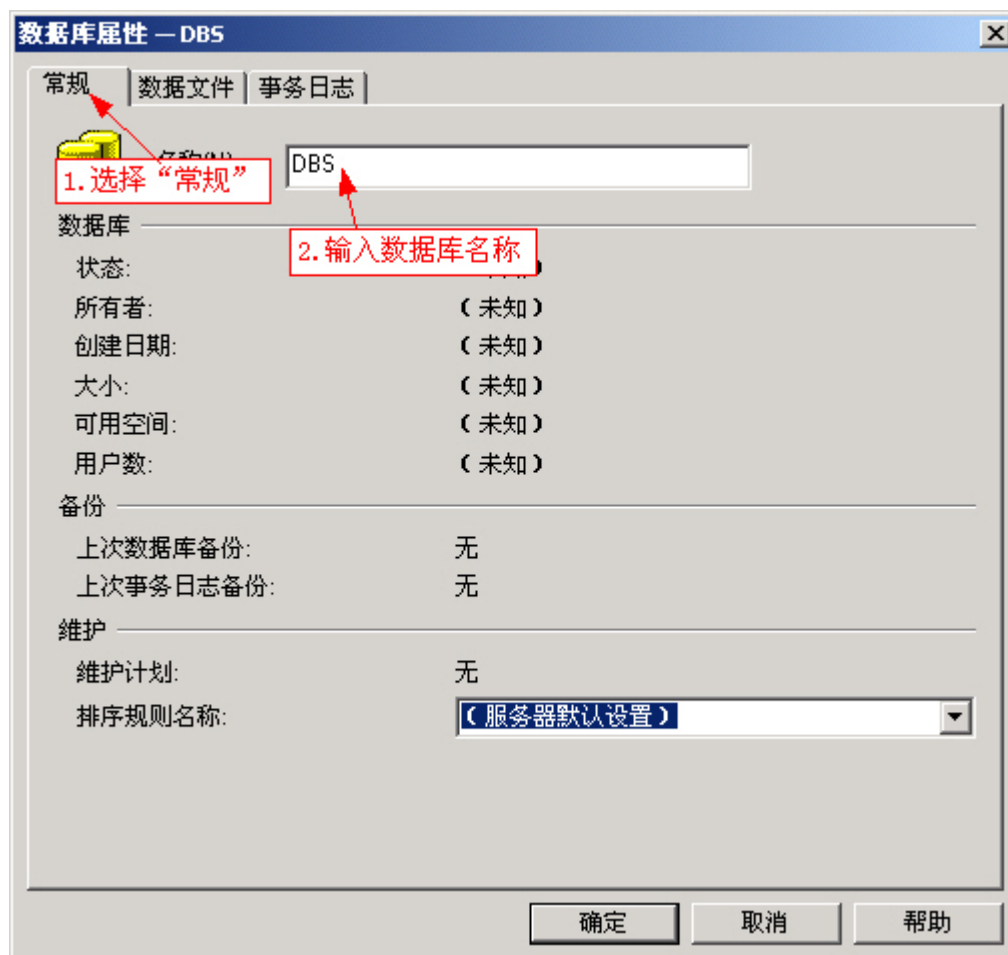
注：事务日志文件不属于任何文件组。

二、用企业管理器创建数据库

例 1：在服务器 LWHNOTE 中建立一个 DBS 的数据库，数据库文件和事务日志文件存储到 D:\SQL_DBS 文件夹中，数据库文件的初始容量设为 10MB。



按如下方式设置数据库名称



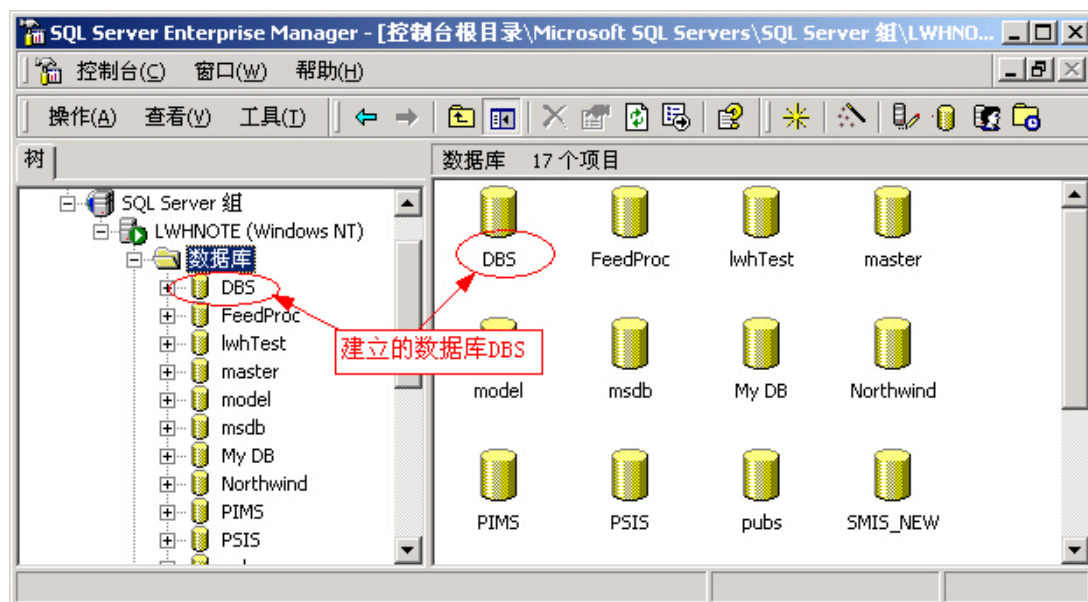
按如下方式设置数据库文件属性



按如下方式设置事务日志文件属性

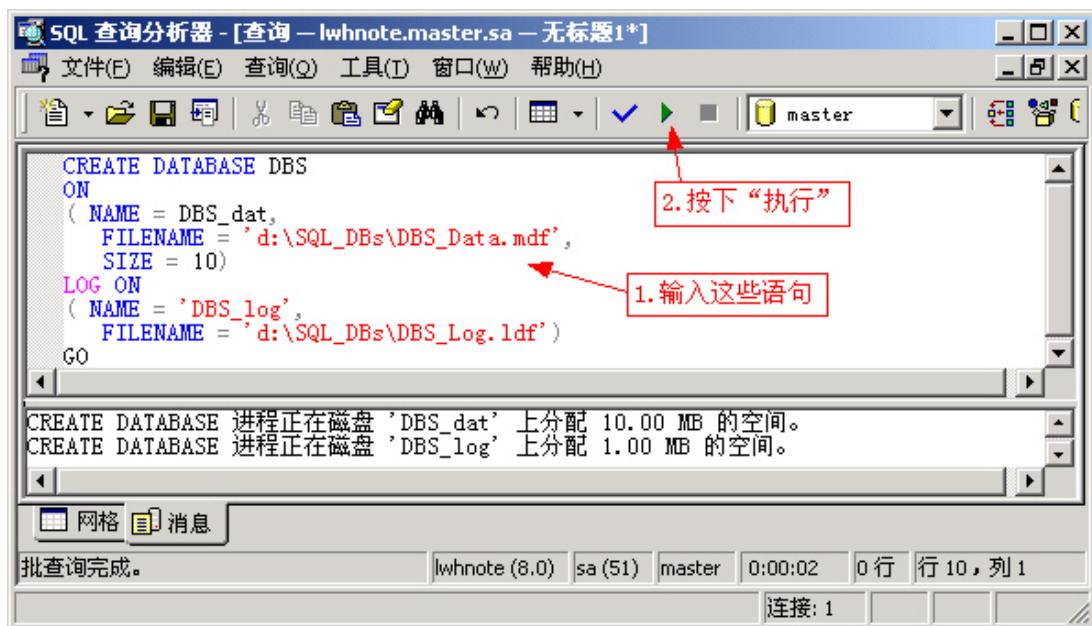


按“确定”按钮，即可建立数据库

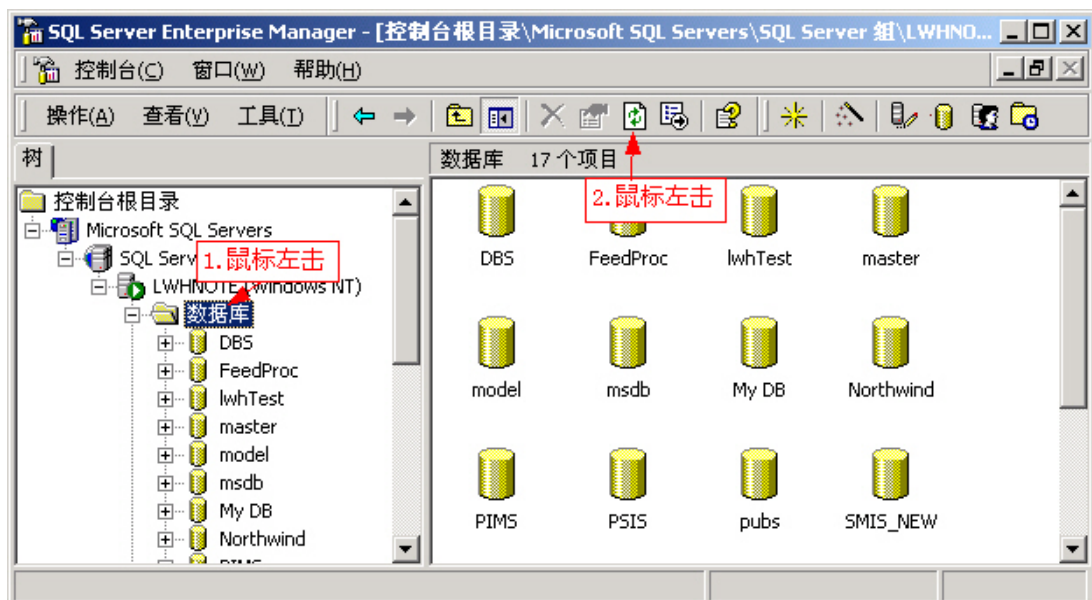


三、用查询分析器创建数据库

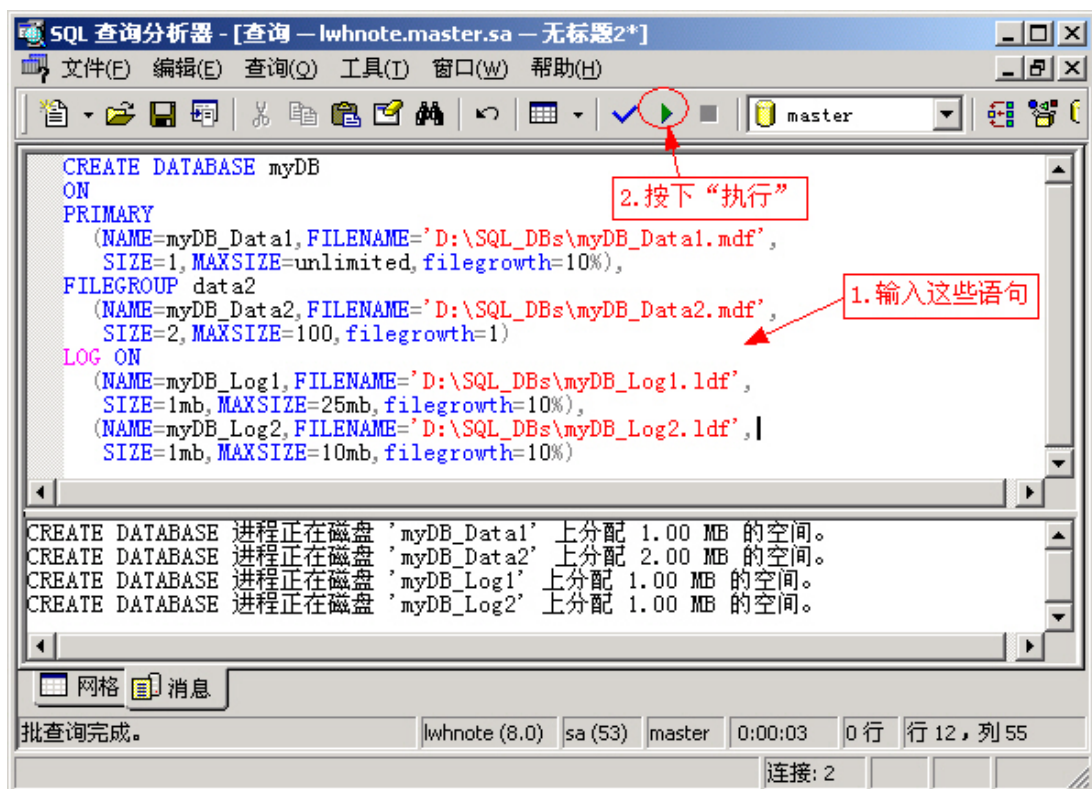
A. 用 CREATE DATABASE 创建一个类似企业管理器中创建的数据库。



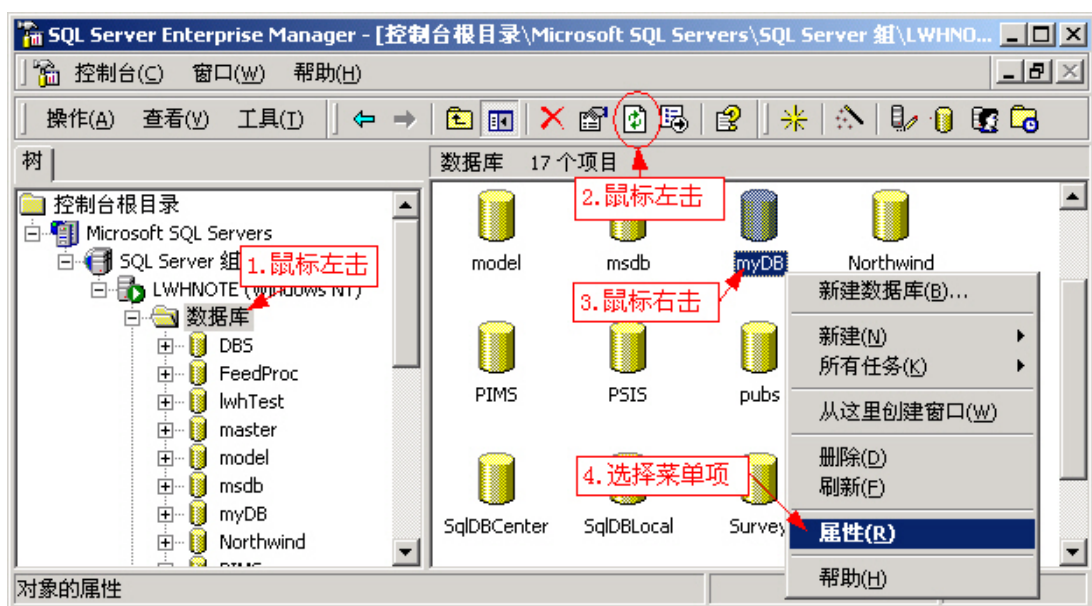
创建后，在企业管理器中可能看不见 DBS，如下方式即可



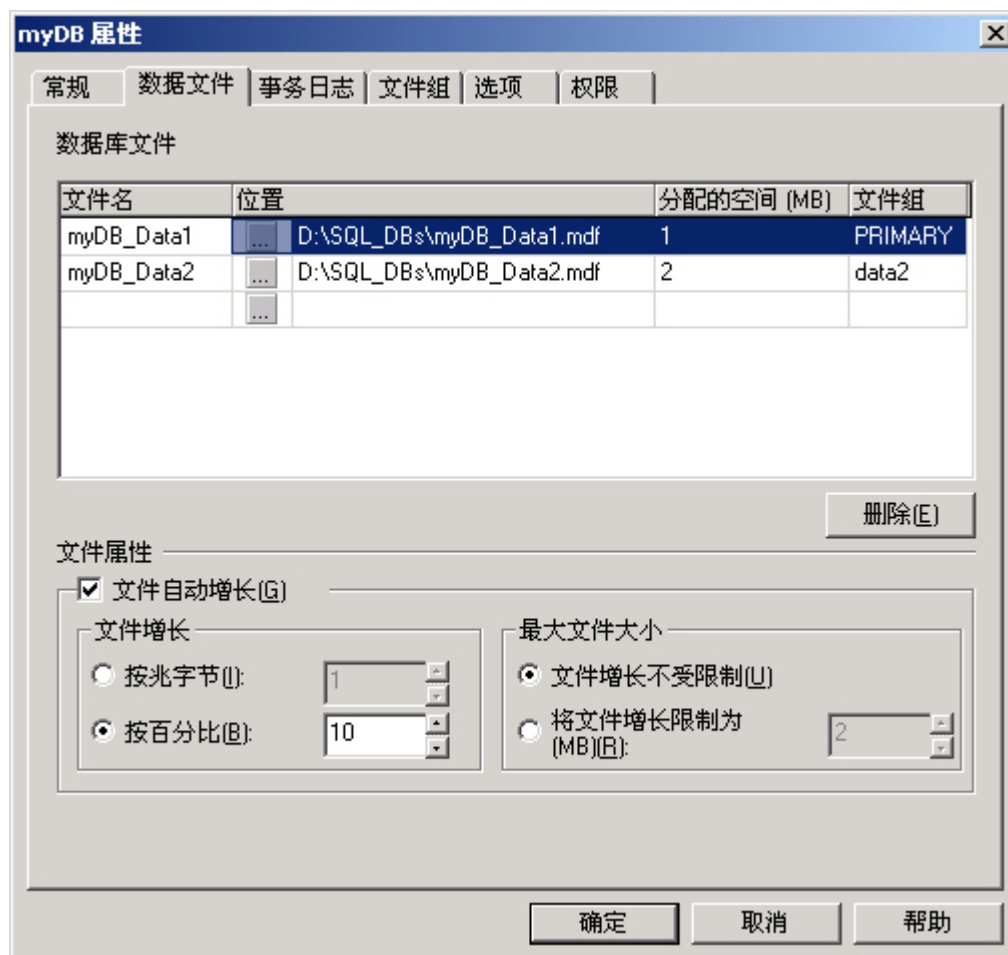
B. 用 CREATE DATABASE 创建一个多文件组的数据库。



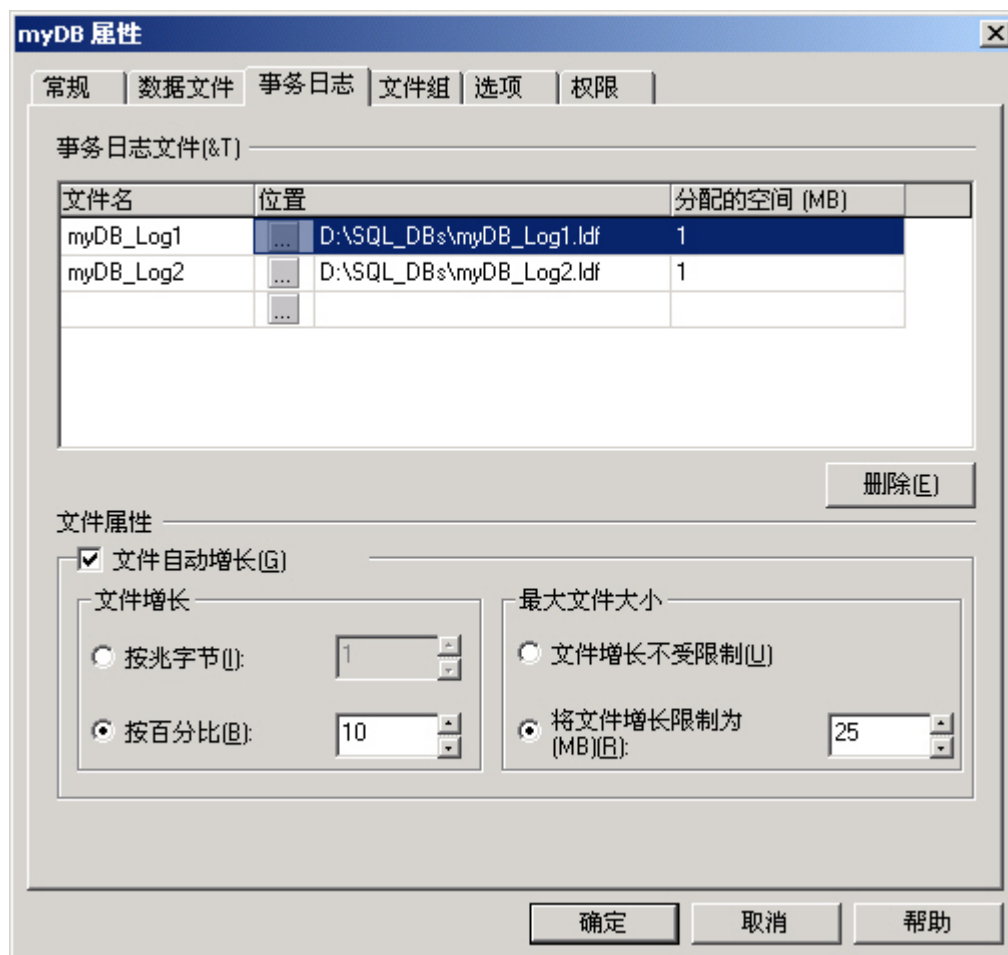
在企业管理器中查看数据库 myDB 的设置



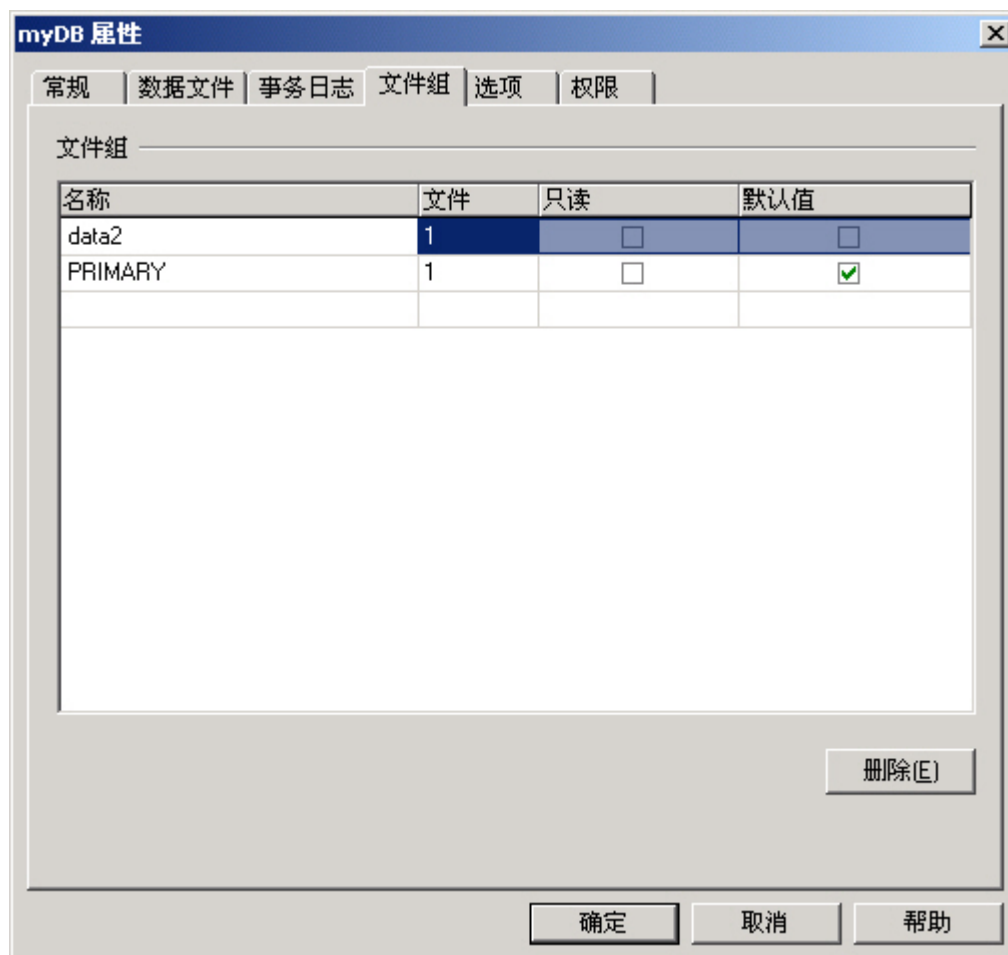
显示数据库 myDB 的数据文件属性页



显示数据库 myDB 的事务日志属性页



显示数据库 myDB 的文件组属性页



说明：在数据库的属性页中可以修改该数据库的逻辑文件名、物理文件名、文件大小等参数。

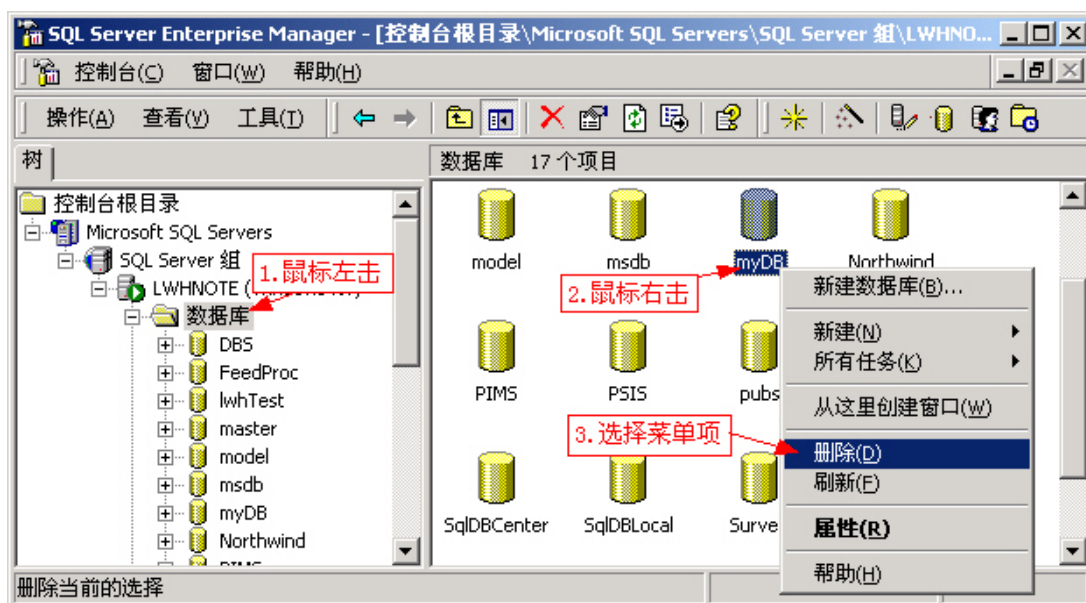
1.5.2 删除数据库

 帮助索引：<删除数据库|删除数据库>

一、用企业管理器删除数据库

如：删除数据库 myDB。

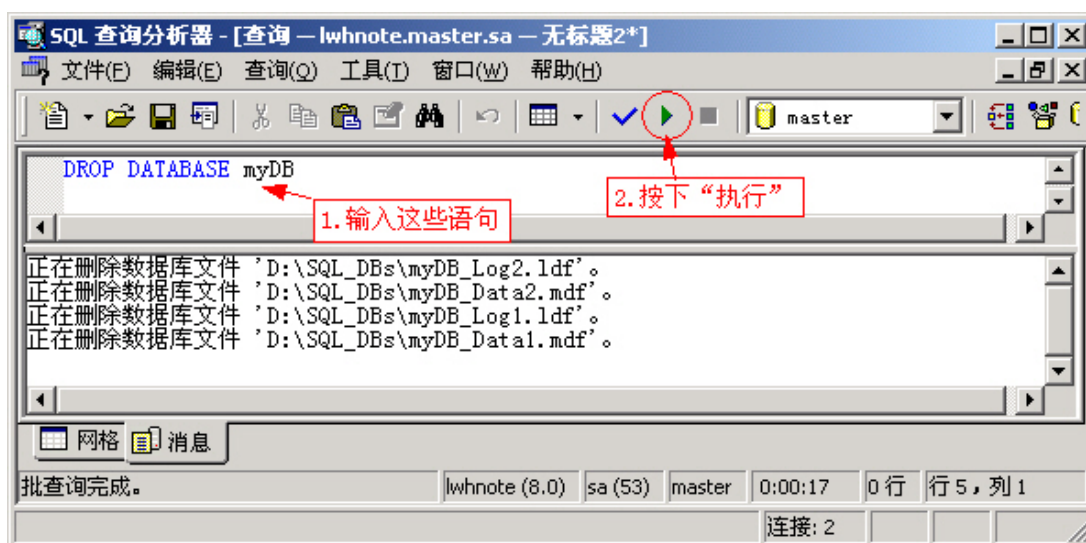
按如下方式操作即可。



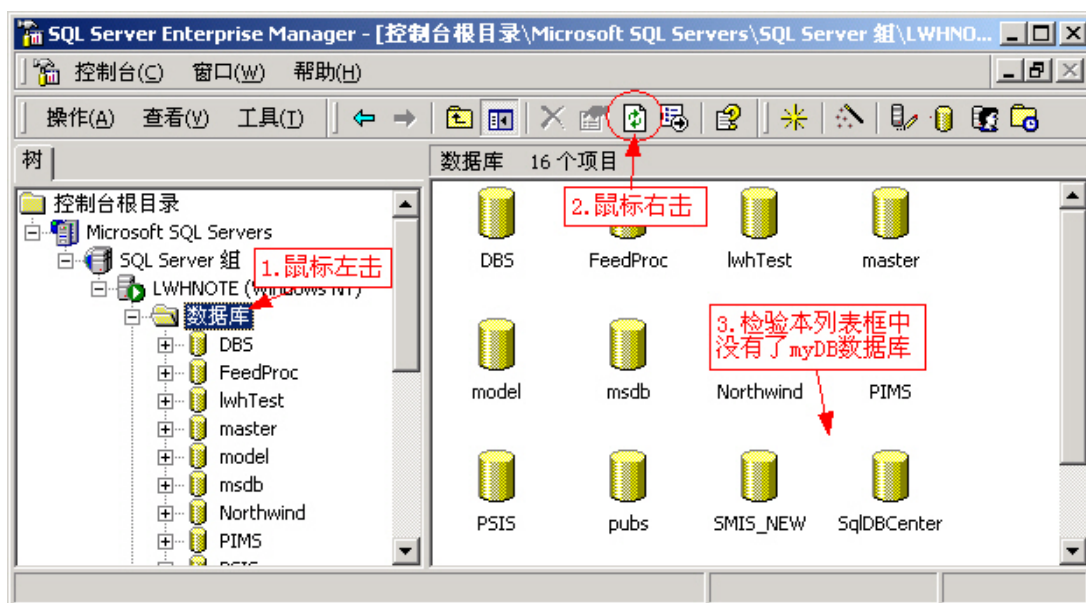
注：当数据库处于以下状态时不能被删除：数据库正在使用；数据库正在被恢复。

二、用查询分析器删除数据库

如：删除数据库 myDB。



在企业管理器中检验数据库 myDB 是否已经被删除。



1.5.3 数据库介绍

 帮助索引: <系统数据库>

SQL Server 2000 有四个系统数据库(System Database): Master、Model、Msdb、Tempdb 和两个实例数据库(Sample Database): pubs 和 Northwind。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。

一、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:

- 所有用户的登录信息;
- 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。

二、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, 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, 并将经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;
- 数据库的选项设置;
- 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。

三、Msdb 数据库

SQL Server、SQL Server 企业管理器、SQL Server Agent 使用 Msdb 数据库来存储计划信息以及备份和恢复相关的信息。

四、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 数据库中所做操作不会被记录, 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。

注: 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。

五、pubs 和 Northwind 数据库

pubs 和 Northwind 数据库可以作为 SQL Server 的学习工具, 很多 SQL Server 的书籍都是以此作为例子来讲解的。其中 pubs 数据为存储了一个虚构的图书出版公司的基本情况。Northwind 数据库包含了一个公司的销售数据。此公司名为 Northwind 商人, 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, 可以用查询分析器运行“安装目录\MSSQL\Install”中的 Instpubs.sql 或

Instnwnd.sql 程序来重建 pubs 或 Northwind 数据库。

1.6 用户和安全性管理

对任何企业组织来说数据的安全性最为重要。安全性主要是指允许那些具有相应的数据访问权限的用户能够登录到 SQL Server 并访问数据以及对数据库对象实施各种权限范围内的操作, 但是要拒绝所有的非授权用户的非法操作。因此安全性管理与用户管理是密不可分的。SQL Server 2000 提供了内置的安全性和数据保护, 并且这种管理有效而又容易。

SQL Server 2000 的安全性管理是建立在认证 (authentication) 和访问许可

(permission) 两者机制上的。认证是指来确定登录 SQL Server 的用户的登录账号和

密码是否正确，以此来验证其是否具有连接 SQL Server 的权限。但是通过认证阶段并不代表能够访问 SQL Server 中的数据，用户只有在获取访问数据库的权限之后，才能够对服务器上的数据库进行权限许可下的各种操作（主要是针对数据库对象，如表、视图、存储过程等）这种用户访问数据库权限的设置是通过用户账号来实现的，同时在 SQL Server 中，角色作为用户组的代替物大大地简化了安全性管理。

1.6.1 SQL Server 的登录认证

 帮助索引：<登录到 SQL Server>

一、SQL Server 的登录认证简介

MS SQL Server 能在两种安全模式下运行：

- WINDOWS 认证模式
- 混合模式

A. WINDOWS 认证模式

SQL Server 数据库系统通常运行在 NT 服务器平台或基于 NT 构架的 WINDOWS2000 上，而 NT 作为网络操作系统，本身就具备管理登录、验证用户合法性的能力，所以 WINDOWS 认证模式正是利用这一用户安全性和账号管理的机制，允许 SQL Server 也可以使用 NT 的用户名和口令。在该模式下，用户只要通过 WINDOWS 的认证就可连接到 SQL Server，而 SQL Server 本身也没有必要管理一套登录数据。

WINDOWS 认证模式比起 SQL Server 认证模式来有许多优点，原因在于 WINDOWS 认证模式集成了 NT 或 WINDOWS 2000 的安全系统，并且 NT 安全管理具有众多特征，如安全合法性，口令加密，对密码最小长度进行限制等。所以当用户试图登录到 SQL Server 时，它从 NT 或 WINDOWS 2000 的网络安全属性中获取登录用户的账号与密码，并使用 NT 或 WINDOWS 2000 验证账号和密码的机制来检验登录的合法性，从而提高了 SQL Server 的安全性。

在 WINDOWS NT 中使用了用户组，所以当使用 WINDOWS 认证时，我们总是把用户归入一定的 NT 用户组，以便当在 SQL Server 中对 NT 用户组进行数据库访问权限设置时，能够把这种权限设置传递给单一用户，而且当新增加一个登录用户时，也总把它归入某一 NT 用户组，这种方法可以使用户更为方便地加入到系统中，并消除了逐一为每一个用户进行数据库访问权限设置而带来的不必要的工作量

B. 混合认证模式

在混合认证模式下，WINDOWS 认证和 SQL Server 认证这两种认证模式都是可用的。NT 的用户既可以使用 NT 认证，也可以使用 SQL Server 认证。前面已经介绍了 WINDOWS 认证的含义，下面介绍 SQL Server 认证模式。

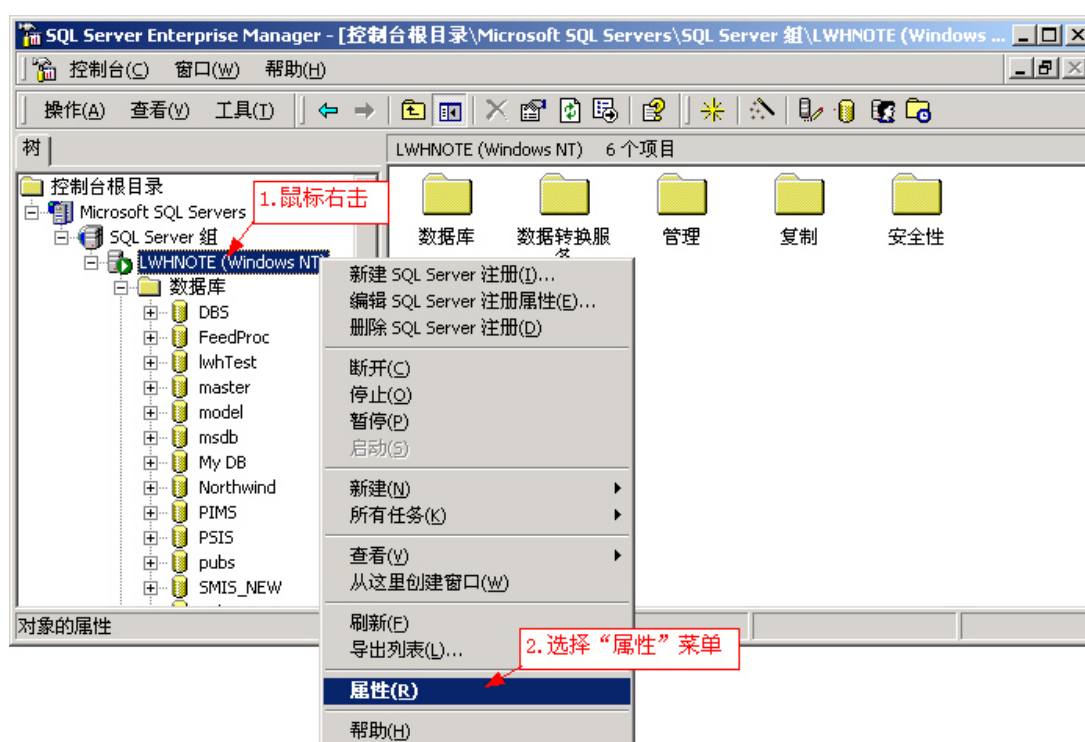
C. SQL Server 认证

在该认证模式下，用户在连接 SQL Server 时必须提供登录名和登录密码，这些登录信息存储在系统表 syslogins 中，与 NT 的登录账号无关。SQL Server 自己执行认证处理，如果输入的登录信息与系统表 syslogins 中的某条记录相匹配则表明登录成功。

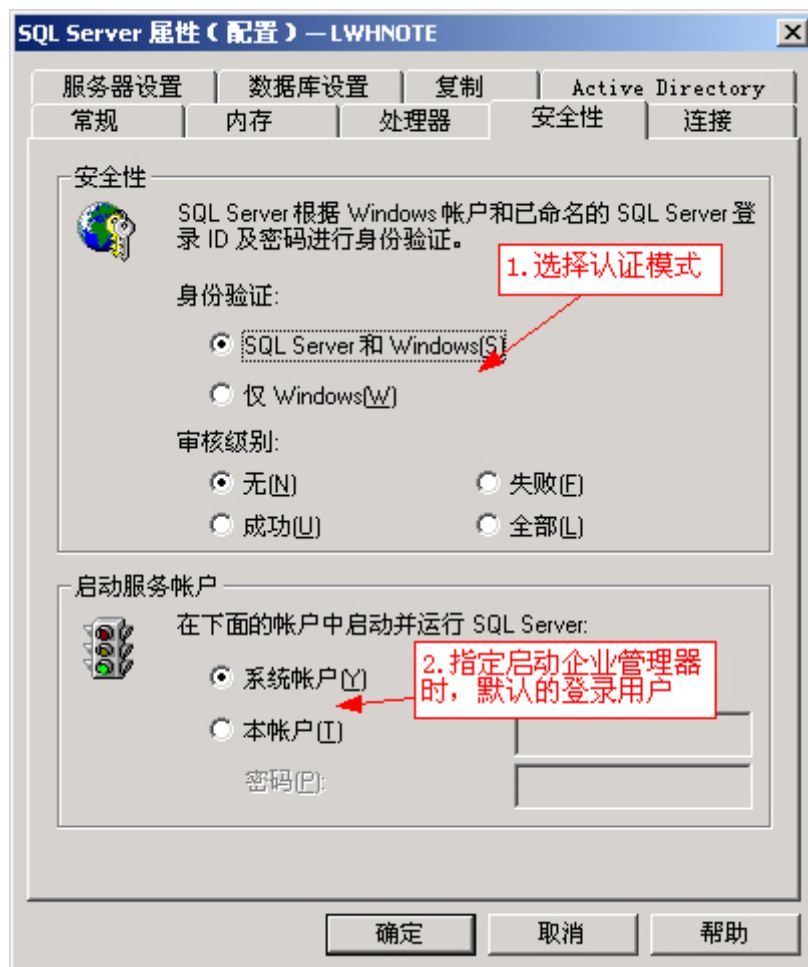
注：该方式常用于系统开发中，因为客户机常常与服务不是同一台计算机，必须使用该方式进行登录。

二、SQL Server 认证模式的设置

在对登录进行增加、删除等操作前，必须首先设置 SQL Server 的认证模式。通过企业管理器来进行认证模式的设置 3002



弹出“SQL Server 属性配置”对话框。



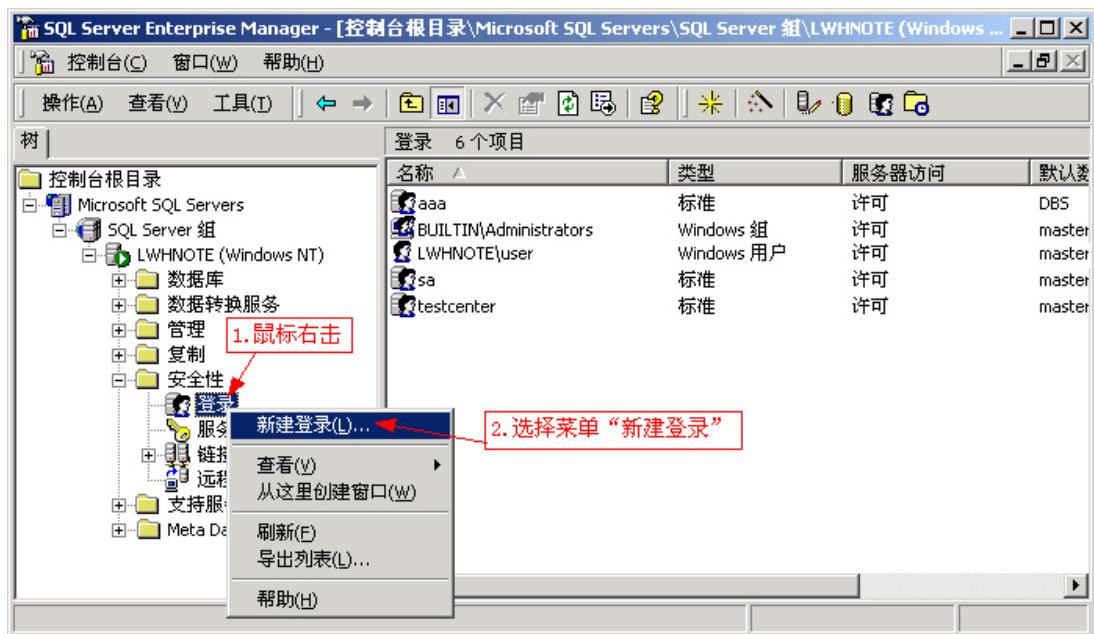
1.6.2 管理登录用户

 帮助索引: <登录到 SQL Server>

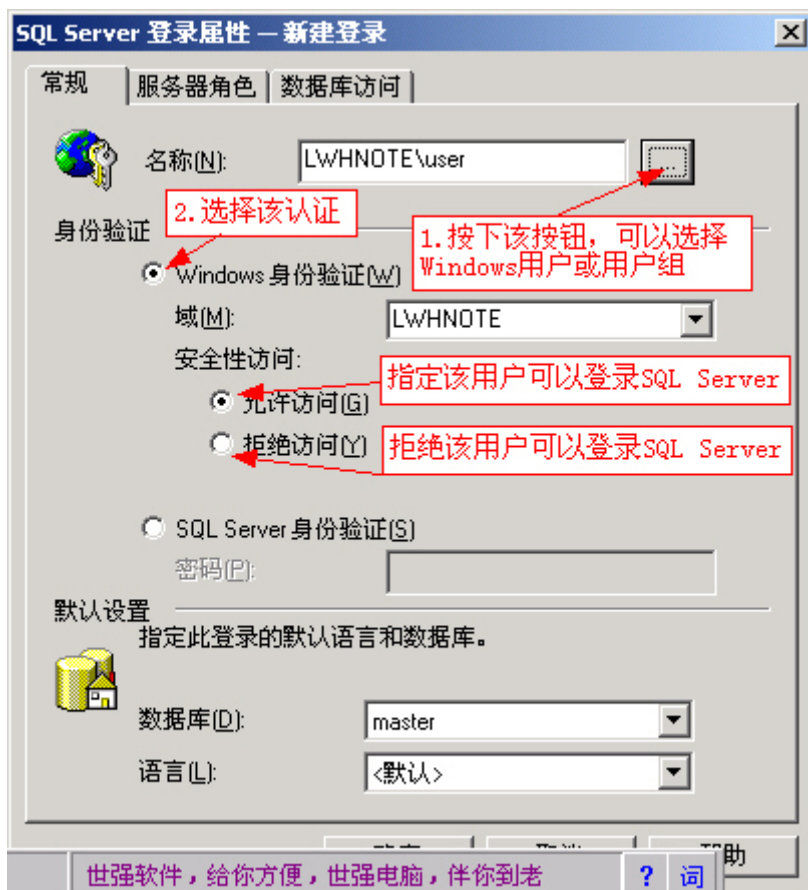
一、用企业管理器管理 SQL Server 登录

A. 增加一个 Windows NT 用户或用户组

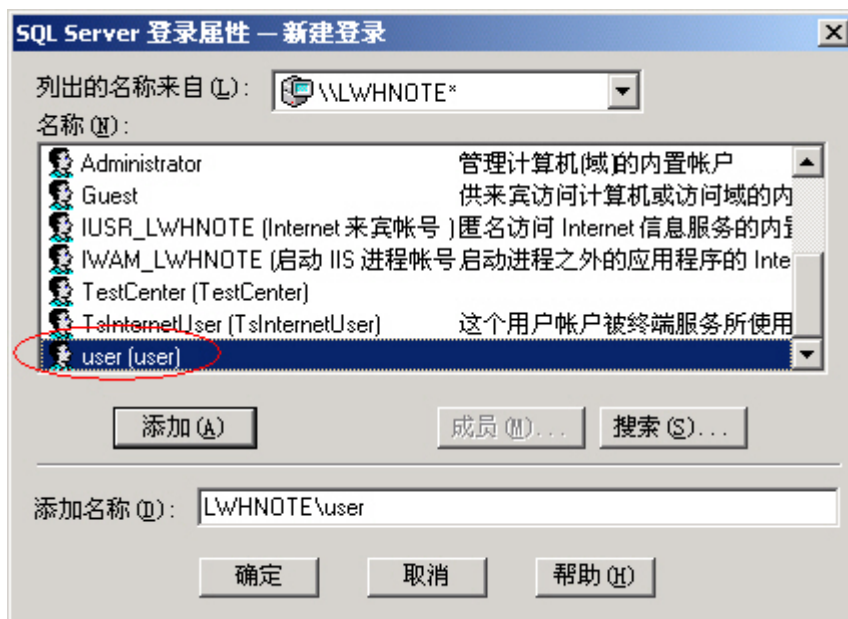
如: 增加一个 Windows NT 用户为 “LWHNOTE\User”。



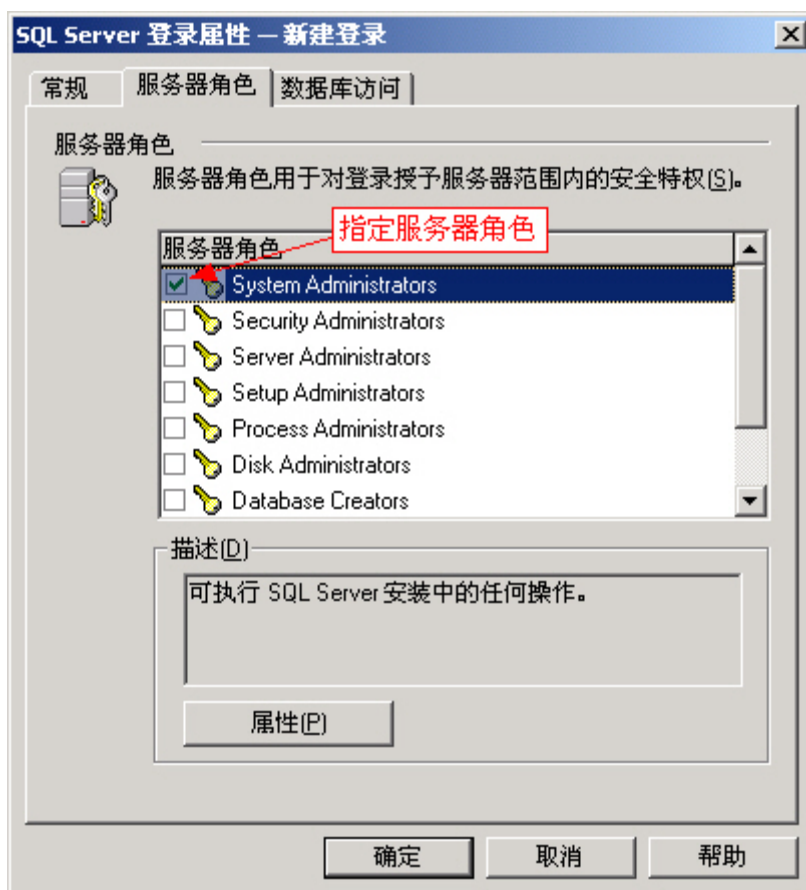
设置登录用户的常规属性



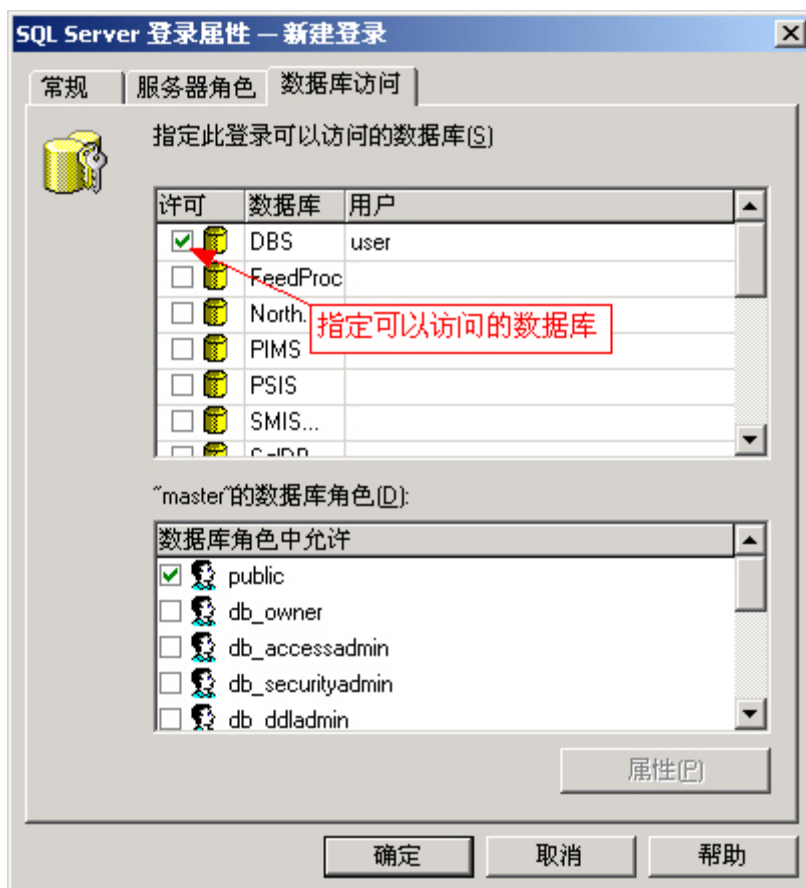
选择 Windows NT 用户或用户组



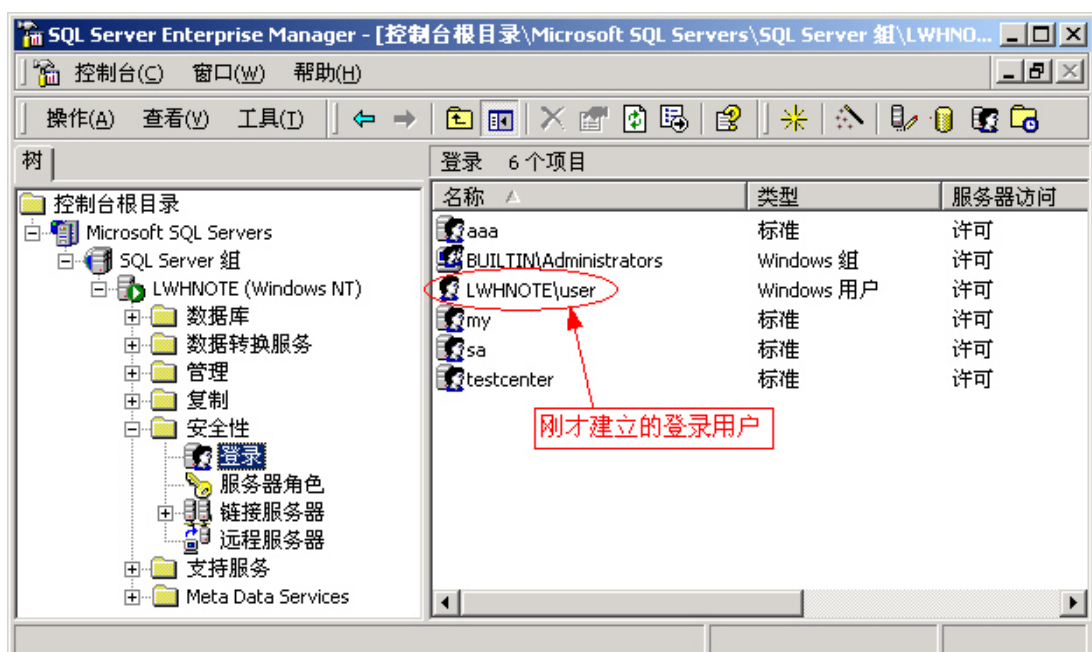
设置登录用户的服务器角色为系统管理员



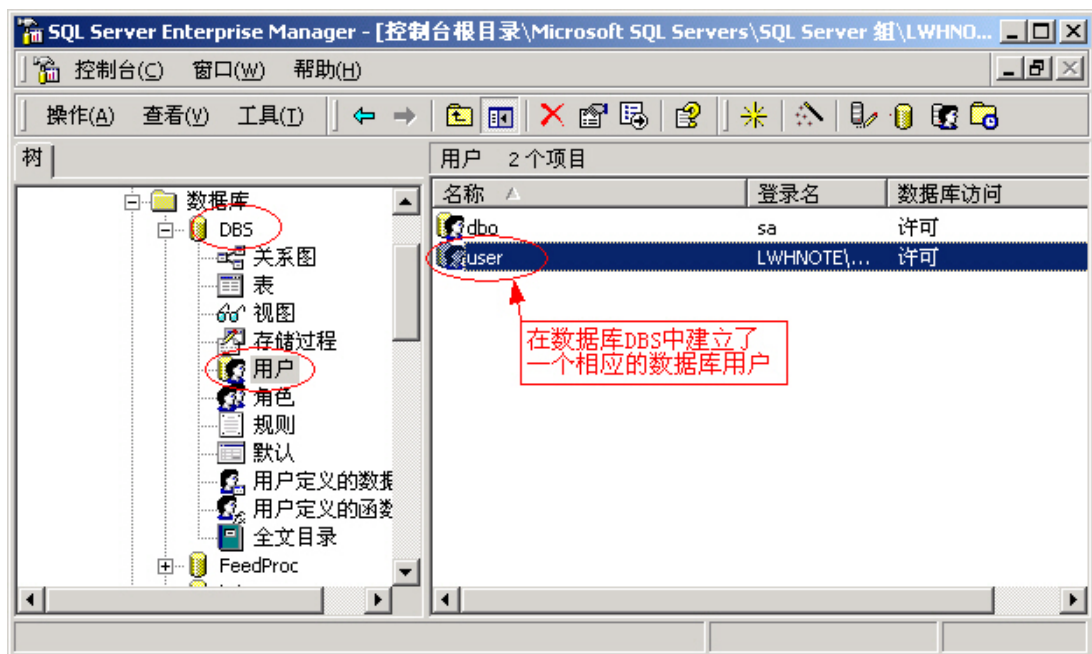
设置登录用户可访问的数据库为DBS。一个登录用户可以访问很多个数据库，可以在这里指定，指定后在相应的数据库中建立了相应的[数据库用户](#)。



按下“确定”，即可建立一个 Windows NT 用户，然后，可以在企业管理器中检验登录用户是否已经建立。

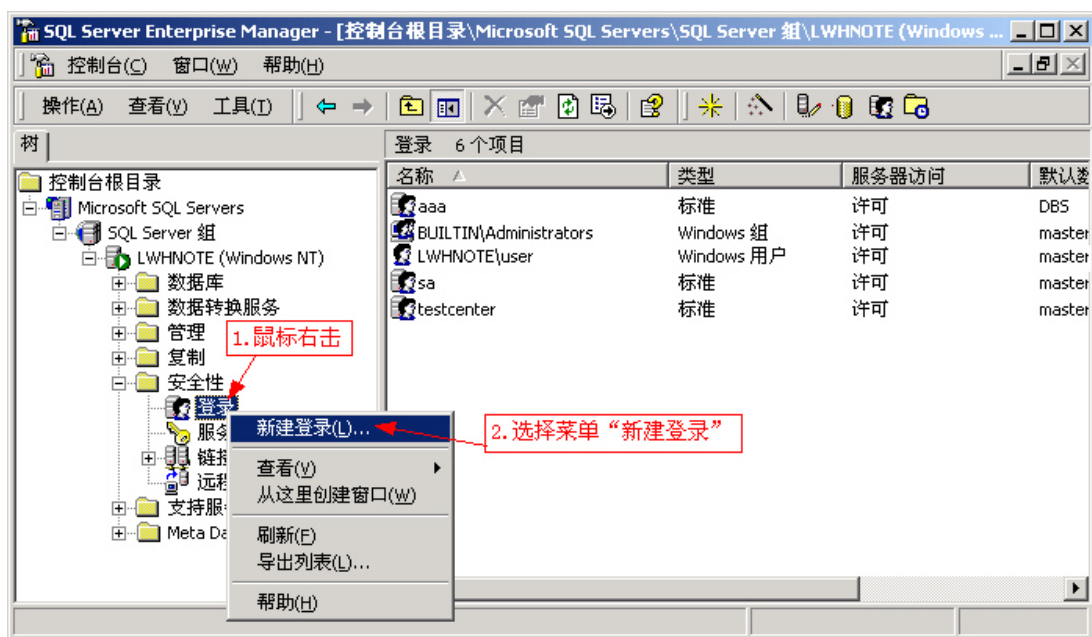


可以在企业管理器中检验数据库 DBS 是否已经建立一个数据库用户 user。

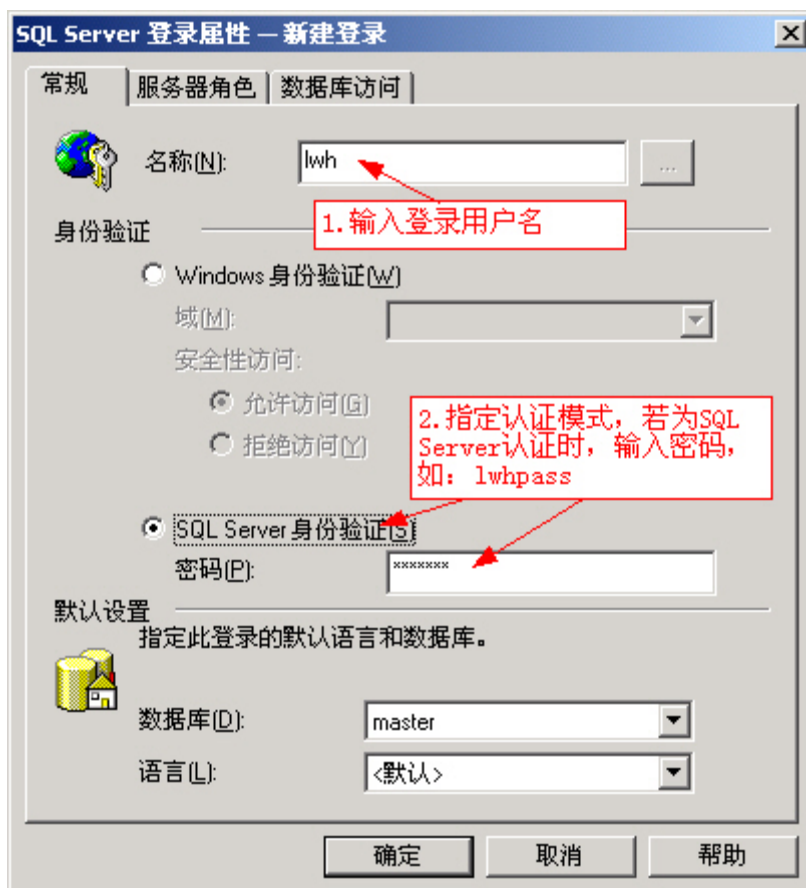


B. 增加一个 SQL Server 用户

如：增加一个 SQL Server 用户，用户名为“lwh”，密码为“lwhpass”。



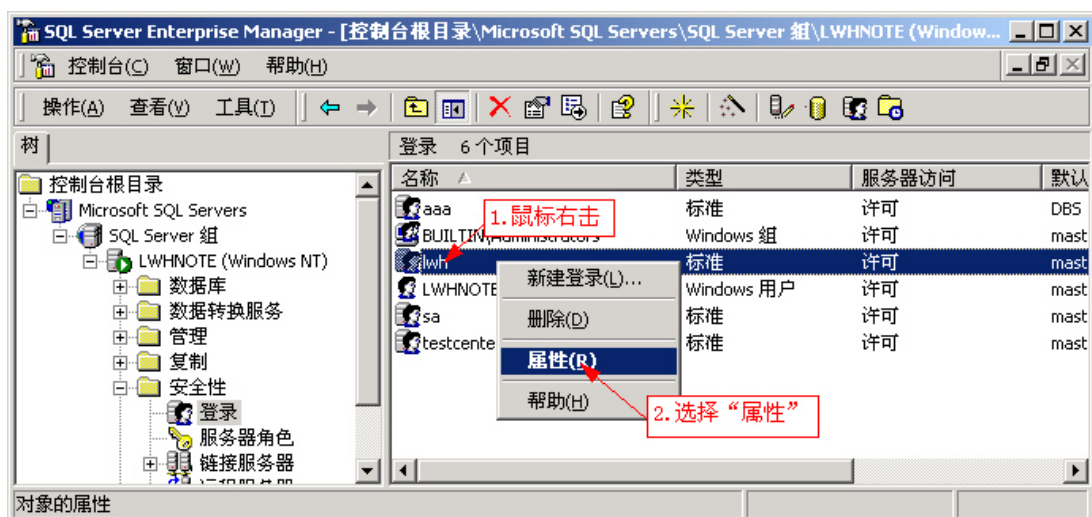
设置登录用户的常规属性

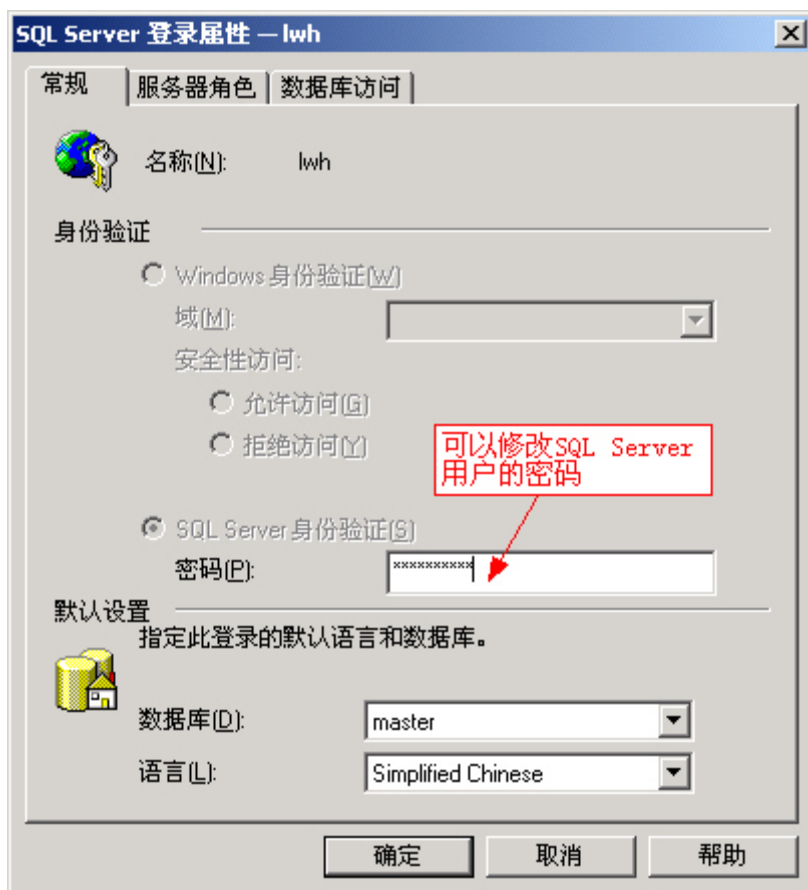


参见<增加一个 Windows NT 用户或用户组>, 设置该用户的“服务器角色”和“数据库访问”属性。

C. 修改用户

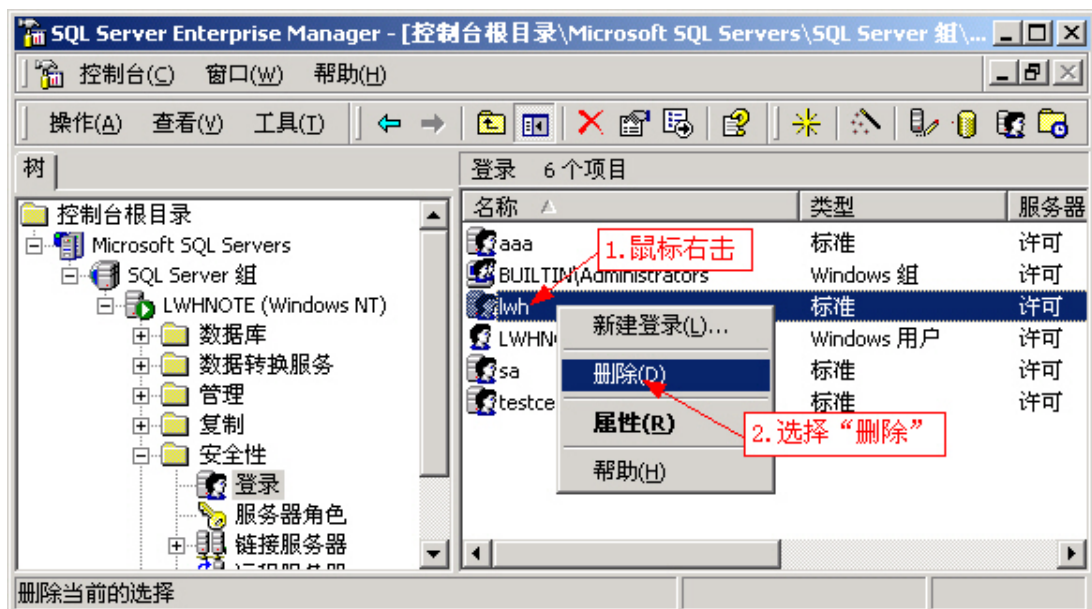
如: 修改 SQL Server 用户 lwh 的属性。





D. 删除用户

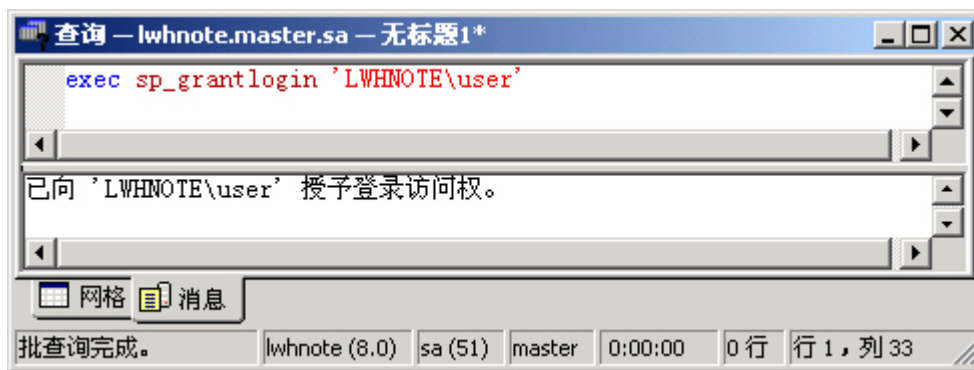
如：删除 SQL Server 用户 lwh。



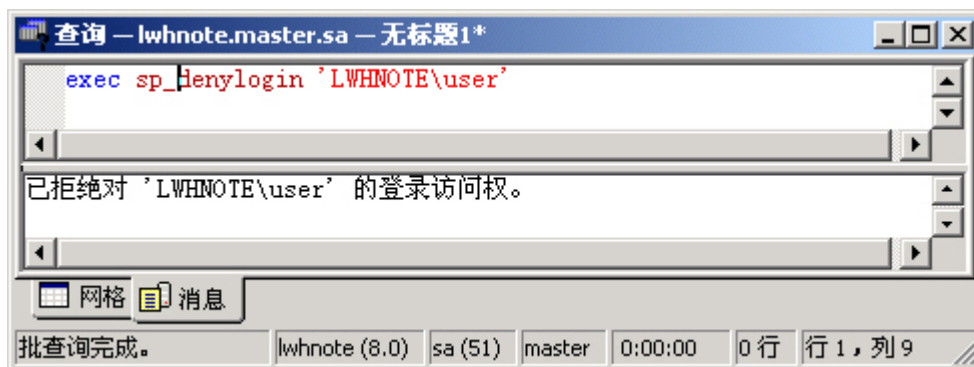
二、用查询分析器管理 QL Server 登录

A. 增加一个 Windows NT 用户或用户组

如：增加一个 Windows NT 用户为“LWHNOTE\User”，授予登录访问权。

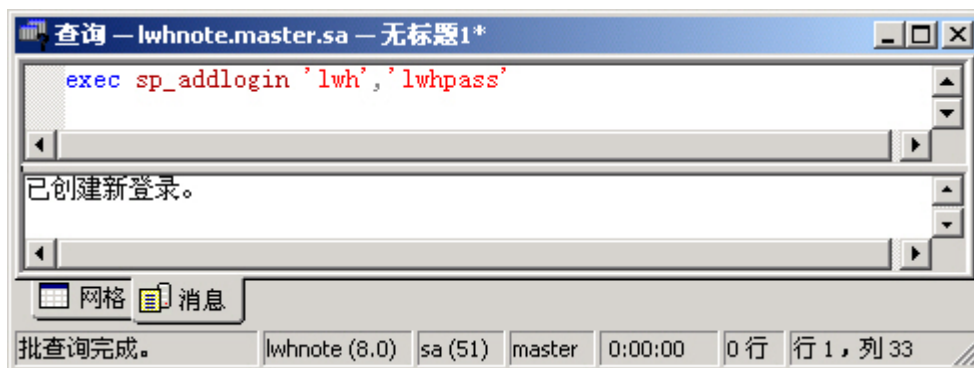


取消 Windows NT 用户“LWHNOTE\User”的登录访问权。



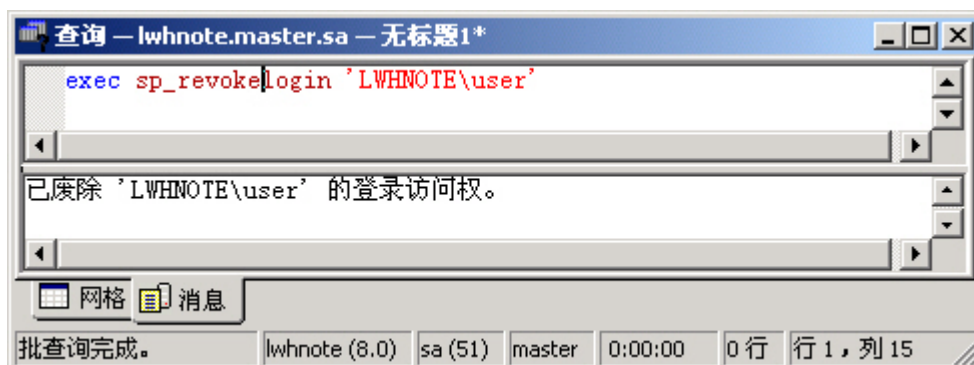
B. 增加一个 SQL Server 用户

如：增加一个 SQL Server 用户，用户名为“lwh”，密码为“lwhpass”。

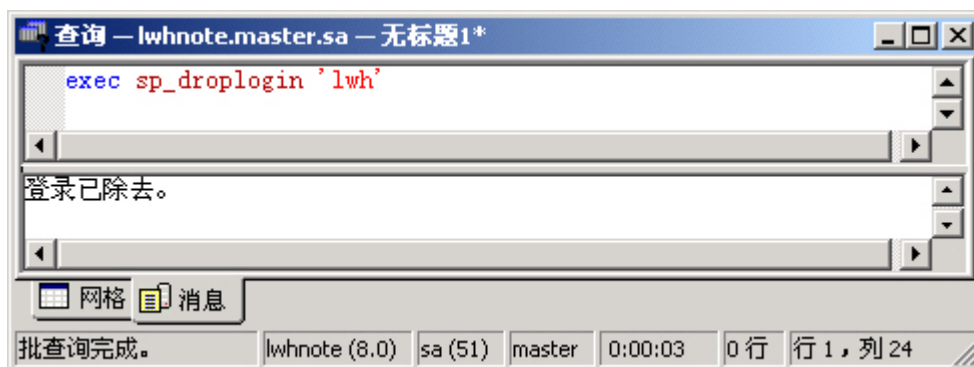


C. 删除用户

如：删除 Windows NT 用户“LWHNOTE\User”。



如：删除 SQL Server 用户 lwh。



1.6.3 管理数据库用户

 帮助索引：<登录到 SQL Server>

一、数据库用户简介

数据库用户用来指出哪一个人可以访问哪一个数据库。在一个数据库中用户 ID 唯一标识一个用户，用户对数据的访问权限以及对数据库对象的所有关系都是通过用户账号来控制的。用户账号总是基于数据库的，即两个不同数据库中可以有相同的用户账号。

在数据库中用户账号与登录账号是两个不同的概念。一个合法的登录账号只表明该账号通过了 NT 认证或 SQL Server 认证，但不能表明其可以对数据库数据和数据对象进行某种或某些操作，所以一个登录账号总是与一个或多个数据库用户账号（这些账号必须分别存在相异的数据库中）相对应，这样才可以访问数据库。例如，登录账号 sa 自动与每一个数据库用户 dbo 相关联。

注：由于登录帐号 sa 自动与数据库用户 dbo 相关联，一般情况下，不必另外创建数据库用户。

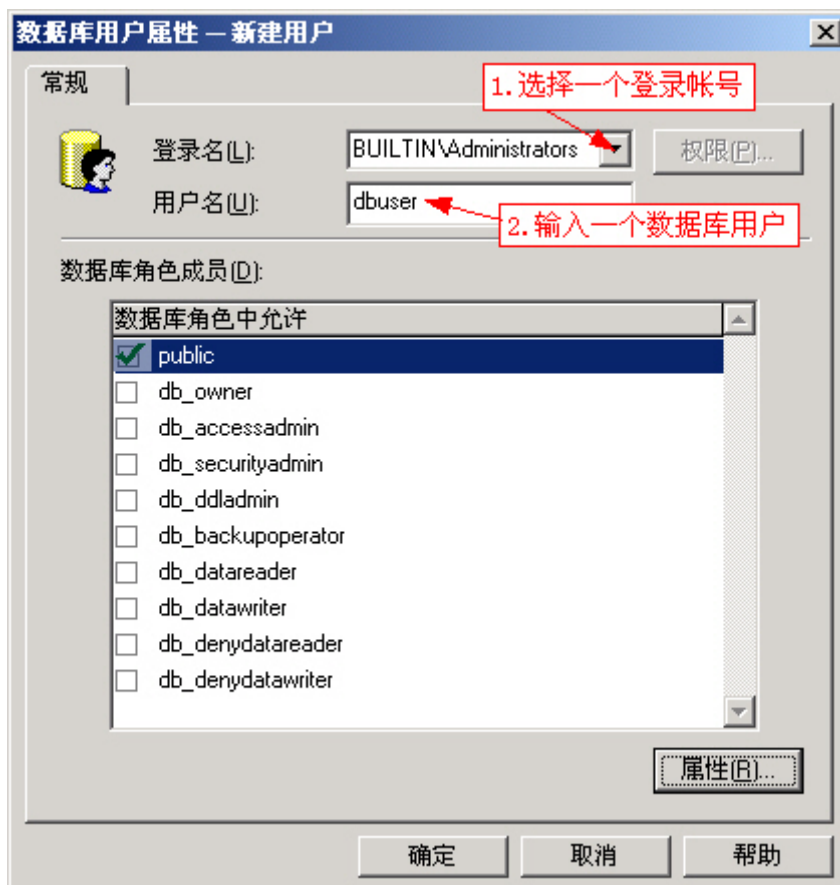
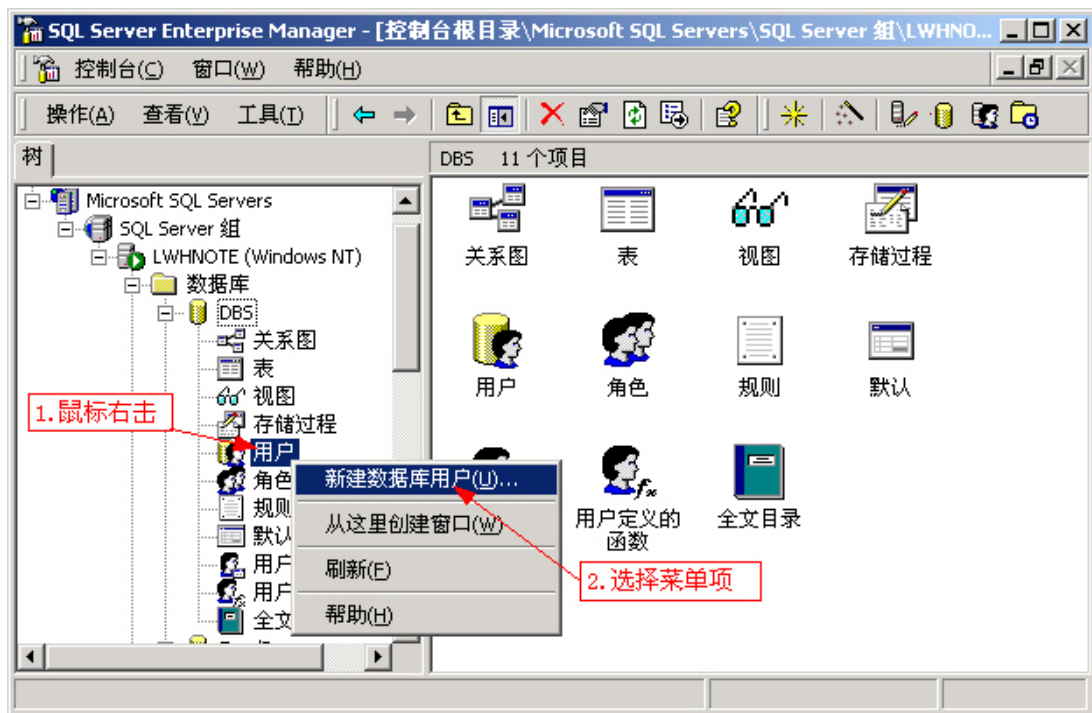
二、用企业管理器管理数据库用户

A. 通过建立一个登录用户增加一个数据库用户

<参见[1.6.2 管理登录用户](#)中“A. 增加一个Windows NT用户或用户组”>

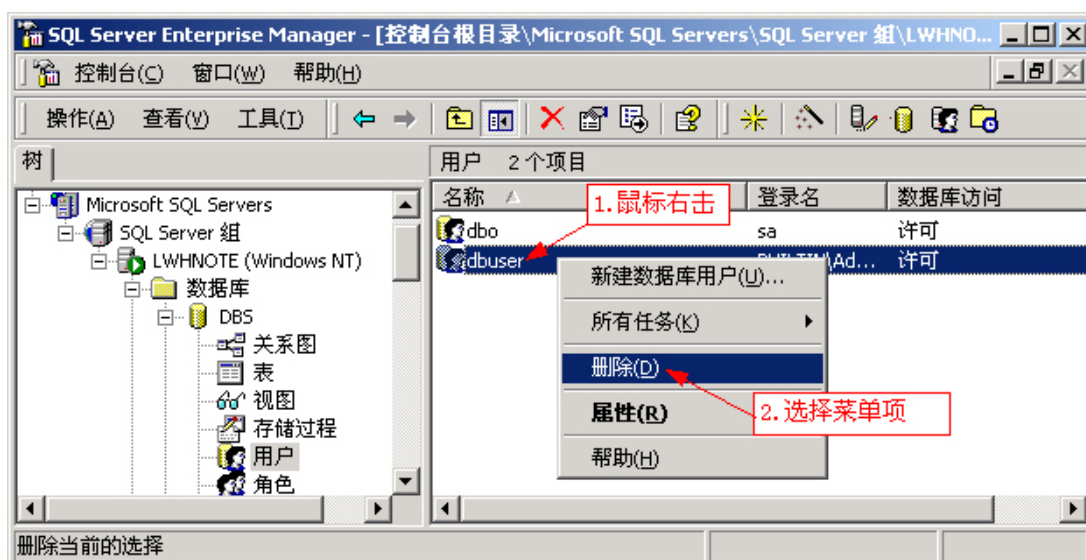
B. 增加一个数据库用户

如：给数据库 dbs 增加一个 dbuser 用户。



C. 删除一个数据库用户

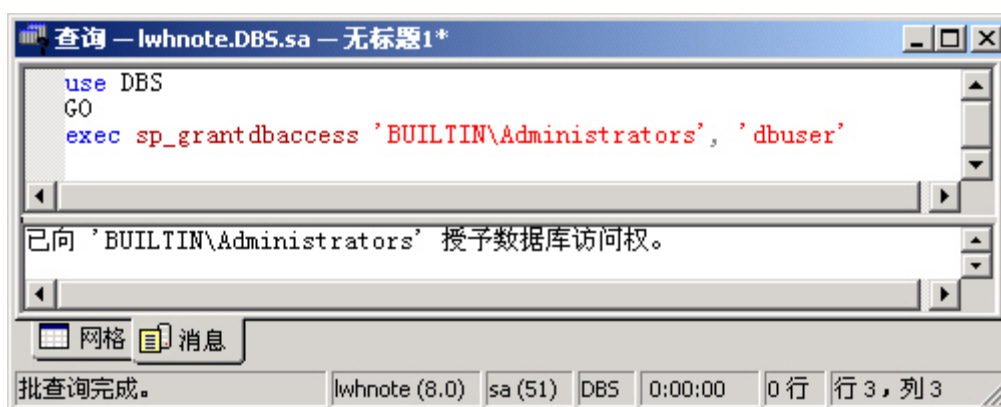
如：删除数据库 **db**s 中的数据库用户 **dbuser**。



三、用查询分析器管理数据库用户

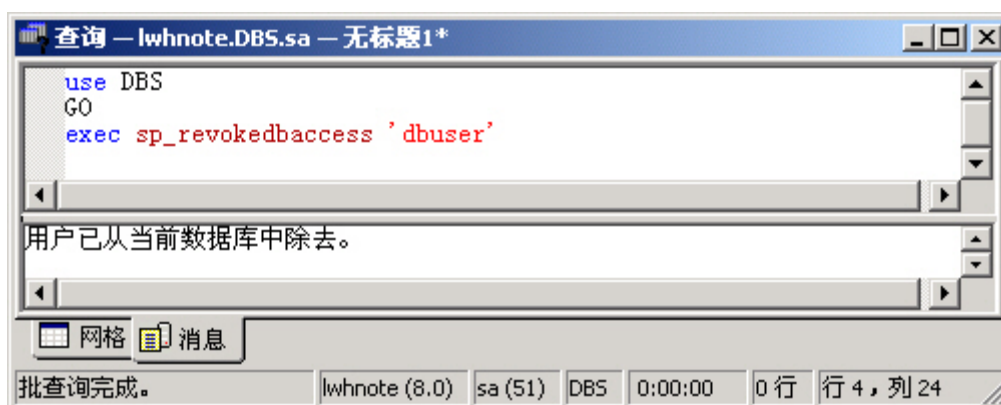
A. 增加一个数据库用户

如：给数据库 **db**s 增加一个 **dbuser** 用户。



B. 删除一个数据库用户

如：删除数据库 **db**s 中的数据库用户 **dbuser**。



1.6.4 权限管理

一、权限管理简介

用户在登录到 SQL Server 之后, 其安全账号(用户账号)所归属的 NT 组或角色所被授予的权限决定了该用户能够对哪些数据库对象执行哪种操作以及能够访问、修改哪些数据。在 SQL Server 中包括两种类型的权限, 即对象权限和语句权限。

A. 对象权限

对象权限总是针对表、视图、存储过程而言, 它决定了能对表、视图、存储过程执行哪些操作(如 UPDATE、DELETE、INSERT、EXECUTE)。如果用户想要对某一对象进行操作, 其必须具有相应的操作的权限。例如, 当用户要成功修改表中数据时, 则前提条件是他已经被授予表的 UPDATE 权限。

不同类型的对象支持不同的针对它的操作, 例如不能对表对象执行 EXECUTE 操作。下面列举了针对各种对象的可能操作。

对象	操 作
表	SELECT, INSERT, UPDATE, DELETE, REFERENCE
视图	SELECT, UPDATE, INSERT, DELETE
存储过程	EXECUTE
列	SELECT, UPDATE

B. 语句权限

语句权限主要指用户是否具有权限来执行某一语句, 这些语句通常是一些具有管理性的操作, 如创建数据库、表、存储过程等。这种语句虽然仍包含有操作(如 CREATE)的对象, 但这些对象在执行该语句之前并不存在于数据库中(如创建一个表, 在 CREATE TABLE 语句未成功执行前, 数据库中没有该表), 所以将其归为语句权限范畴。下面列举了所有语句权限清单。

语句	含 义
CREATE DATABASE	创建数据库
CREATE TABLE	创建表
CREATE VIEW	创建视图
CREATE RULE	创建规则
CREATE DEFAULT	创建缺省
CREATE PROCEDURE	创建存储过程
BACKUP DATABASE	备份数据库
BACKUP LOG	备份事务日志

在 SQL Server 中我们使用 GRANT、REVOKE 和 DENY 三种命令来管理权限。

- GRANT: 用来把权限授予某一用户, 以允许该用户执行针对该对象的操作(如 UPDATE、SELECT、DELETE、EXECUTE)或允许其运行某些语句(如 CREATE TABLE、CRETAE DATABASE);

- REVOKE: 取消用户对某一对象或语句的权限(这些权限是经过 GRANT 语句授予的),

不允许该用户执行针对数据库对象的某些操作（如 UPDATE、SELECT、DELETE、EXECUTE）或不允许其运行某些语句（如 CREATE TABLE、CREATE DATABASE）；

- **DENY**：用来禁止用户对某一对象或语句的权限，明确禁止其对某一用户对象执行某些操作（如 UPDATE、SELECT、DELETE、EXECUTE）或运行某些语句（如 CREATE TABLE、CREATE DATABASE）。

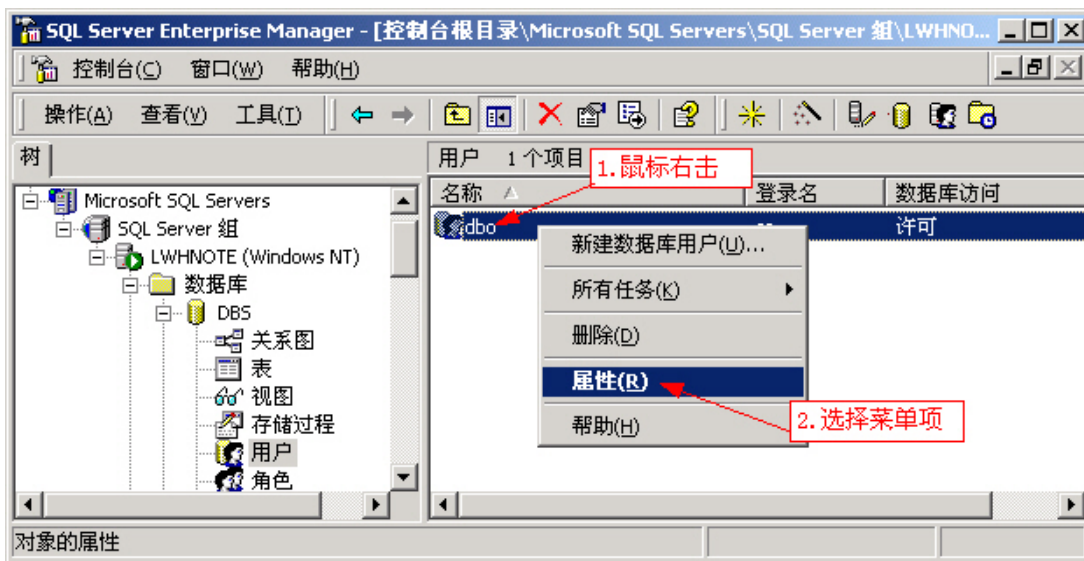
注：一般情况下使用系统分配给用户的缺省权限。

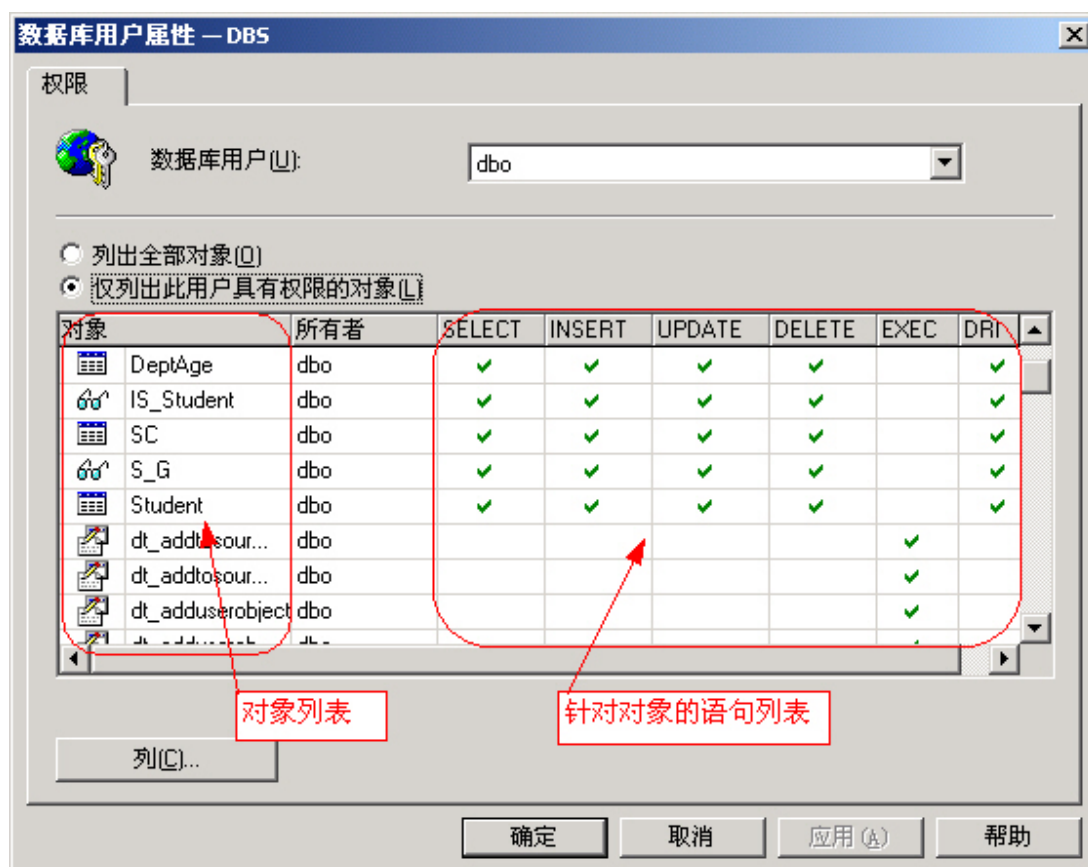
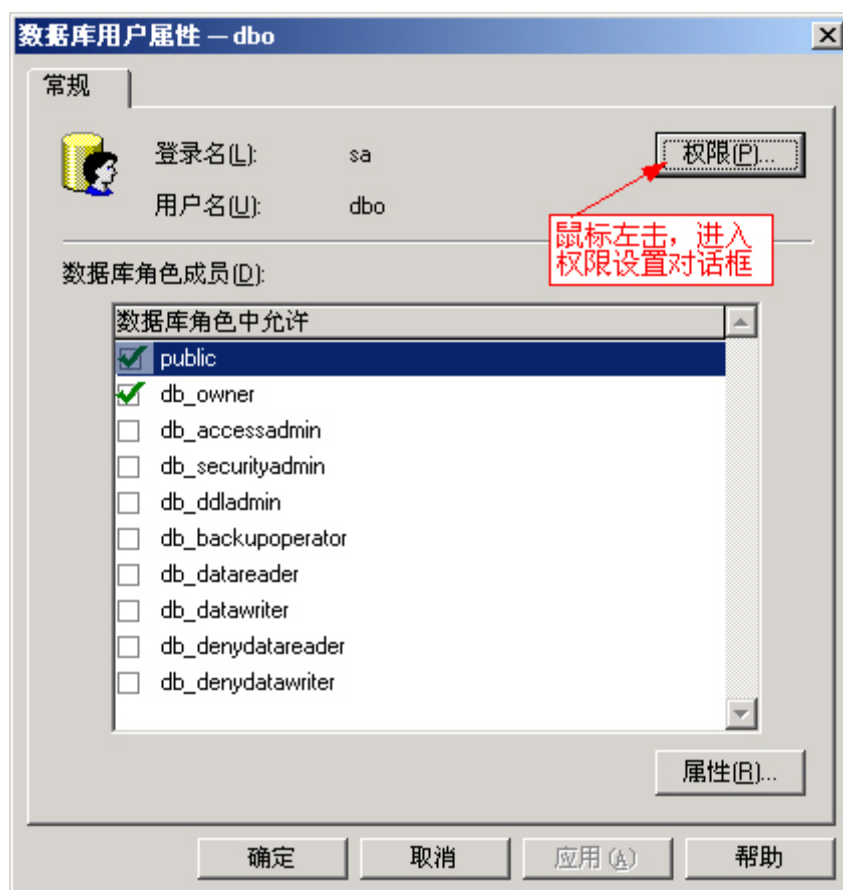
二、用企业管理器管理权限

在 SQL Server 中通过两种途径可实现对语句权限和对象权限的管理，从而实现对用户权限的设定。这两种途径分别为面向单一用户和面向数据库对象两种权限设置。

A. 面向单一用户的权限设置

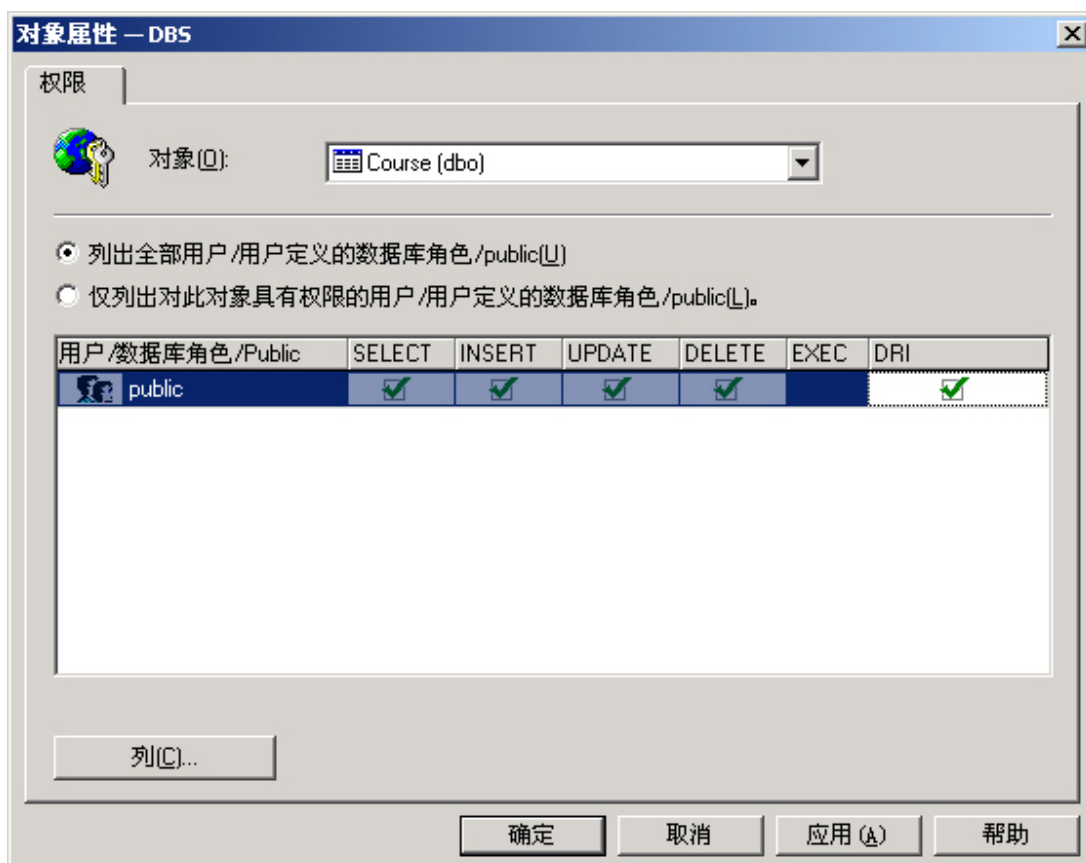
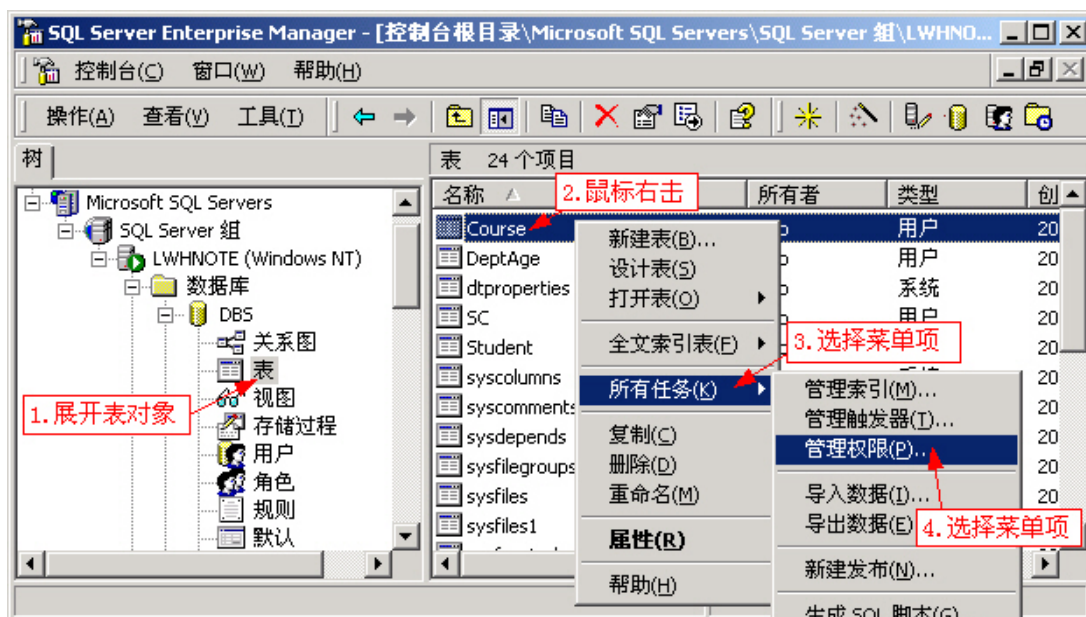
为某一用户设置其对当前数据库所有对象的访问权限。如：设置数据库 **db1** 的 **dbo** 用户的权限。





B. 面向数据库对象的权限设置

为某一数据库对象设置当前数据库所有用户对其的访问权限。如：设置数据库 **dbb** 中数据表 **Course** 的权限。



三、用查询分析器管理权限

(参见权限管理简介中内容)

1.6.5 角色管理

一、角色管理简介

SQL Server 管理者可以将某些用户设置为某一角色,这样只对角色进行权限设置便可实现对所有用户权限的设置,大大减少了管理员的工作量。在 SQL Server 中主要有两种角色类型:服务器角色与数据库角色。

A. 服务器角色

服务器角色是指根据 SQL Server 的管理任务以及这些任务,相对的重要性等级来把具有 SQL Server 管理职能的用户划分成不同的用户组,每一组所具有管理 SQL Server 的权限已被预定义。服务器角色适用在服务器范围内同,并且其权限不能被修改。例如,具有 sysadmin 角色的用户在 SQL Server 中可以执行任何管理性的工作,任何企图对其权限进行修改的操作都将会失败,这一点与数据库角色不同。

SQL Server 共有 7 种预定义的服务器角色,如下表。

服务器角色	描 述
sysadmin	可以在 SQL Server 中做任何事情
serveradmin	管理 SQL Server 服务器范围内的配置
setupadmin	增加、删除连接服务器,建立数据库复制,管理扩展存储过程
securityadmin	管理数据库登录
processadmin	管理 SQL Server 进程
dbcreator	创建数据库,并对数据库进行修改
diskadmin	管理磁盘文件

B. 数据库角色

在 SQL Server 中我们常会发现我们要将一套数据库专有权限授予给多个用户,但这些用户并不属于同一个 NT 用户组,或者虽然这些用户可以被 NT 管理者划为同一 NT 用户组,但遗憾的是,我们却没有管理 NT 账号的权限,这时我们就可以在数据库中添加新数据库角色或使用已经存在的数据库角色,并让这些有着相同数据库权限的用户归属于同一角色。

由此可见数据库角色能为某一用户或一组用户授予不同级别的管理或访问数据库或数据库对象的权限,这些权限是数据库专有的。而且还可以使一个用户具有属于同一数据库的多个角色。

SQL Server 提供了两种数据库角色类型:预定义的数据库角色、用户自定义的数据库角色。

下表列出了预定义数据库角色。

预定义数据库角色	含 义
db_owner	数据库的所有者,可以执行任何数据库管理工作
db_accessadmin	可以增加或删除登录用户
db_datareader	只能对数据库中任何表执行 SELECT 操作
db_datawriter	只能对数据库中任何表执行 INSERT、UPDATE、

	DELETE 操作
db_ddladmin	可以新建、修改、删除数据库中任何对象，但不能发出 GRANT、REVOKE 或 DENY 语句
db_securityadmin	可以发出 GRANT、REVOKE 或 DENY 语句
db_backupoperator	可以备份数据库
db_denydatareader	不能对数据库中任何表执行 SELECT 操作
db_denydatawriter	不能对数据库中任何表执行 INSERT、UPDATE、DELETE 操作

二、用企业管理器管理权限

（鉴于角色管理的复杂性，暂不做要求）

三、用查询分析器管理权限

（鉴于角色管理的复杂性，暂不做要求）

1.7 备份和恢复

对任何企业组织来说数据的安全性最为重要。安全性主要是指允许那些具有相应的数据访问权限的用户能够登录到 SQL Server 并访问数据以及对数据库对象实施各种权限范围内的操作，但是要拒绝所有的非授权用户的非法操作。因此安全管理与用户管理是密不可分的。SQL Server 2000 提供了内置的安全性和数据保护，并且这种管理有效而又容易。

SQL Server 2000 的安全性管理是建立在认证（authentication）和访问许可（permission）两者机制上的。认证是指来确定登录 SQL Server 的用户的登录账号和密码是否正确，以此来验证其是否具有连接 SQL Server 的权限。但是通过认证阶段并不代表能够访问 SQL Server 中的数据，用户只有在获取访问数据库的权限之后，才能够对服务器上的数据库进行权限许可下的各种操作（主要是针对数据库对象，如表、视图、存储过程等）这种用户访问数据库权限的设置是通过用户账号来实现的，同时在 SQL Server 中，角色作为用户组的代替物大大地简化了安全管理。

1.7.1 备份和恢复概述

 帮助索引：<备份>

一、数据库备份的类型

在 SQL Server 2000 中有四种备份类型，分别为：

- 数据库备份(Database Backups)
- 事务日志备份(Transaction Log Backup)
- 差异备份(Differential Database Backups)
- 文件和文件组备份(File and File Group Backup)

A. 数据库备份(Database Backups)

数据库备份是指对数据库的完整备份，包括所有的数据以及数据库对象。实际上备份数据库过程就是首先将事务日志写到磁盘上，然后根据事务创建相同的数据库和数据库对象以及拷贝数据的过程。由于是对数据库的完全备份，所以这种备份类型不仅速度较慢，而且将占用大量磁盘空间。正因为如此，在进行数据库备份时，常将其安排在晚间，因为此时整个数据库系统几乎不进行其它事务操作，从而可以提高数据库备份的速度。

在对数据库进行完全备份时，所有未完成的事务或者发生在备份过程中的事务都不会被备份。如果您使用数据库备份类型，则从开始备份到开始恢复这段时间内发生的任何针对数据库的修改将无法恢复。所以我们总是在一定的要求或条件下才使用这种备份类型，比如：

- 数据不是非常重要，尽管在备份之后恢复之前数据被修改，但这种修改是可以忍受的；
- 通过批处理或其它方法，在数据库恢复之后可以很容易地重新实现在数据损坏前发生的修改；
- 数据库变化的频率不大。

在进行数据库备份时，如果您在备份完成之后又进行了事务日志备份，则在数据库备份过程中发生的事务将被备份；但若只进行数据库备份，常将数据库选项“trunc.log onchkpt”设置为 true，这样每次在运行到检查点(checkpoint)时，都会将事务日志截断。

B. 事务日志备份(Transaction Log Backup)

事务日志备份是指对数据库发生的事务进行备份，包括从上次进行事务日志备份、差异备份和数据库完全备份之后，所有已经完成的事务。在以下情况下我们常选择事务日志备份。

- 不允许在最近一次数据库备份之后发生数据丢失或损坏现象；
- 存储备份文件的磁盘空间很小或者留给进行备份操作的时间有限，例如兆字节级的数据库需要很大的磁盘空间和备份时间；
- 准备把数据库恢复到发生失败的前一点；
- 数据库变化较为频繁。

由于事务日志备份仅对数据库事务日志进行备份，所以其需要的磁盘空间和备份时间都比数据库备份（备份数据和事务）少得多，这是它的优点所在。正是基于此，我们在备份时常采用这样的策略，即每天进行一次数据库备份，而以一个或几个小时的频率备份事务日志。这样利用事务日志备份，我们就可以将数据库恢复到任意一个创建事务日志备份的时刻。

但是，创建事务日志备份却相对比较复杂。因为在使用事务日志对数据库进行恢复操作时，还必须有一个完整的数据库备份，而且事务日志备份恢复时必须按一定的顺序进行。比如在上周末对数据库进行了完整的数据库备份，在从周一到本周末的每一天都进行一次事务日志备份，那么若要打算对数据库进行恢复，则首先恢复数据库备份，然后按照顺序恢复从周一到本周末的事务日志备份。

有些时候数据库事务日志会被中断，例如数据库中执行了非日志操作（如创建索引、创建或删除数据库文件、自动或手工缩小数据库文件大小），此时应该立即创建数据库或差异备份，然后再进行事务日志备份。以前进行的事务日志备份也没有必要了。

C. 差异备份(Differential Database Backups)

差异备份是指将最近一次数据库备份以来发生的数据变化备份起来，因此差异备份实际上是一种增量数据库备份。与完整数据库备份相比，差异备份由于备份的数据量较小，所以

备份和恢复所用的时间较短。通过增加差异备份的备份次数，可以降低丢失数据的风险，将数据库恢复至进行最后一次差异备份的时刻，但是它无法像事务日志备份那样提供到失败点的无数据损失备份。

但在实际中为了最大限度地减少数据库恢复时间以及降低数据损失数量，我们常一起使用数据库备份、事务日志备份和差异备份，而采用的备份方案是这样的：

- 首先有规律地进行数据库备份，比如每晚进行备份；
- 其次以较小的时间间隔进行差异备份，比如三个小时或四个小时；
- 最后在相临的两次差异备份之间进行事务日志备份，可以每二十或三十分钟一次。

这样在进行恢复时，我们可先恢复最近一次的数据库备份，接着进行差异备份，最后进行事务日志备份的恢复。

但是，在更多的情况下我们希望数据库能恢复到数据库失败那一时刻，那么我们该怎样做呢？下面的方法也许会有大帮助。

- 首先如果能够访问数据库事务日志文件则应备份当前正处于活动状态的事务日志；
- 其次恢复最近一次数据库备份；
- 接着恢复最近一次差异备份；
- 最后按顺序恢复自差异备份以来进行的事务日志备份。

当然，如果无法备份当前数据库正在进行的事务，则只能把数据库恢复到最后一次事务日志备份的状态，而不是数据库失败点。

D. 文件和文件组备份 (File and File Group Backup)

文件或文件组备份是指对数据库文件或文件夹进行备份，但其不像完整的数据库备份那样同时也进行事务日志备份。使用该备份方法可提高数据库恢复的速度，因为其仅对遭到破坏的文件或文件组进行恢复。

但是在使用文件或文件组进行恢复时，仍要求有一个自上次备份以来的事务日志备份来保证数据库的一致性。所以在进行完文件或文件组备份后，应再进行事务日志备份。否则备份在文件或文件组备份中所有数据库变化将无效。

如果需要恢复的数据库部分涉及到多个文件或文件组，则应把这些文件或文件组都进行恢复。例如，如果在创建表或索引时，表或索引是跨多个文件或文件组，则在事务日志备份结束后应再对表或索引有关的文件或文件组进行备份，否则在文件或文件组恢复时将会出错。

二、备份和恢复的策略

通常而言我们总是依赖所要求的恢复能力如将数据库恢复到失败点备份文件的大小如完成数据库备份或只进行事务日志的备份或是差异数据库备份以及留给备份的时间等来决定该使用哪种类型的备份常用的备份选择方案有仅仅进行数据库备份或在进行数据库备份的同时进行事务日志备份或使用完整数据库备份和差异数据库备份。

选用怎样的备份方案将对备份和恢复产生直接影响而且也决定了数据库在遭到破坏前后的一致性水平所以在做出该决策时您必须认识到以下几个问题：

■ 如果只进行数据库备份那么将无法恢复自最近一次数据库备份以来数据库中所发生的所有事务这种方案的优点是简单而且在进行数据库恢复时操作也很方便；

■ 如果在进行数据库备份时也进行事务日志备份那么可以将数据库恢复到失败点那些在失败前未提交的事务将无法恢复但如果您在数据库失败后立即对当前处于活动状态的事务进行备份则未提交的事务也可以恢复。

从以上可以看出对数据库一致性的要求程度成为我们选择这样或那样的备份方案的主要的普遍性原因但在某些情况下对数据库备份提出更为严格的要求例如在处理比较重要业务的应用环境中常要求数据库服务器连续工作至多只留有一小段时间来执行系统维护任务在该情况下一旦出现系统失败则要求数据库在最短时间内立即恢复到正常状态以避免丢失过多的重

要数据由此可见备份或恢复所需时间往往也成为我们选择何种备份方案的重要影响因素。

那么如何才能减少备份和恢复所花费时间呢 SQL Server 提供了几种方法来减少备份或恢复操作的执行时间。

- 使用多个备份设备来同时进行备份处理同理可以从多个备份设备上同时进行数据库恢复操作处理;

- 综合使用完整数据库备份差异备份或事务日志备份来减少每次的需要备份的数据数量;

- 使用文件或文件组备份以及事务日志备份这样可以只备份或恢复那些包含相关数据的文件而不是整个数据库。

另外需要注意的是在备份时我们也要决定该使用哪种备份设备如磁盘或磁带并且决定如何在备份设备上创建备份比如将备份添加到备份设备上或将其覆盖。

在 SQL Server 2000 中有三种数据库恢复模式它们分别是简单恢复 SimpleRecovery 完全恢复 Full Recovery 批日志恢复 Bulk-logged Recovery。

A. 简单恢复(Simple Recovery)

所谓简单恢复就是指在进行数据库恢复时仅使用了数据库备份或差异备份，而不涉及事务日志备份。简单恢复模式可使数据库恢复到上一次备份的状态，但由于不使用事务日志备份来进行恢复，所以无法将数据库恢复到失败点状态。当选择简单恢复模式时常使用的备份策略是：首先进行数据库备份，然后进行差异备份。

B. 完全恢复(Full Recovery)

完全数据库恢复模式是指通过使用数据库备份和事务日志备份将数据库恢复到发生失败的时刻，因此几乎不造成任何数据丢失，这成为对付因存储介质损坏而数据丢失的最佳方法。为了保证数据库的这种恢复能力，所有的批数据操作比如 SELECT...INTO、创建索引都被写入日志文件。选择完全恢复模式时常使用的备份策略是：

- 首先进行完全数据库备份;

- 然后进行差异数据库备份;

- 最后进行事务日志的备份。

如果准备让数据库恢复到失败时刻必须对数据库失败前正处于运行状态的事务进行备份。

C. 批日志恢复(Bulk-logged Recovery)

批日志恢复在性能上要优于简单恢复和完全恢复模式，它能尽最大努力减少批操作所需要的存储空间。这些批操作主要：是 SELECT...INTO、批装载操作（如 bcp 操作或批插入操作）、创建索引、针对大文本或图像的操作（如 WRITETEXT、UPDATETEXT）。选择批日志恢复模式所采用的备份策略与完全恢复所采用的恢复策略基本相同。

从以上的论述中我们可以看到，在实际应用中，备份策略和恢复策略的选择不是相互孤立的，而是有着紧密的联系。我们并不仅仅是因为数据库备份为数据库恢复提供了“原材料”这一事实，以便在采用何种数据库恢复模式的决策中考虑该怎样进行数据库备份，更多是因为在选择该使用哪种备份类型时我们必须考虑到当使用该备份进行数据库恢复时，它能把遭到损坏的数据库“带”到怎样的状态（是数据库失败的时刻，还是最近一次备份的时刻）。但有一点我们必须强调，即备份类型的选择和恢复模式的确定都应服从于这一目标：尽最大可能，以最快速度减少或消灭数据丢失。

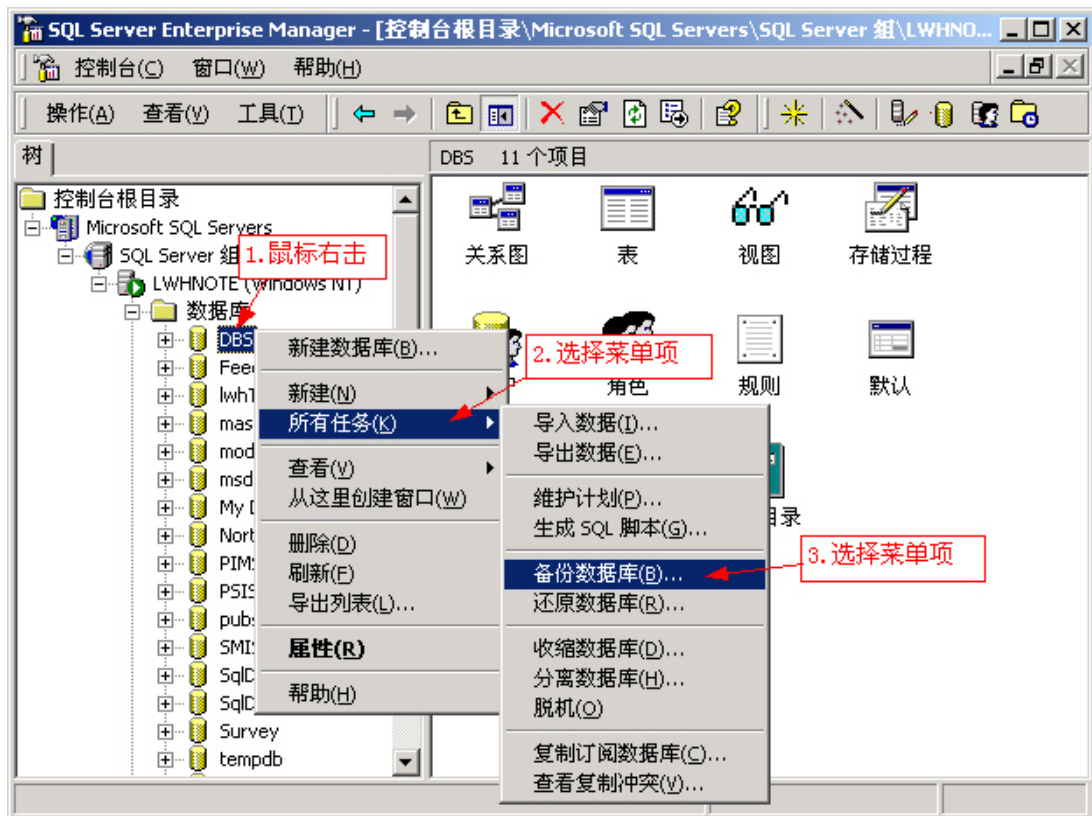
1.7.2 数据库备份

 帮助索引: <备份数据库>

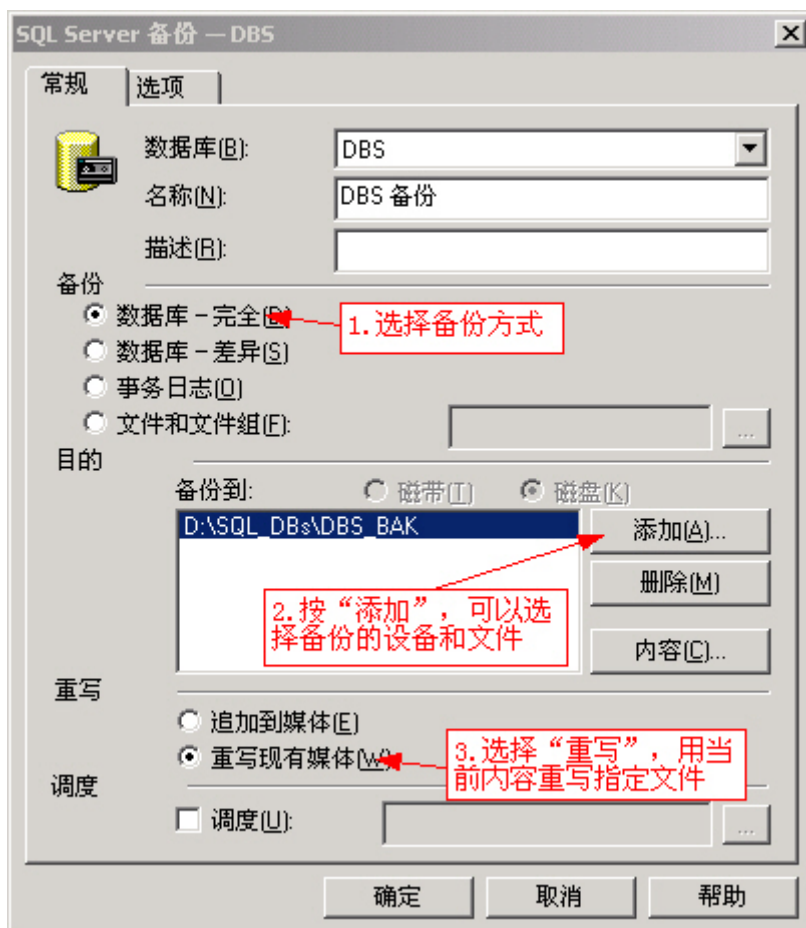
一、用企业管理器进行备份

在 SQL Server 中无论是数据库备份, 还是事务日志备份、差异备份、文件或文件组备份都执行相同的步骤。

如: 完全备份数据库 DBS 到一个磁盘文件 d:\SQL_DBS\DBS_BAK。

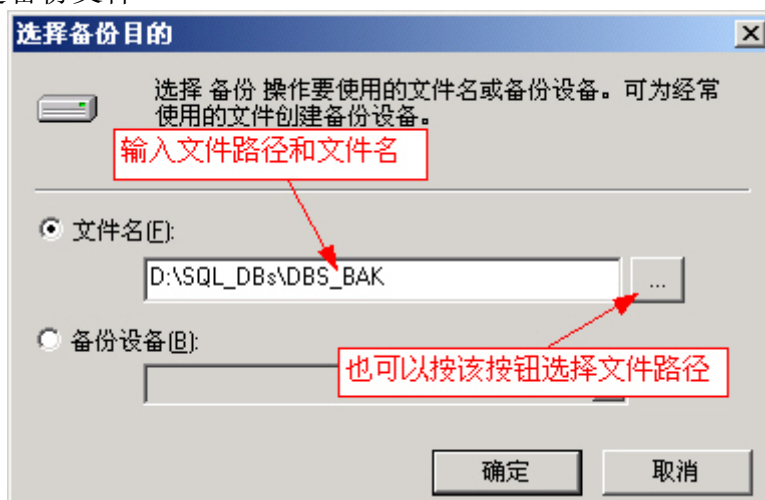


选择数据库备份方式, 然后按“添加”按钮, 指定备份文件。

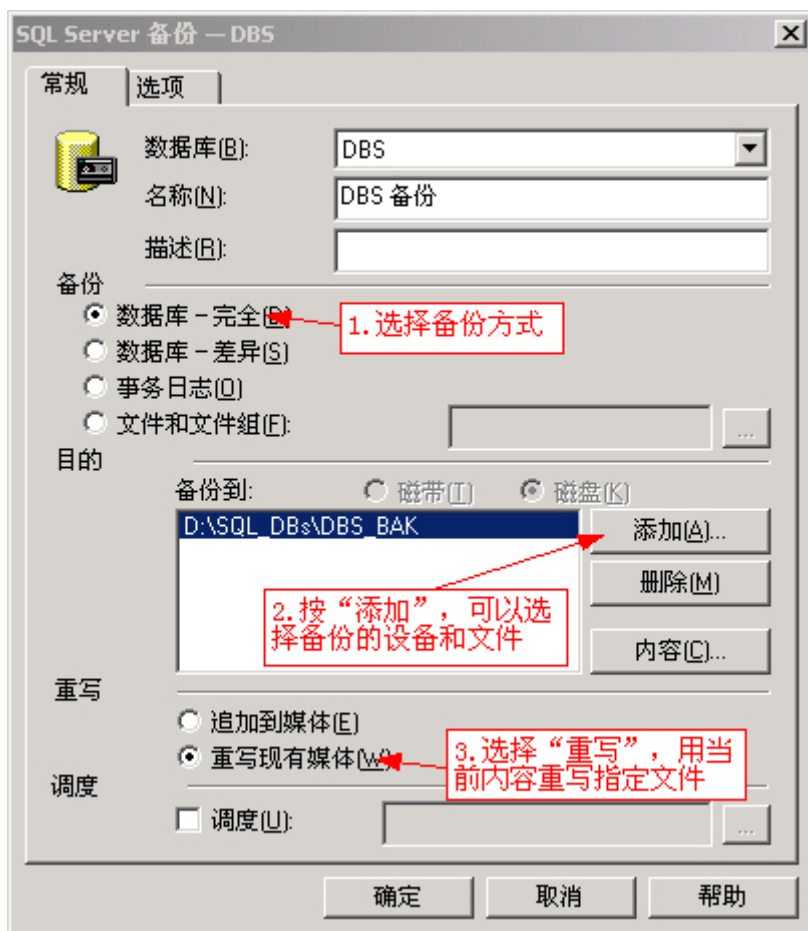


说明：选择不同的备份方式，可以实现完全数据库备份、差异数据库备份、事务日志备份等。

指定备份文件



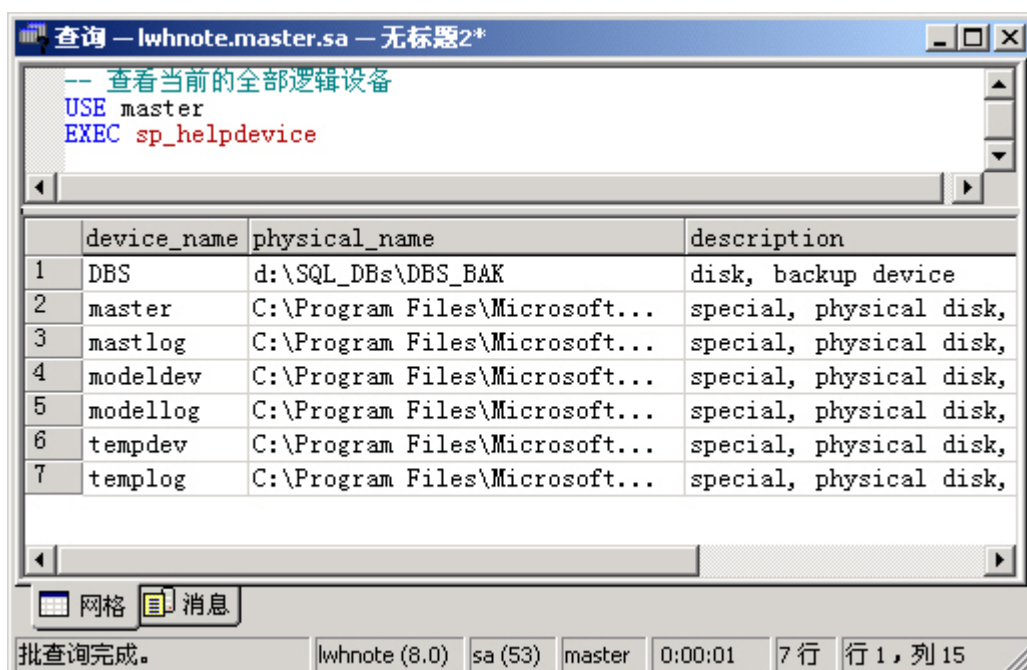
最后，按“确定”按钮，即可完成备份。



二、用查询分析器进行备份

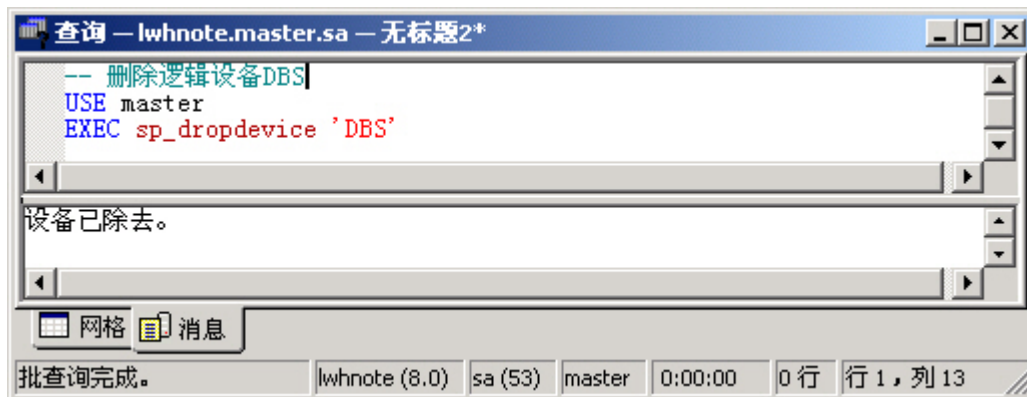
在备份前必须首先建立逻辑设备（可以理解为对应于磁盘上的一个文件），然后再进行备份。若指定的逻辑设备已经存在，可以删除逻辑设备，还可以查询当前的逻辑设备。

A. 查询当前的全部逻辑设备。



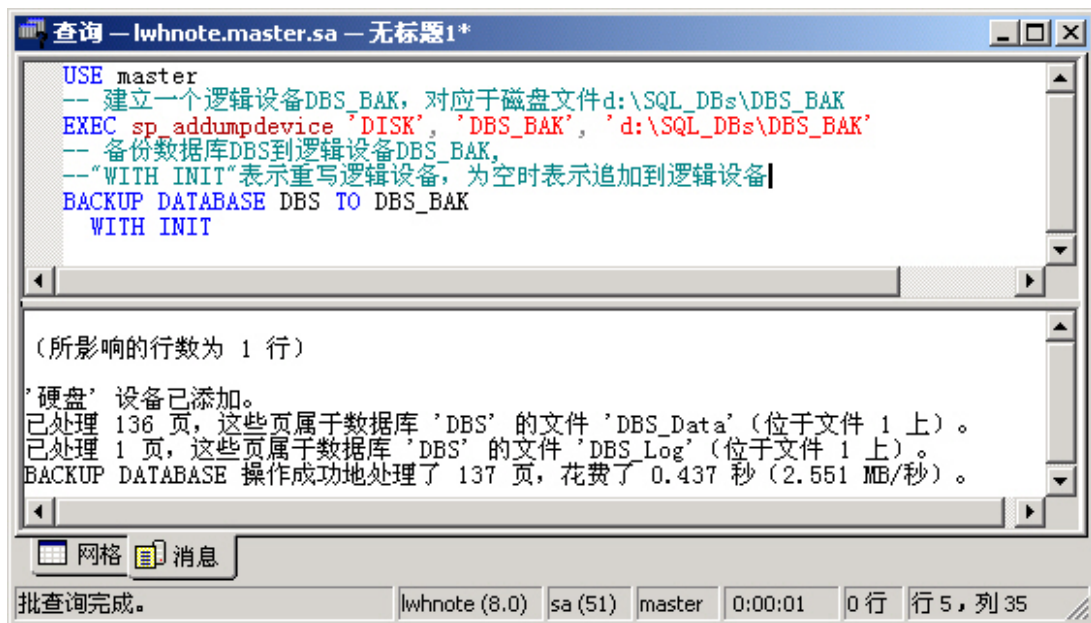
B. 删除逻辑设备

如：删除逻辑设备 DBS。



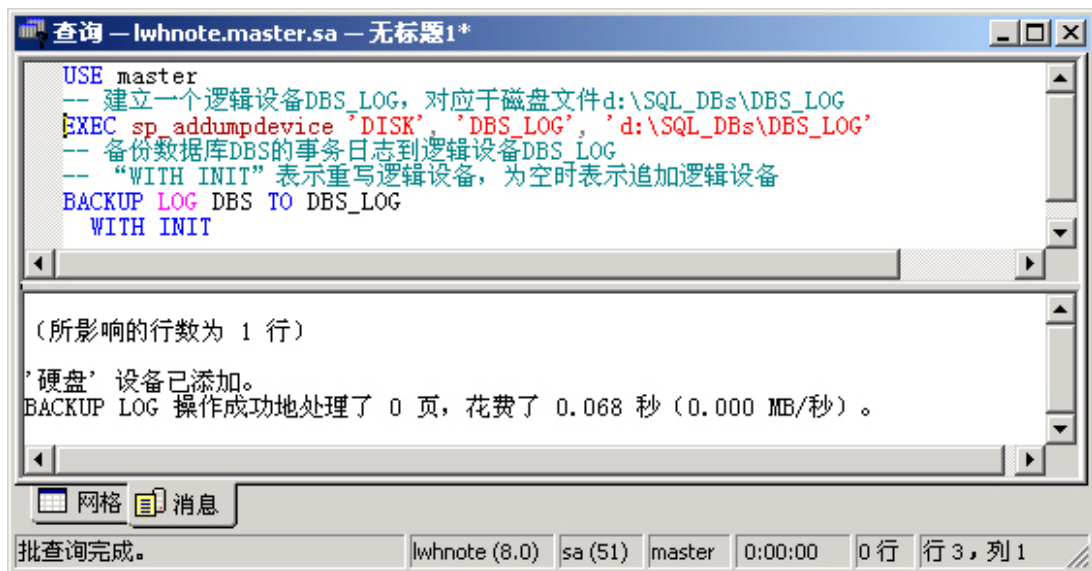
C. 完全备份数据库

如：完全备份数据库 DBS 到一个磁盘文件 d:\SQL_DBS\DBS_BAK。



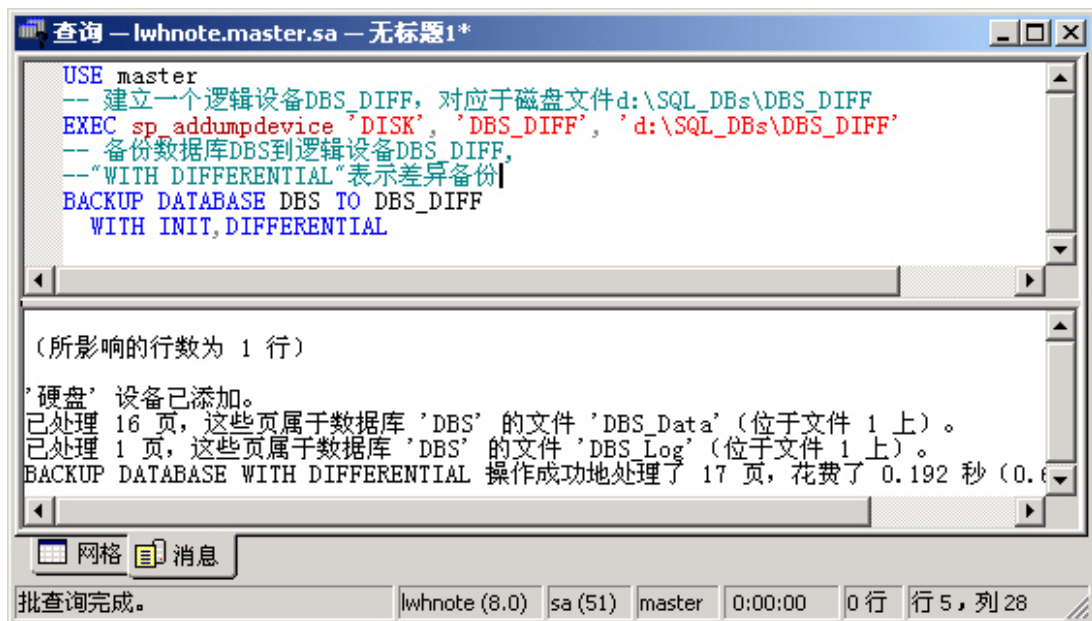
D. 备份事务日志

如：备份数据库 DBS 的事务日志到一个磁盘文件 d:\SQL_DBS\DBS_LOG。

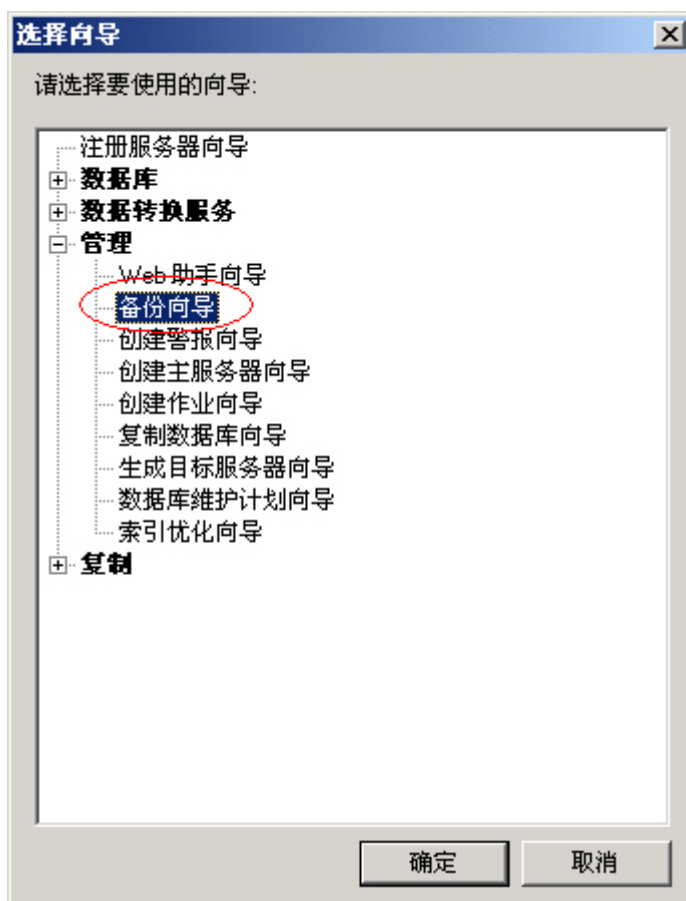
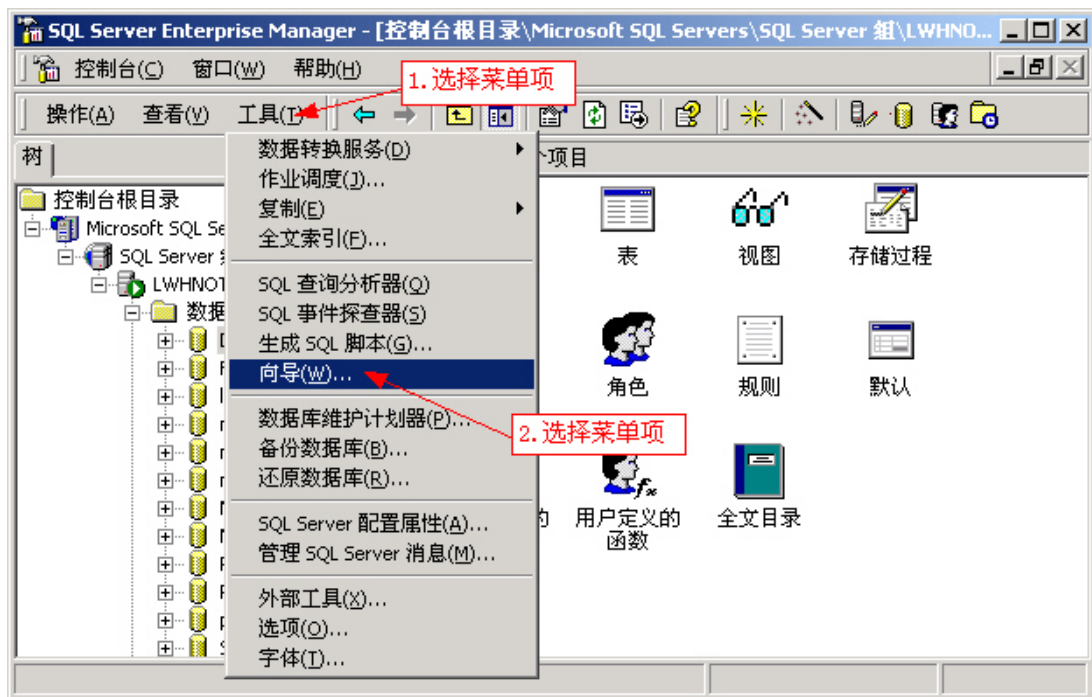


E. 差异备份数据库

如：差异备份数据库 DBS 到一个磁盘文件 d:\SQL_DBS\DBS_DIFF。



三、用备份向导进行备份



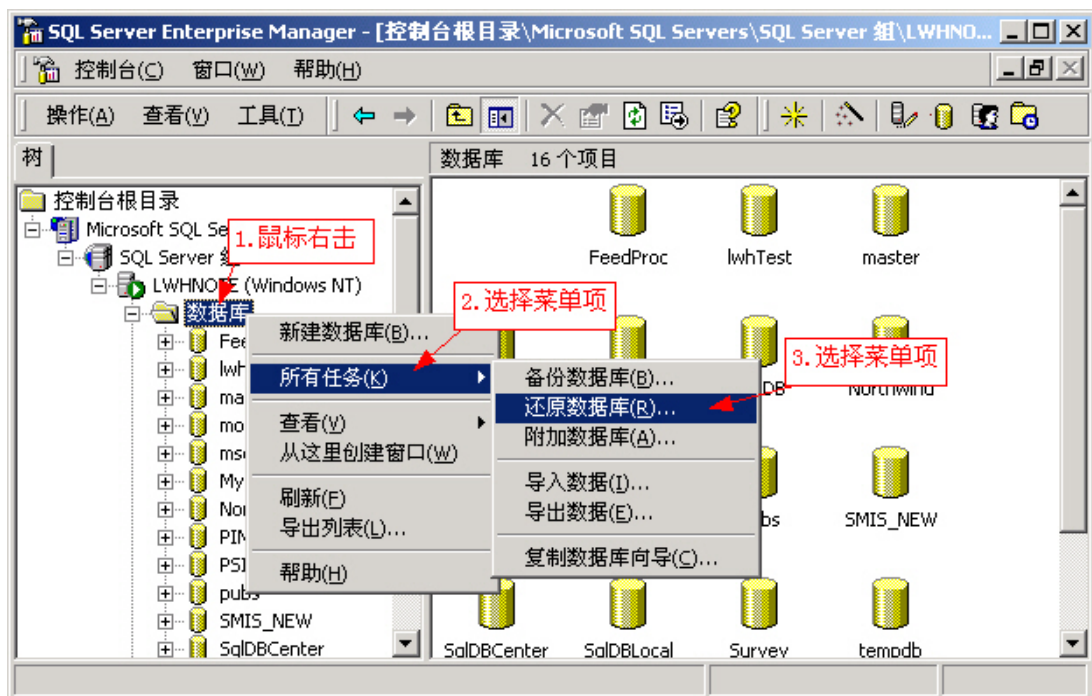
接下来，根据提示进行操作。

1.7.3 数据库恢复

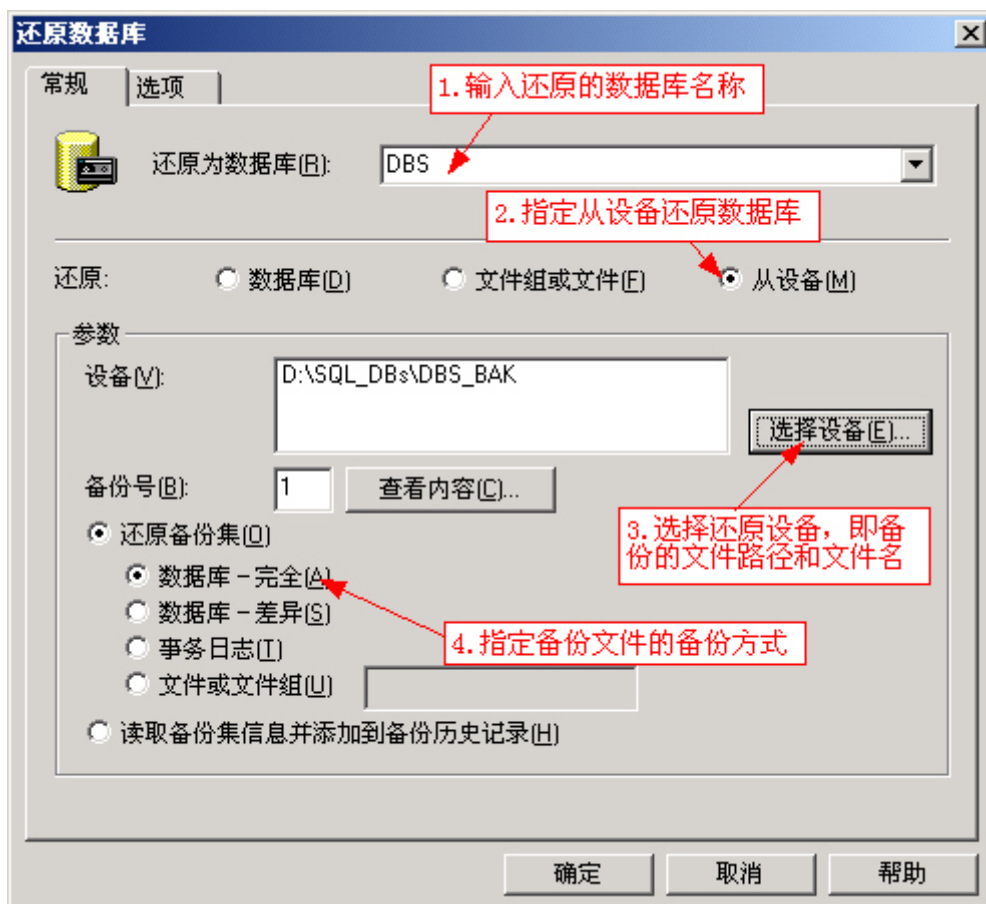
 帮助索引: <恢复数据库-SQL Server>

一、用企业管理器进行恢复

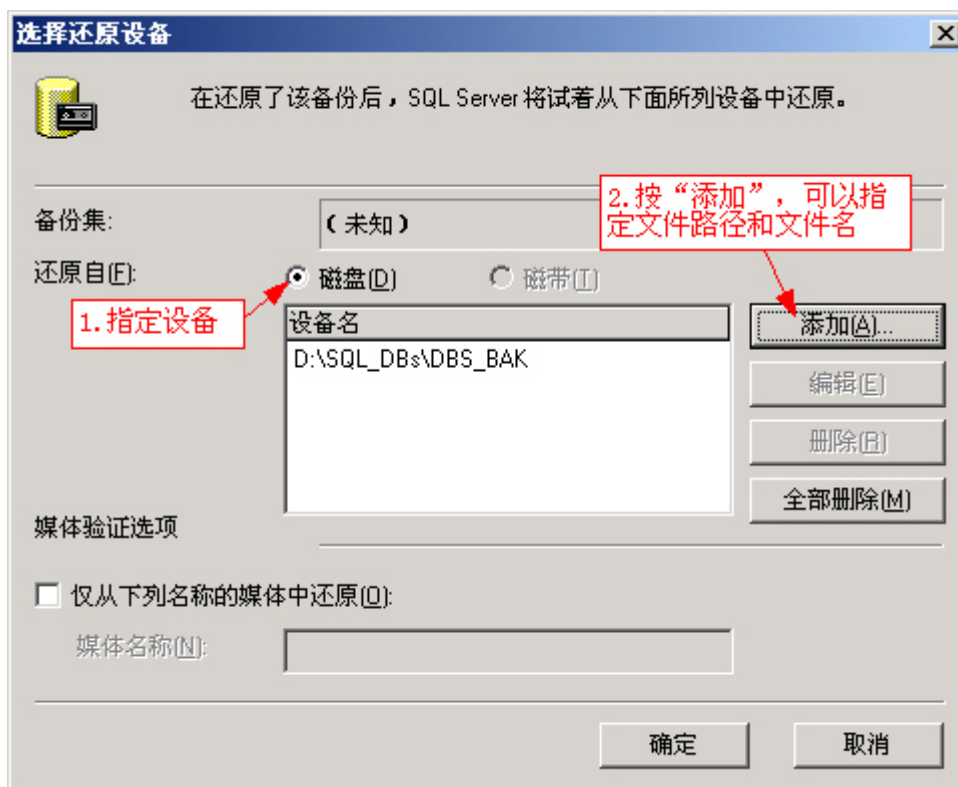
如：完全备份数据库 DBS 到一个磁盘文件 d:\SQL_DBS\DBS_BAK。



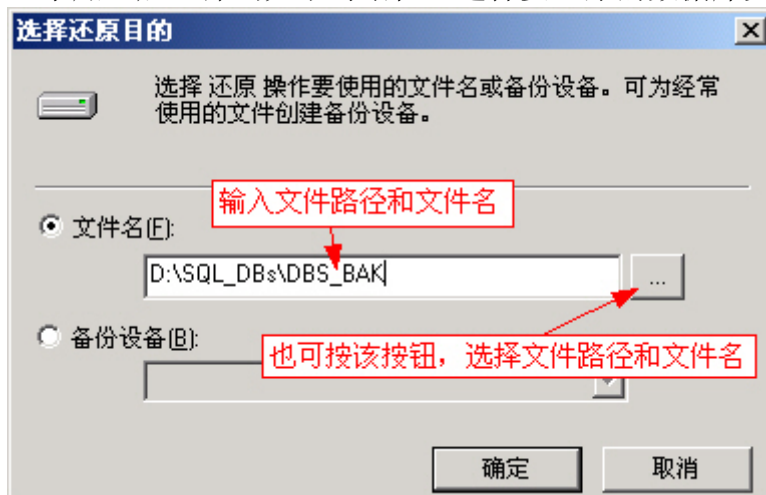
指定还原的数据库名称、还原的数据库文件（以前的备份数据库文件）、还原方式（以前的备份方式）等。



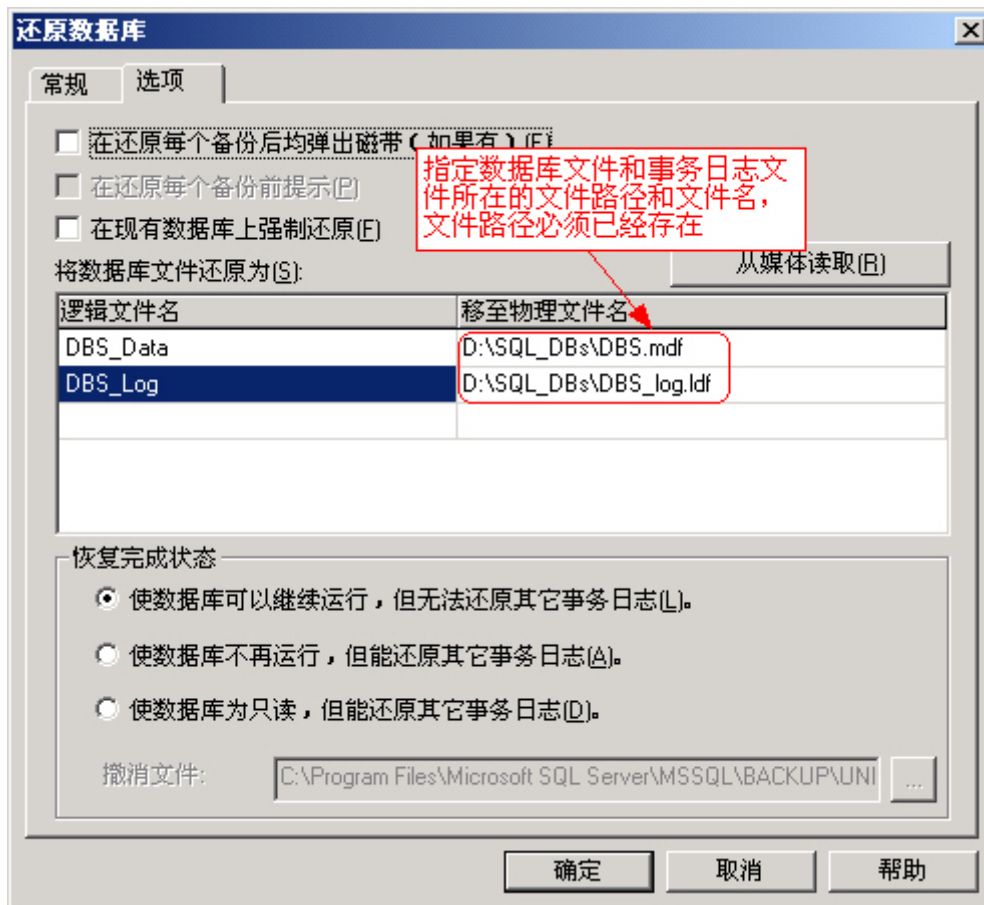
按下“选择设备”后，弹出如下对话框



按下“添加”后，弹出如下对话框，选择要还原的数据库文件



设置还原的“选项”参数，主要指定目的数据库文件和事务日志文件

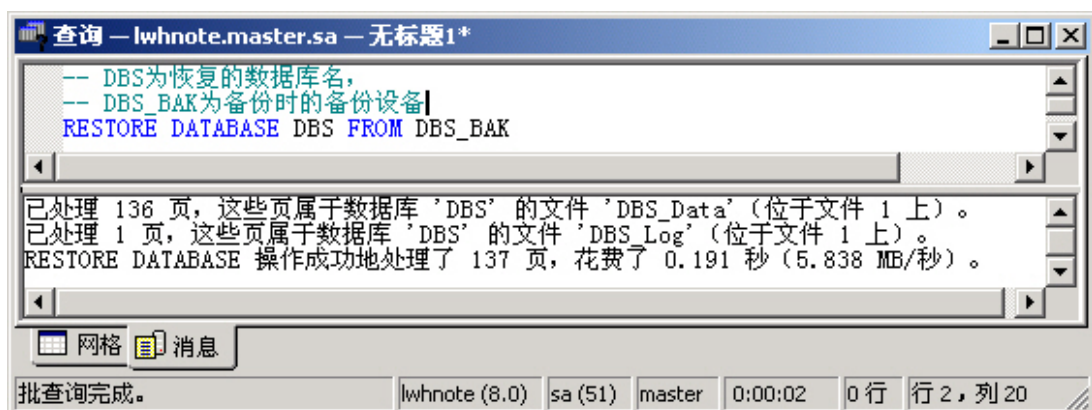


二、用查询分析器进行恢复

在备份前必须首先建立逻辑设备（可以理解为对应于磁盘上的一个文件），然后再进行备份。若指定的逻辑设备已经存在，可以删除逻辑设备，还可以查询当前的逻辑设备。

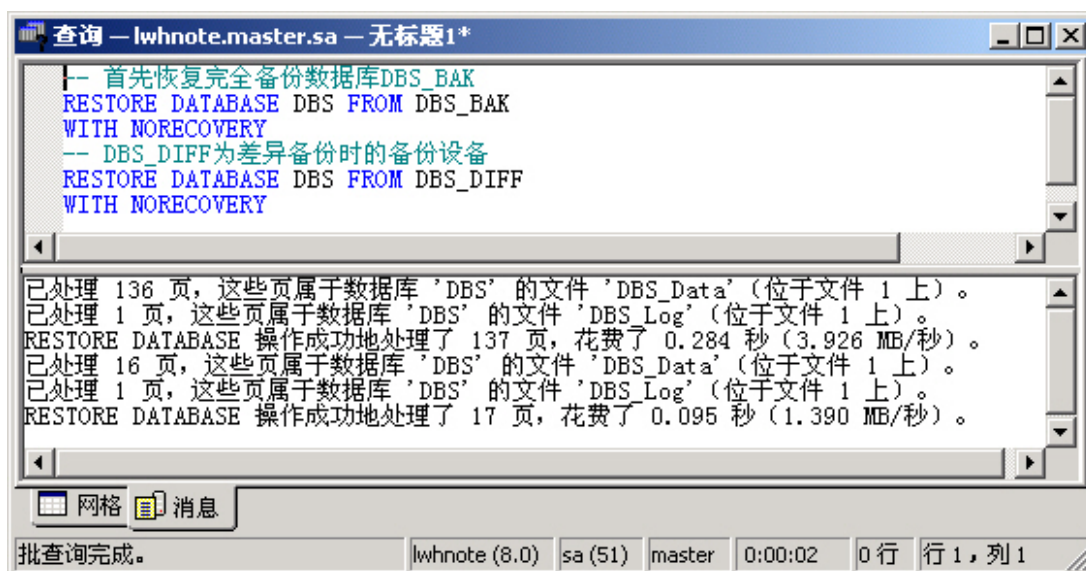
A. 恢复完全数据库备份

如：恢复完全数据库备份 DBS_BAK 为数据库 DBS



B. 恢复部分数据库备份

如：首先恢复完全备份数据库 DBS_BAK 后，再恢复部分数据库备份 DBS_DIFF 为数据库 DBS。



注：当还原数据库备份和多个事务日志时，或在需要多个 RESTORE 语句时（例如在完整数据库备份后进行差异数据库备份），SQL Server 要求在除最后的 RESTORE 语句外的所有其它语句上使用 WITH NORECOVERY 选项。

C. 恢复事务日志备份

如：首先恢复完全备份数据库 DBS_BAK 后，再恢复事务日志备份 DBS_LOG 为数据库 DBS。



1.8 实验内容

阅读本实验 1.1~1.7 的内容，完成下面的实验内容。

一、数据库管理

① 使用企业管理器创建一个 10MB 的数据库 myDB，数据库文件为

d:\SQL_DBs\myDB.mdb，事务日志文件为 d:\SQL_DBs\myDB_log.ldf。完成后，在查

询分析器中使用 **DROP DATABASE** 删除这个数据库，然后用企业管理器检验是否已经删除这个数据库。

② 使用查询分析器创建一个学生成绩管理系统数据库 **DBS**，数据库文件为

d:\SQL_DBS\DBS_Data.mdb，事务日志文件为 **d:\SQL_DBS\DBS_Log.ldb**，其它参数使用缺省参数。

二、[用户和安全性管理](#)

① 使用企业管理器在你的服务器中建立一个名字为“**my**”的登录用户，它使用 **SQL Server** 认证，登录口令为“**mypass**”，缺省访问数据库 **master**，服务器角色是系统管理员，可以访问数据库 **master**、**pubs**、**Northwind**。定义后，通过企业管理器检验三个数据库中是否有该数据库用户“**my**”，然后用查询分析器删除该用户。

② 使用企业管理器在你的服务器中建立一个以你自己名字命名的登录用户（如：**lwh**），它使用 **SQL Server** 认证，用户口令为登录用户名+“**pass**”（如：**lwhpass**）。缺省访问的数据库为 **JXGL**，服务器角色是系统管理员，可以访问数据库 **master**、**JXGL**。

三、[备份和恢复](#)

① 使用企业管理器完全备份一个用户数据库（如：**myDB**），备份前记住备份数据库的属性（如：数据库名称、数据库文件名、事务日志文件名、初始容量大小等），然后删除该数据库，检验该数据库是否已经删除，之后，用查询分析器恢复刚才备份的数据库，并检验恢复后数据库的属性是否与备份前的数据库属性一致。

② 使用企业管理器完全备份学生成绩管理数据库 **DBS** 到 **D:\SQL_DBS\DBS_ALL** 中。
注意：备份时均采用“重写”的方式重写备份设备。

四、为下次实验准备

① 建立学生成绩管理系统 **DBS** 并备份到文件 **DBS_ALL** 中，并用软盘或 U 盘保存。

② 建立一个用你自己名字命名的 **SQL Server** 登录用户。

1.9 思考题

- ① 如何修改数据库的数据库文件名和事务日志文件名？
- ② 如何在企业管理器中建立一个有文件组的数据库？
- ③ 如何让一个登录用户名能够访问某个数据库？登录用户名与数据库用户名有什么关系？
- ④ 你能理解登录用户的角色和数据库用户的权限？

- ⑤ 你能理解备份设备与备份物理文件间的关系？
- ⑥ 在使用查询分析器时，如何改变当前使用的数据库？如何改变登录用户？