

信息工程学院实验指导书系列

数据库原理实验指导

洪源 编



北京科技大学信息工程学院

计算机科学与技术系

二〇〇六年九月

二〇〇九年四月修订

前 言

《数据库原理实验指导》是北京科技大学本科计算机专业学生数据库课程实验用教材。

全书共两部分。第一部分较为详细地介绍了 SQL Server 2000 的使用方法中与本书实验相关的内容，供学生在实验前进行学习。第二部分是一个综合设计实验，学生通过该实验提高综合运用所学知识的能力。附录中给出了综合设计实验的参考文档。

本书第二部分的综合设计实验建议 4 个学时完成。

在本书的编写过程中，研究生关勤同学参加了资料的搜集和整理工作。

在本书的使用过程中，计算机科学与技术系 2006 级王力同学发现了本书中的疏漏并提出了修改建议；计算机科学与技术系 2006 级邱晓峰同学也发现了本书中的一些疏漏。在此表示感谢。

书中仍难免存在其他不足之处，敬请读者批评指正。

编者
2009 年 4 月

目 录

前 言	2
目 录	3
第一部分 预备知识	5
第一章 SQL Server 2000 简介	5
第二章 SQL Server 2000 的服务器的使用	7
一、服务器的启动与停止	7
二、服务器的连接和断开	8
第三章 SQL Server2000 的企业管理器的使用	10
一、企业管理器的启动与关闭	10
二、数据库的创建与管理	11
三、表的创建与管理	17
四、约束的创建与管理	22
五、输入、修改、删除数据	31
第四章 SQL Server2000 的查询分析器的使用	33
一、查询分析器的启动与关闭	33
二、执行 SQL 语句	35
三、查询结果的显示模式	37
四、创建和运行脚本文件	38
五、查询分析器的常用快捷键	42
第五章 SQL Server 2000 用户与权限管理	43
第六章 数据库备份机制	63
一、为什么要进行数据库备份	63
二、SQL Server 2000 数据库备份的种类	63
三、SQL Server 2000 数据库备份的方法	65
第七章 数据库恢复机制	78
一、为什么要进行数据库恢复	78
二、SQL Server 2000 数据库恢复的种类	78
三、SQL Server 2000 数据库恢复的方法	79
第八章 练习	85
第二部分 综合设计实验	90
实验一 课程成绩管理数据库	90
一、实验目的	90
二、实验内容	90

三、实验步骤.....	94
四、思考题.....	95
五、实验报告内容及要求.....	95
附录一：实验报告格式.....	96
附录二：实验参考文档.....	97

第一部分 预备知识

第一章 SQL Server 2000 简介

SQL Server 是由 Microsoft 公司开发和推广的关系数据库管理系统 (DBMS)，最初由 Microsoft、Sybase 和 Ashton-Tate 三家公司共同开发。第一个 OS/2 版本在 1998 年推出。1990 年后，Ashton-Tate 公司中途退出了 SQL Server 的开发。1992 年，SQL Server 移植到 Windows NT 上之后，Microsoft 公司专注于开发和推广 SQL Server 的 Windows NT 版本，而 Sybase 公司则较专注于 SQL Server 在 UNIX 操作系统上的应用。SQL Server 近年来不断更新版本。Microsoft 公司于 1996 年推出了 SQL Server 6.5 版本，1998 年推出了 SQL Server 7.0 版本。SQL Server 2000 是 Microsoft 公司于 2000 年推出的最新版本。

SQL Server 2000 继承了 SQL Server 7.0 高性能、可靠性和可扩充性的优点，同时又进了许多新特性，使它成为一个杰出的数据库平台。

SQL Server 2000 是一个真正的客户机/服务器的数据库系统，其基本体系结构如图 4-1 所示，它把工作负载划分在服务器上运行的任务和是客户机上运行的任务。客房机制的应用程序负责数据处理、提供用户操作界面，可以运行于一台或多台客户机上。SQL Server 服务器管理数据库和在多个客户机请求之间分配可用的服务器资源，如内存、网络带宽和磁盘操作等。客户机/服务器体系结构使用户能为各种环境最优化地设计和布局应用程序。

SQL Server 2000 还可以在多种 Windows 环境下安装运行，同时与 Windows NT 完全集成。与 Windows NT 服务的集成，SQL Server 是作为 Windows NT 的一个服务运行的，通过设置，可以在启动 Windows 操作系统时自动启动 SQL Server；与 Windows NT 安全系统集成，由于 Windows NT 服务器具有大型商业系统所需要的安全性和可靠性，因此 SQL Server 2000 可以利用 Windows NT 的加密特性获得数据库的安全，使用户可以使用同一个用户名和口令访问 SQL Server 2000 与 Windows NT。SQL Server 2000 还有许多其他优点，这里不再一一介绍。

在安装了 SQL Server 2000 以后，计算机的[开始]菜单中会产生新的程序组。

各个工具和应用程序的功能简述如下：

- 1) 查询分析器 (Query Analyzer) 通过输入和执行 Transact-SQL

语句与 SQL Server 进行通信。所有的远程工作站及 SQL Server 开发人员使用的计算机都应该安装这个工具。这个工具使开发人员能够直接访问 SQL Server，交互地执行 Transact-SQL 语句和 SQL 脚本。

- 2) 导入和导出数据 (Input and Export Data) 用于进行 SQL Server 和其他数据库数据的相互转换。
- 3) 服务管理器 (Service Manager) 用来启动、停止、暂停或者继续 SQL Server 数据库服务。它是在服务器端实际工作时最有用的实用程序，所提供的服务包括 SQL Server、SQL Server Agent 及 MSDTC (Microsoft Distributed Transaction Coordinator, 微软分布式事务协调器)。
- 4) 服务器网络实用工具 (Server Network Utility) 用于配置服务器端网络通信协议的相关参数等。
- 5) 客户端网络实用工具 (Client Network Utility) 用于配置客户端网络通信协议的相关参数。
- 6) 联机丛书 SQL Server 2000 提供了大量的联机帮助文档 (Online Book)，它具有索引和全文搜索工具，能够指导和帮助用户完成各种任务。
- 7) 企业管理器 (Enterprise Manager) 是基于微软管理控制台 (Microsoft Management Console) 的公共服务器管理环境，是 SQL Server 提供的最主要的数据库管理工具，大部分的数据库管理工作都可以在其中完成。企业管理器以树形结构的形式来管理数据库服务器、数据库和数据库中的对象，并能在同一界面管理网络中的多个 SQL Server。
- 8) 事件探查器 (Event Detector) 使系统管理员得以监视 SQL Server 实例中的事件。可以捕获有关事件的数据并将其保存到文件或 SQL Server 表中供以后分析。
- 9) 在 IIS 中配置 SQL XML 支持 是用于 SQL Server 的 IIS 虚拟目录管理实用工具。

第二章 SQL Server 2000 的服务器的使用

一、服务器的启动与停止

1. 服务器的启动

SQL Server2000 提供了三种方法来启动服务器.

(1) 用 SQL Server 服务管理器启动

启动步骤如下:

- 1) 在 Windows 操作系统中,依次选择“开始”→“程序”→“Microsoft SQL Servers”→“服务管理器”菜单项,即可启动服务管理器,出现“SQLServer 服务管理器”窗口,如图 2.1 所示。
- 2) 在“服务器”和“服务”栏中选择要启动的服务器和服务。
- 3) 单击“开始/继续”按钮以启动服务器。一旦启动成功,Windows 系统的任务栏中会出现指示 SQL Server 服务器正在运行的图标。



a) 停止



b) 启动

图 2.1 启动服务器

(2) 自动启动

在 SQL Server 服务管理器窗口(见图 2.1)中,选中“当启动 OS 时自动启动服务”选项,则以后每当启动 Window 时,SQL Server 服务器将自动启动,Windows 系统的任务栏中同样会出现指示 SQL Server 服务器正在运行的图标。

(3) 用企业管理器启动

在企业管理器的“SQL Server 组”中选中启动的服务器，然后用鼠标右击该服务器，在弹出的菜单上选择“启动”选项(见图 2.2)，即可启动服务器。

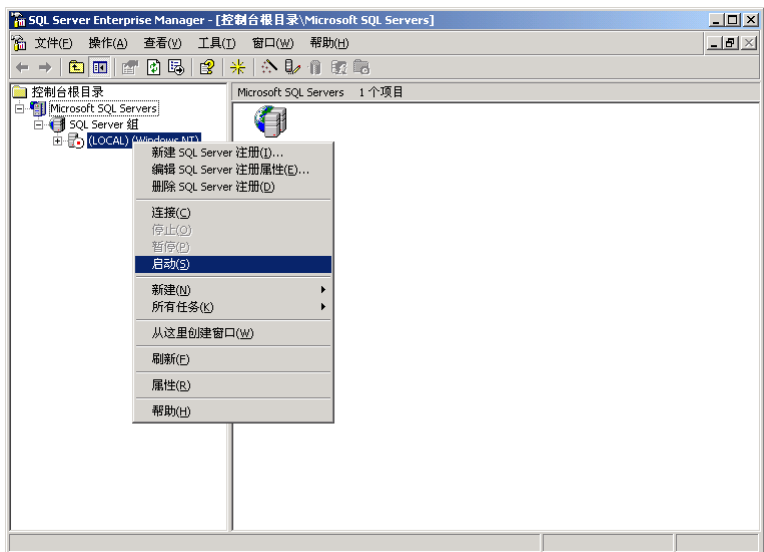


图 2.2 用企业管理器启动服务器

2. 服务器的停止

暂停或停止服务器的方法与启动服务器的方法类似，只需在相应窗口选择“暂停”或者“停止”即可。通常，在停止服务器前会先进行暂停操作，因为一旦暂停 SQL Server 服务，则不允许任何新的连接用户连接服务器，而原来的连接用户仍然可以正常工作，这样可以保证正在进行的操作不会被意外地终止。

二、服务器的连接和断开

服务器的“连接”是指数据库合法用户登录到服务器上，“断开”是指已连接到数据库服务器的用户终止连接状态，从服务器上退出。

1. 服务器的连接

在 SQL Server 企业管理器中选中所要连接的服务器，然后用鼠标右击该服务器，在弹出的菜单上选择“连接”选项。如果在注册服务器时选择“在连接时提示输入 SQL Server 帐户信息选项”，那么每次用户试图连接服务器时都会被提示输入登录名和密码。只有在数据库服务器已启动的情况下，连接服务器才可能成功。

2. 服务器的断开

在同一菜单上选择“断开”选项,即可切断当前登录用户与服务器的连接。
在关闭服务器时,会自动断开登录用户的连接。

第三章 SQL Server2000 的企业管理器的使用

企业管理器不仅能够配置系统环境和管理 SQL Server，而且由于它能够以树形结构的形式来显示所有的 SQL Server 对象，因而所有 SQL Server 对象的建立与管理都可以通过它来完成。利用企业管理器可以完成的操作有：管理 SQL Server 服务器，建立与管理数据库，建立与管理表、视图、存储过程等数据库对象，备份数据库和事务日志，恢复数据库，复制数据库，管理用户帐户，建立 Transact-SQL 命令语句等。下面分别介绍企业管理器最常用的使用功能。

一、企业管理器的启动与关闭

1. 企业管理器的启动

在 SQL Server2000 服务器已经启动的前提下，启动企业管理器的步骤如下：

- 1) 单击 Windows 【开始】菜单。
- 2) 依次单击“程序”→“Microsoft SQL Server”→“企业管理器”，即可进入如图 3.1 所示界面。

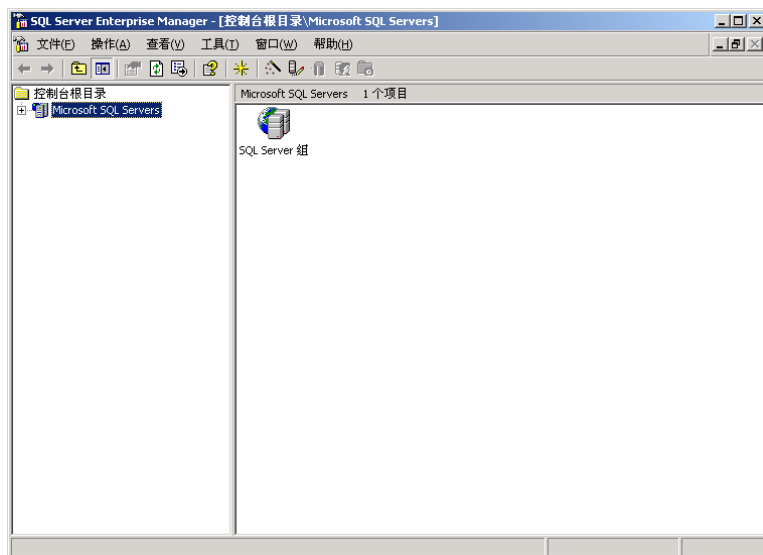


图 3.1 “企业管理器”窗口

企业管理器使用的是树形结构。树形结构的根节点是“控制台根目录”，是所有服务器的总控制节点。SQL Server 允许用户注册多个服务器组，而图 3.1 只显示了一个服务器组。一般情况下，我们会按照功能用途来划分服务器组。“SQL Server 组”是系统默认的服务器组，用户成功安装的服务器就注册在这个服务器组的下面。用户可以删掉注册信息，然后重新注册。要打开一个包含子目录或者其他内容的文件夹，可以单击文件夹左侧的加号(+) (或者双击该文件夹，在选定文件夹时使用键盘上的右箭头键)。

每个注册服务器直接包含的对象有数据库、数据转换服务、管理、复制、安全性、支持服务和 Meta Data Services (元数据服务)。旁边有加号(+)的组件可以被进一步展开。

2. 企业管理器的关闭

关闭企业管理器有两种方法：

- 1) 单击窗口的标题栏最右侧的控制图标来关闭企业管理器，这是 Windows 风格的应用程序所通用的关闭窗口的方法。
- 2) 单击企业管理器的菜单栏中的“文件”菜单，然后单击“退出”选项即可。

二、数据库的创建与管理

1. 使用企业管理器创建数据库

创建数据库是创建表及其他数据库对象的第一步。在 SQL Server 中，创建数据库需要拥有特定的权限。在默认情况下，只有系统管理员和数据库拥有者可以创建数据库，但是系统管理员也可以将这个权限授予其他用户。数据库被创建后，创建数据库的用户自动成为该数据库的拥有者。

所在的新数据库都是系统样本数据库 model 副本，所以每个数据库都由以下几部分组成：关系图、表、视图、存储过程、用户、角色、规则、默认、用户自定义数据和用户自定义函数。model 数据库的大小为 1MB，新创建的数据库容量则自然不会比 model 数据库小。一台服务器上最多可以创建 32767 个数据库。


创建数据库的过程就是为数据库确定名称、所占用的存储容量和所存放的数据库文件位置及其相关特性的过程。数据库的名称必须遵循 SQL Server 的命名规范，建议使用意义的名称。新建数据库的信息存放在系统库 master 中，当新数据库被创建时。SQL Server 自动地更新 master 数据库的“sysdatabases”系统表。因而可以使用 SELECT 语句查询“sysdatabases”来获得数据库信息。

在创建新数据库的数据文件和事务日志文件时，建议在数据文件使用的

文件逻辑名上加入“_data”后缀，在事务日志使用的文件逻辑名上加入“_log”后缀(例如“XXX_dafa”和“XXX_log”)，这样标识比较明确。

数据库创建的方法有三种；使用企业管理器创建、使用数据库创建向导创建，以及使用 Transact-SQL 语句创建。使用企业管理器创建数据库比其他方法都来得简捷，下面以创建学生选课数据库(StudentsInfo)为例说明如何使用企业管理器创建数据库。

- 1) 单击 Microsoft SQL Server 服务器组，进入已经配置注册的服务器(本例中为 GUAN)，单击选中“数据库”文件夹。
- 2) 选择“操作”→“新建数据库”项；或在“数据库”文件夹上右击鼠标，在弹出的快捷菜单上选择“新建数据库”(见图 3.2)；

或者单击工具栏上图标。系统将进入创建数据库的“属性”窗口(见图 3.3)。

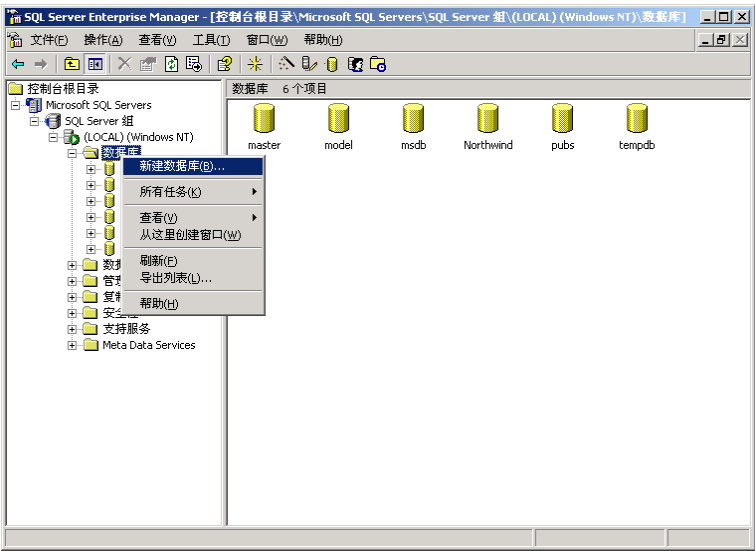


图 3.2 创建数据库



图 3.3 “常规”选项卡

该窗口有 3 个选卡：常规、数据文件、事务日志、事务日志。在“常规”选项卡中，要求用户输入数据库名称以及排序规则名称，输入学生选课数据库的名称“StudentsInfo”，排序规则按默认设置不变(见图 3.3)。注意，数据库的名称必须遵循 SQL Server 命名规范，并且不能与已有的数据库名重复。“数据文件”选项卡中输入数据库文件的逻辑名称、存储位置、初始容量大小和所属文件组名称(见图 3.4)。“事务日志”选项卡中设置事务日志文件信息(见图 3.5)。

3) 单击“确定”按钮，数据库的创建工作完成。

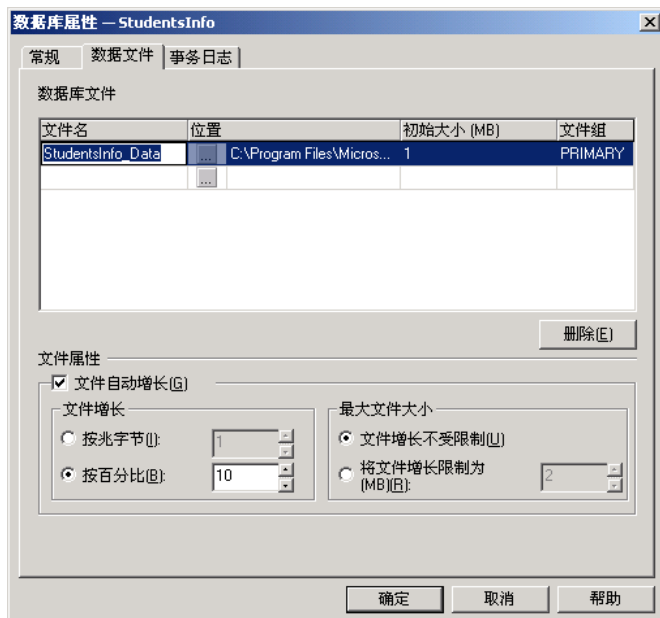


图 3.4 “数据文件”选项卡




图 3.5 “事务日志”选项卡

2. 使用企业管理器修改数据库

我们可以使用企业管理器对已创建的数据库进行属性查看和修改创建时的某些设置，具体方法说明如下：

选中要修改的数据库 **StudentsInfo** ,单击鼠标右键，在弹出的菜单中选择“属性”(如图 3.6 所示)，出现如图 3.7 所示的“**StudentsInfo** 属性”高置对话框。

- 1) “常规”选项卡显示了状态、所有者、创建日期、大小等数据库基本信息；备份信息；维护信息。
 - 2) “数据文件”选项卡和“事务日志”选项卡中的内容与创建数据库时类仪，可以在此重新设定。
 - 3) 在“文件组”选项卡中，可以添加或删除文件组。若文件组长中包含文件，则不能删除，必须将文件组清空后才可删除该文件组。
 - 4) 在“选项”选项卡中，可以设置数据库的访问、故障还原、兼容性等属性，如图 3.8 所示。
 - 5) 在“权限”选项卡中，可以设置用户对数据库的使用权限。
3. 使用企业管理器删除数据库

对于不再需要的数据库，用户可以删除它以释放在磁盘上所占用的空间。用鼠标右击要删除的数据库，在出现的快捷菜单单击“删除”选项；或者选中数据库文件夹后单击图标 。系统将会出现一个数据库将要被删除的警告消息，单击“确定”按钮后就会删除这个数据库。

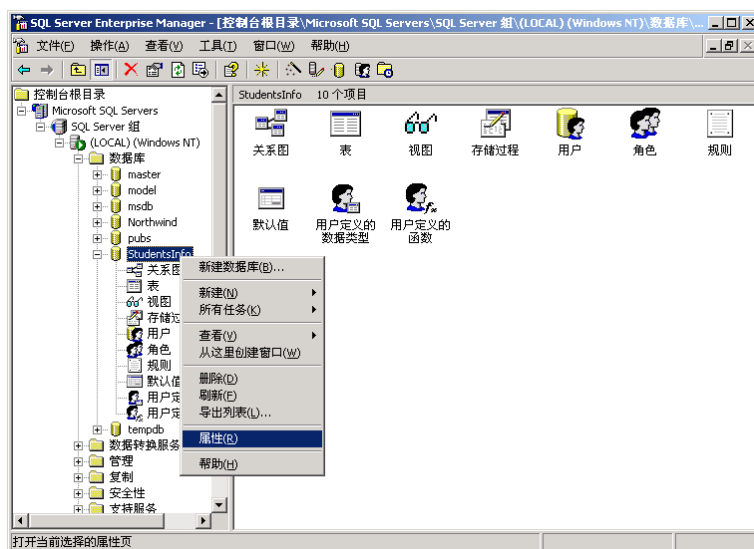


图 3.6 查看和修改数据库属性

注意：不能删除 master、tempdb、model、msdb 这 4 个系统数据库，任何时候删除数据库前都应先备份 master 数据库。

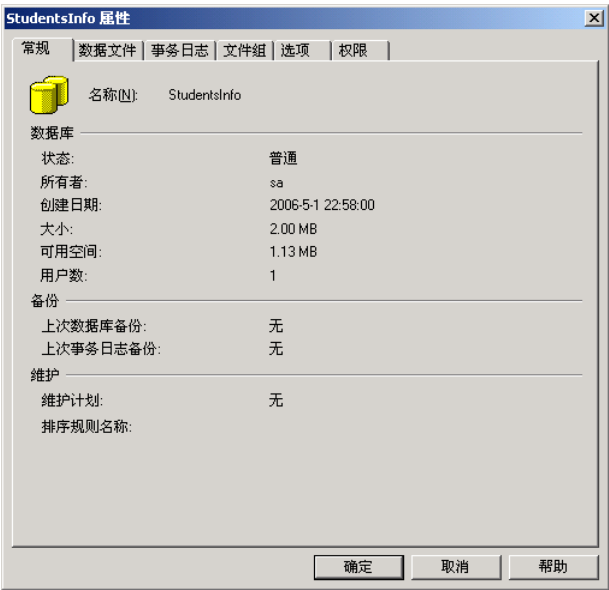


图 3.7 数据库“属性”窗口

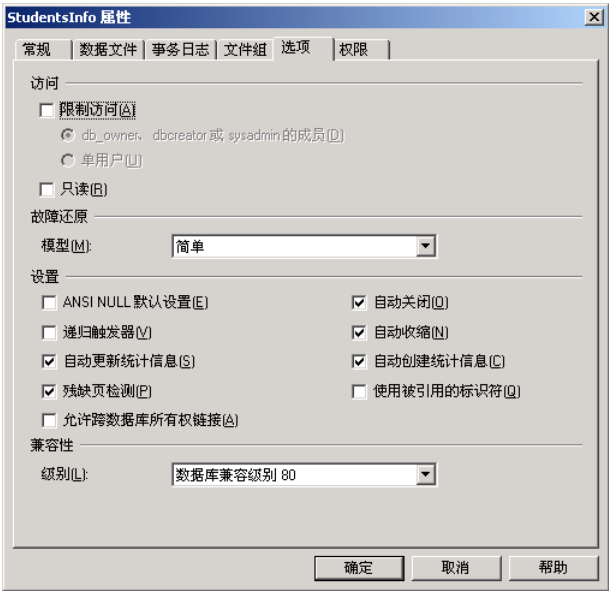


图 3.8 “选项”选项卡

三、表的创建与管理

1. 使用企业管理器创建表

创建了数据库之后，接着就应该创建表来存储数据了。表是在使用数据库的过程中接触最多的数据库对象。表定义为列的集合，数据在表中是按行和列的格式组排列的，每行代表惟一的一条记录，行的惟一性可以通过定义主键来实现。在 SQL Server2000 中，每个数据库中最多可以创建 200 万个表。用户创建数据库表时，最多可以定义 1024 列，也就是说，可以定义 1024 个字段，每行最多 8060 字节的用户数据。

SQL Server2000 提供了两种方法创建数据。


- 1) 利用企业管理器(Enterprise Manager)创建表。
- 2) 利用 Transact-SQL 语句中的 CREATE 命令创建表。下面举例说明使用企业管理器为 StudentsInfo 数据库创建表 Students (如表 3.1 所示)的过程。

表 3.1 表 Students 的结构示意

字段名	含义	数据类型	是否允许为空	备注
Sno	学号	CHAR(10)	不允许	主键
Sname	姓名	CHAR(8)	不允许	
Ssex	性别	CHAR(2)	允许	非空值，取‘F’或‘M’
Sage	年龄	INT	允许	
Sdept	系	CHAR(15)	允许	默认为‘Computer’

创建数据库表的第一种方法的操作步骤如下：

- 1) 进入企业管理器，打开想要创建亲表的数据库 StudentsInfo，使作鼠标右键单击“表”，在弹出的菜单中选择“新建表”

项(见图 3.9)(或者在工具栏中单击图标 )、系统就会弹出新建表对话框，在第一行的“列名”里，输入“Sno”，则给出的默认数据类型是 char，长度是 10，不允许为空(如图 3.10 所示)。可以单击“数据类型”的下拉列表来选择或调整一个数据类型，“长度”列和“允许空”列都可以进行手动的更改。对于 Sno 而言，投置类型为 char，长度为 10，不允许为空。随后，依次输入字段名 Sname、Ssex、Sage 及相应的数年

据类型、字段长度等设置值。

“允许空”(即该项被“打勾”)表示该字段对应的数据可以接受空值(NULL)。在数据库中 NULL 是一个特殊值,用来表示未知、不可用或以后添加数据, NULL 不同于空字符或数字 0,也不同于零长度字符串。“不允许空”(NOT NULL)则表示该字段对应的数据拒绝空值(NULL)。

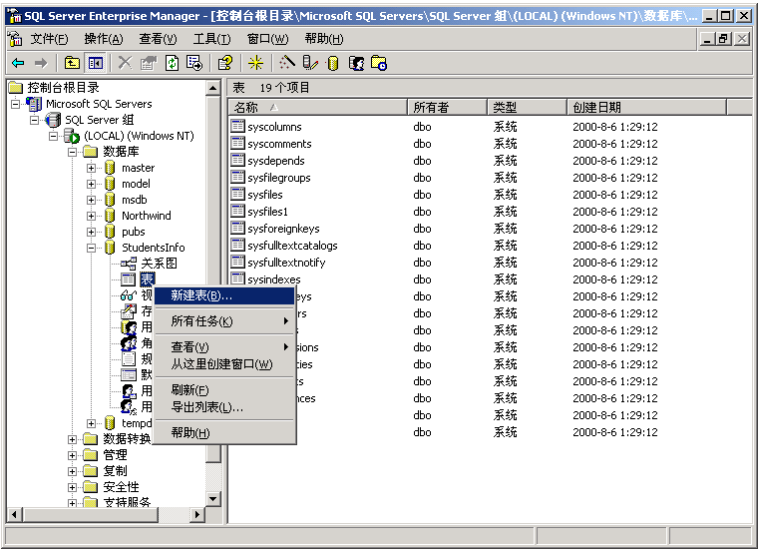


图 3.9 创建新表步骤一

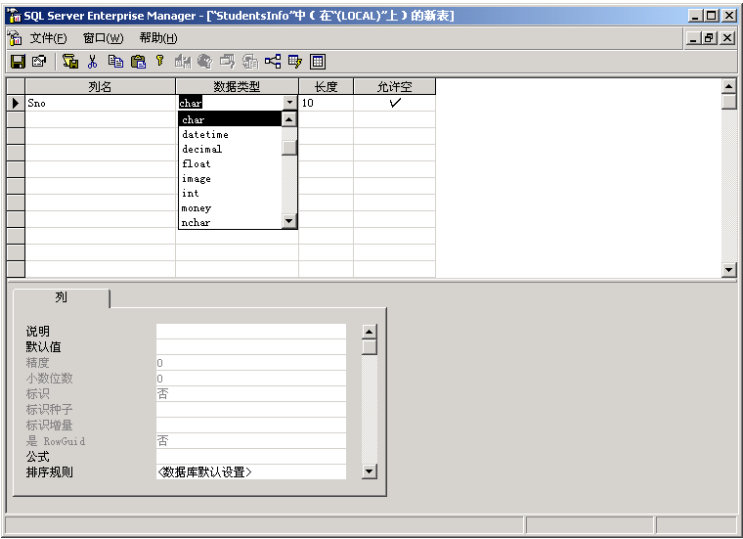


图 3.10 创建新表步骤二

表的下面有一个对应当前的列的“列”选项卡(如图 3.11 所示),在该选项卡中可以备注一些与当前列相关的内容,如对当前列的描述、默认值(表示在用户未输入数据时列的取值)、列上的计算色式,以及列数据的排序规则。

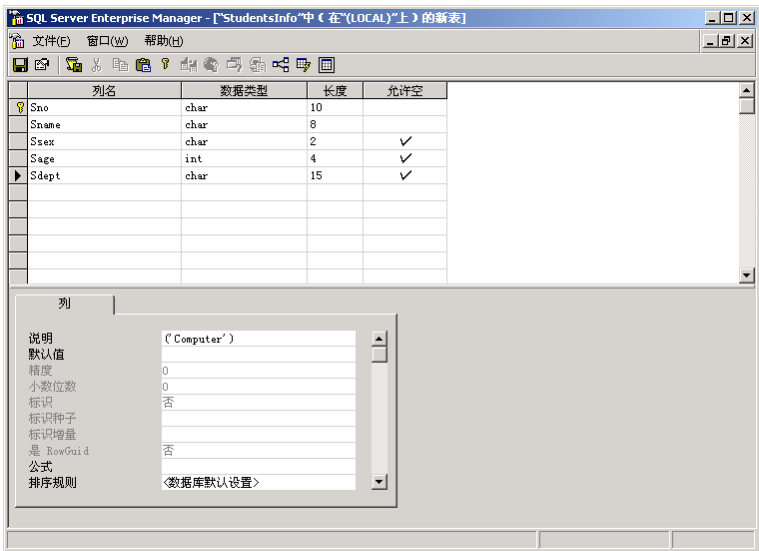




图 3.11 为字段 Sdept 设置默认值

- 2) 为表设置主键。主键是指表中某列或者某些列的组合,其值能惟一地标识表中的每一行,通过主键可强制表的实体完整性。当创建或更改表时可通过定义 **PROMARY DEY** 约束来创建主键。在要设置为主键的字段 **Sno** 上右击鼠标,选择“设置主键”选项;或者选中该字段后单击工具栏中  按钮,则在字段左侧出现钥匙形的图示,表示当前字段被设置为主键。
 - 3) 保存刚刚创建的表。单击工具栏上的  图标系统会出现要求输入新表名称的对话框,输入名称: **Students**,单击“确定”按钮,则在数年数据库中建立了表 **Students**,如图 3.12 所示。到此为止,我们创建了一个具有基本结构的表。但是一个具有实际应用价值的表,往往还要根据高驻地的需要,再设置其他约束。关于约束的概念和定义方法请参阅 4.3.4 节。
2. 使用企业管理器修改表
- 在创建好表以后,可以对表的结构进行更改,包括重新设置主键及字段

属性、添加或者删除列等。

在数据库 StudentsInfo 中选中 Students 表，单击鼠标右键，在弹出的菜单上选择“设计表”，则系统弹出设计表的窗口(见图 3.13)。右键单击每列会弹出一个菜单，通过这个菜单，可以进行的操作有：将当前的列设置为主键；插入新列；删除当前列；直接设置索引/键、关系(该表与其了表的依赖关系)、CHECK 的束等。也可以通过选择其中的“属性”选项来设置列的属性。

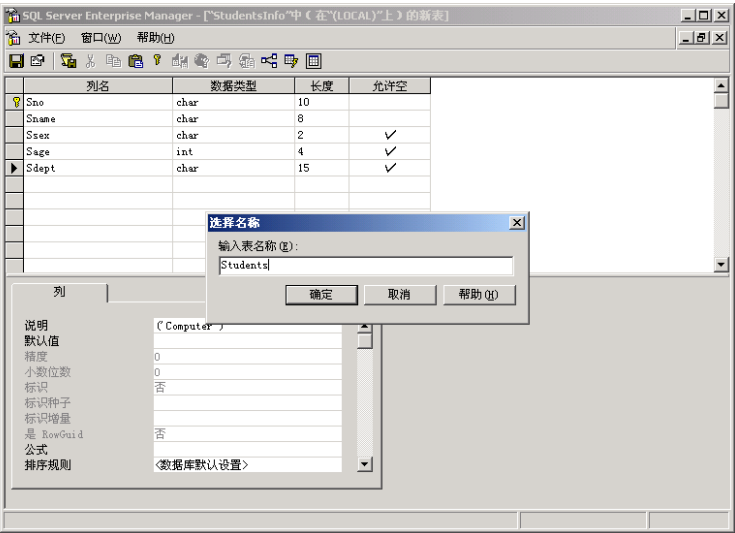


图 3.12 输入新建表对话框

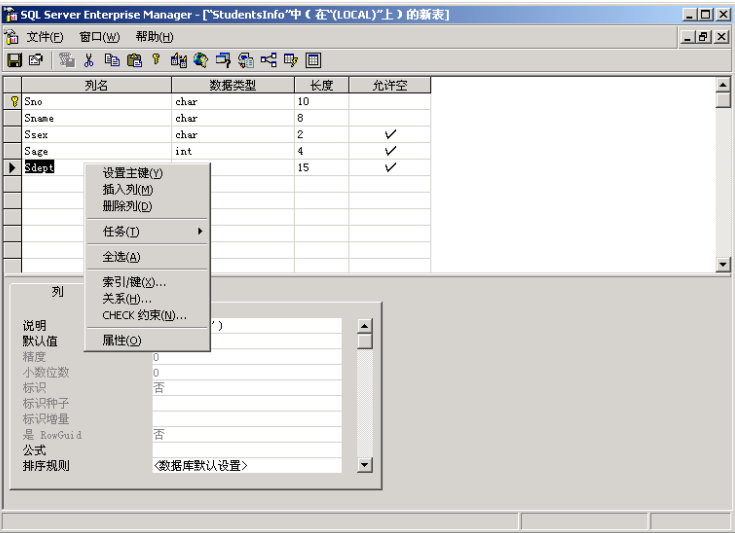


图 3.13 修改表结构的窗口

选择“属性”项后，弹出表的属性窗口如图 3.14 所示，这个窗口包含表、关系、索引/键 CHECK 约束 4 个选项卡。在“表”选项卡中、允许设定表的所有者、更改表名、设定表的标识符列(IDENTITY)、设定表的 ROWGUID 列、设定表存储的文件组、设定文件组及对表进行描述。

“标识列”(IDENTITY)属性用来指定该列的第一行标识数据(种子)，并确定要添加到种子上的增量。在向具有标识列的表中插入值时，SQL Server 通过递增种子值的方法自动生成下一个标识值。注意：一个表只能有一列定义为标识列，而且该列心须是 decimal、int、numeric、smallint、bigint 或 tinyint 的数据类型；种子和增量值的默认值均为 1，也可以任意指定；标识列不允许空值，也不能包含默认值(DEFAULT)定义或对象；在设置标识列后，可以使用 IDENTITYCOL 关键字在选择表中引用该列；可以使用 OBJECTPROPERTY 函数确定一个表是否具有标识列，使用 COLUMNPROPERTY 函数确定标识列的名称。

如果需要生成在整个数据库或全球数数据库中均为惟一的标识列，则可以使用“表的 ROWGUID 列”属性。

3. 使用企业管理器删除表

对于不需要的表，可以删除表它，以释放其在磁盘上所占用的空间。选项要删除的表，单击鼠标右键，在弹出菜单上选择“删除”项；或者单击工具栏中的图标；或者使用键盘上的 Delete 键，系统弹出如图 3.15 所示的删除对象对话框。

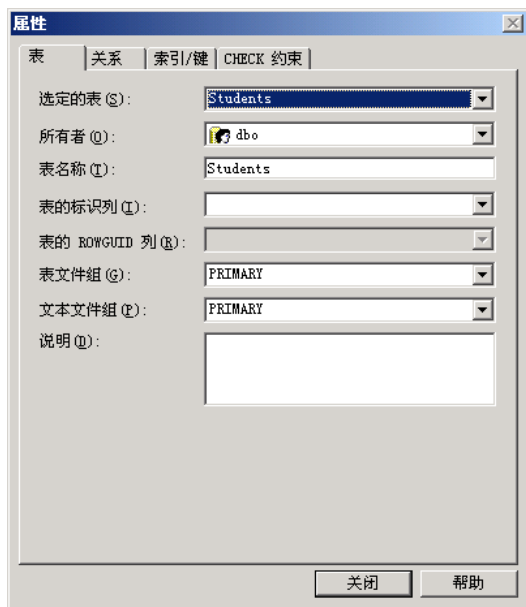


图 3.14 表的“属性”窗口图



图 3.15 删除表 Students 对话框

单击“全部除去”按钮，就会删除该表(注意：在此仅对删除做示范说明，由于 Students 表在后文中还会用到，所以这里并不做真正删除)。

四、约束的创建与管理

约束是 SQL Server 提供的强制完整性的标准机制，它通过限制字段中的数据、记录中数据和表之间的数据来自动保持数据库的完整性。SQL Server 2000 中有 5 类约束：主键约束(Primary Key Constraint)、惟一约束(Unique Constraint)、检查约束(Check Constraint)、默认约束(Default Constraint)和外部环境键约束(Foreign Key Constraint)。使用约束优先于使用触发器、规则和默认值。查询优化器也使用约束来生成高性能的查询执行计划。

1. 主键约束

主键约束(Primary Key Constraint)通过在表中设置主键值来体现，它是惟一确定表示表中每一条记录的标识符，也是最重要的一种约束。在一个表中，不能有两行包含相同的主键值。一个表可以有多个能惟一标识行的组合，每个组合就是一个候选键。数据库管理员从候选键中选择一个作为主键。设置为主键的列值不允许为空(NULL)。

企业管理器可以方便地完成创建、修改和删除主的任务。

(1) 创建主键

在企业管理器中选中表 Students,进入“设计表”窗口(如图 3.13 所示)，选定要作为主键的字段(如果有多个字段要组合作为主键，则按住 Ctrl 键，用鼠标选取)，然后单击鼠标右键，在弹出的菜单中选择“设置主键”项，或者单击工具栏上的图标即可。

(2) 修改和删除主键

修改和删除主键的方法是通过表的属性窗口(见图 3.14)：在该属性窗口中选择“索引/键”选项卡(见图 3.16)，在“选定的索引”下拉列表中选择主键的名称(以 PK 开头)，单击“删除”按钮，则取消对当前字段“主键”的设定。

在前面创建表 Students 时，我们已经了设置 Sno 为主键，从而为表 Students 建立了主键约束。

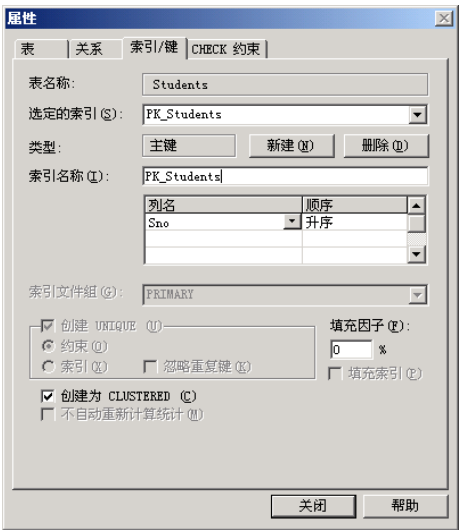


图 3.16 创建、修改和删除主键约束

2. 惟一性约束

惟一性约束(Unique Constraint)用于指定个或者多个列的组合值具有惟一性，以防止在列中输入重复的值。如果需要考虑除主键以外的字段的惟一性时，就可以使用惟一性约束。对于主键，则强制执惟一性约束。使用惟一约束的字段可以为空值，一个表中可以有多个惟一性约束。

用企业管理器创建、修改和删除惟一性约束与主键设置方法相仿。我们以为表 Students 的 Sname 字段添加惟一性约束为例来说明建立惟一性约束的方法：


- 1) 在企业管理器中选中表 Students，进入“设计表”窗口(见图 3.13)。
- 2) 可以在设计表窗口的工具栏中选择  图标，或者在图 3.13 的右键菜单上选择“属性”项或者“索引/键”选项卡(见图 3.17)。
- 3) 在“索引/键”选项卡中，单击“新建”按钮，在列名先栏中选择地字段名“Sname”，选中复选框“创建 UNOQUE”，并同时选中“约束”选项(见图 3.17)。



图 3.18 创建、修改和删除检查约束

我们为表 `Sstudents` 和 `Ssex`, 添加一个检查约束, 用来检查学生的性别输入值是男性('M')或者女性('F'):

- 1) 在企业管理器中 `Students`, 进入“设计表”窗口(如图 3.13 所示)。
 - 2) 可以在设计表窗口的工具栏中选择按钮, 也可以在图 3.13 的右键菜单上选择“属性”项或者“CHECK 约束”项, 都可以进入表属性窗口的“CHECK 约束”选项卡如图 3.18 所示。如果当前已经打开“属性”窗口, 则前两步可以略过。
 - 3) 在表“属性”窗口的“CHECK 约束”选项卡, 单击“新建”按钮, 可以通过编辑“约束名”来修改默认分配的约束名称“CK `Students`”, 输入约束表达式“`Ssex='F' or Ssex='M'`”如图 3.18 所示。
 - 4) 单击“关闭”按钮, 在退出“设计表”窗口前保存设置, 就可以使设置生效。
4. 默认约束

默认约束(Default Constraint)指在向表中做插入时, 如果没有提供字段的数据值, 则系统自动指定值。默认约束可以包括常量、函数、不带变量的内部函数或者空值(NULL)。

使用默认约束时, 应该注意:

- 1) 每个字段只能定义一个默认约束

- 2) 如果定义的默认约束值大于字段的设定长度，则默认值被按最大长度截断。
- 3) 不能为带有标识列 (IDENTITY) 属性或者类型为 timestamp 的字段定义默认约束。

使用企业管理器创建默认约束的方法相当简单：

在企业管理器中选中表 **Srudents**，进入“设计表”窗口(如图 3.13 所示)，对应每个字段都有一个默认值的设定栏，可以对该栏进行编辑。在这里，我们为表 **Students** 的 **Sdept** 字段添加默认约束(如图 3.19 所示)：在 **Sdept** 应的默认值栏中输入“Computer”，表明在没有值输入的情况下，**Sdept** 的默认值为‘Computer’。

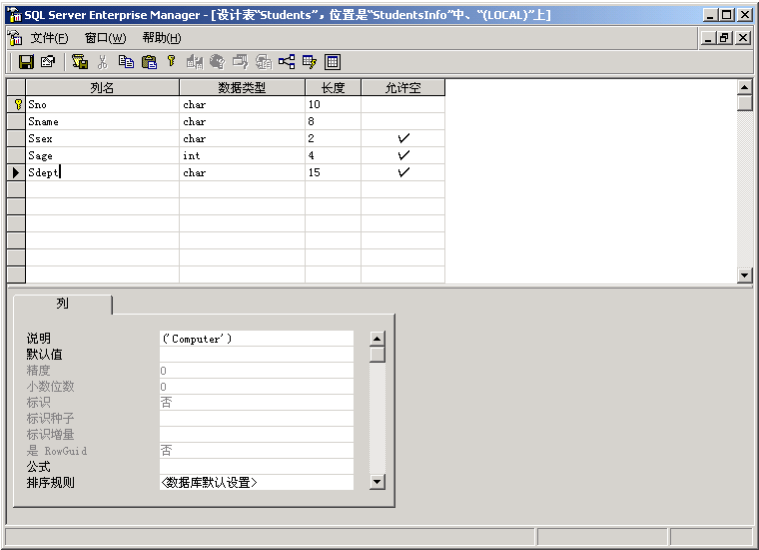


图 3.19 创建、修改和删除默认约束

5. 外部键约束

外部键约束 (Foreign Key Constraint) 用于强制参照完整性，提供单个字段或者多个字段的参照完整性。定义时，该约束参考同一个表或者另外一个表中主键约束字段或者惟一性约束字段。

- 1) 一个表中最多可以有 31 个外部键约束，临时表中不能使用外部键约束。
- 2) 主键和外部键的数据类型必须严格匹配。

我们再来新建表 **Courses** 和 **Enrollment** 来说明外部键的创建方法，这两个表的结构分别如表 3.2 和表 3.3 所示。

表 3.2 Courses 表结构

字段名	含义	数据类型	是否允许为空	备注
Cno	课程号	CHAR(6)	不允许	主键
Cname	课程名称	CHAR(20)	不允许	
PreCno	先修课号	CHAR(6)	允许	
Credits	学分	INT	允许	

表 3.3 Enrollment 表结构

字段名	含义	数据类型	是否允许为空	备注
Sno	学号	CHAR(10)	不允许	外键，参照 Students 的主键
Sname	课程号	CHAR(6)	不允许	外键，参照 Courses 的主键
Grade	年级	INT	允许	
主键为: (Sno, Cno)				

这样，在数据库 StudentsInfo 中就有三个表：Students(学生信息表)、Courses(课程信息表)和 Enrollment(选课表)，这三个表的内容见表 3.4、表 3.5 和表 3.6 所示。

表 3.4 Student 表数据

Sno	Sname	Ssex	Sage	Sdept
20010101	Hone	M	19	Computer
20010102	Sue	F	20	Computer
20010103	Smith	M	19	Math
20010104	Allen	M	18	Automation
20010105	Deepa	F	21	Art

表 3.5 Courses 表数据

Cno	Cname	ProCno	Credits
C1	English		4
C2	Math	C1	2
C3	Datase	C2	2

表 3.6 Enrollment 表数据

Sno	Cno	Grade
20010101	C1	90
20010102	C1	88
20010102	C2	94
20010102	C3	62


在企业管理器中有两种常用的创建外部键约束的方法：

1) 使用表属性窗口的“关系”选项卡(见图 3.20)。这种创建外部键约束的方法以与上述的约束创建方法以完全类似。

单击“新建”按钮创建外部键约束，选择主键所在的表和字段名、外部键所在的表和字段名即可。也可单击“删除”来删除对应的约束设置。

2) 使用关系图。

① 在企业管理器中打开 StudentsInfo 数据库文件夹，单击“关系图”选项。单击鼠标右键，在弹出的菜单上选择“新建关系图”，则出现“新建关系图”窗口和“欢迎使用创建数据库关系图向导”对话框(如图 3.21 所示)，使用这个向导可以帮助我们建立数据库关系图。也可以使用手动的方式，则单击向导窗口的“取消”按钮。

②在新建关系图上单击后在弹出的菜单上选取“添加表”或者使用工具栏上的  图标，添加表 Students、Courses 和 Enrollment，如图 3.22 所示。

③在这个图上单击 Courses 表的主键字段 Cno，并在鼠标左键按下的同时拖动到需要建立连接的外部键字段 Enrollment 表中的 Cno 字段上，然后释放鼠标左键，系统出现“创建关系”对话框，选择“确定”按钮即可，如图 3.23 所示。按照相同的方法可以为表 Enrollment 中的 Sno 字段建立外部键约束。建立完毕后的关系图如图 3.24 所示。

④单击保存这个关系图。

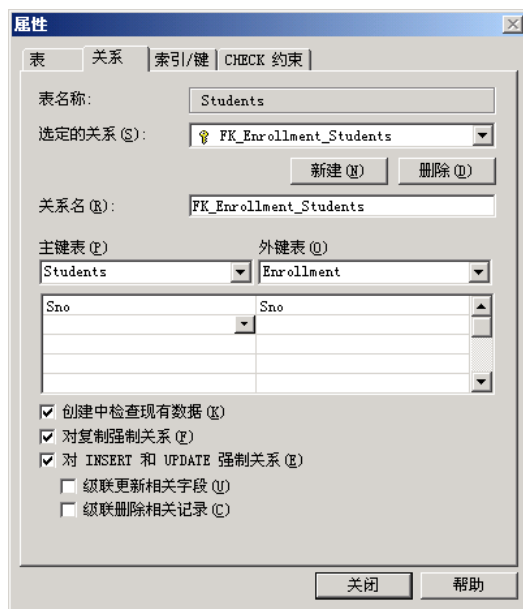


图 3.20 创建新关系图

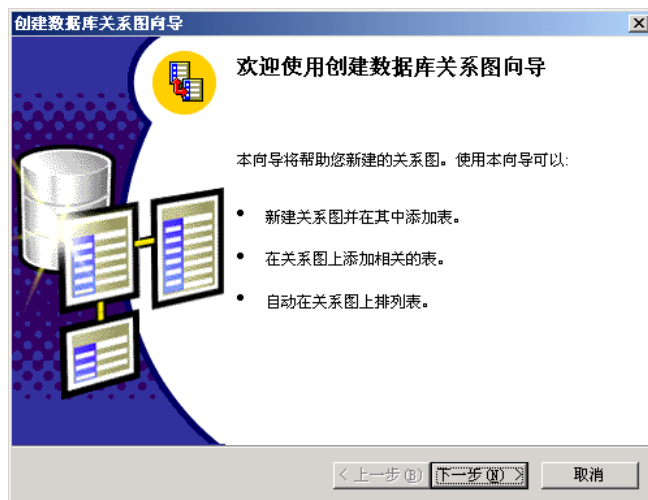


图 3.21 创建新关系向导图.

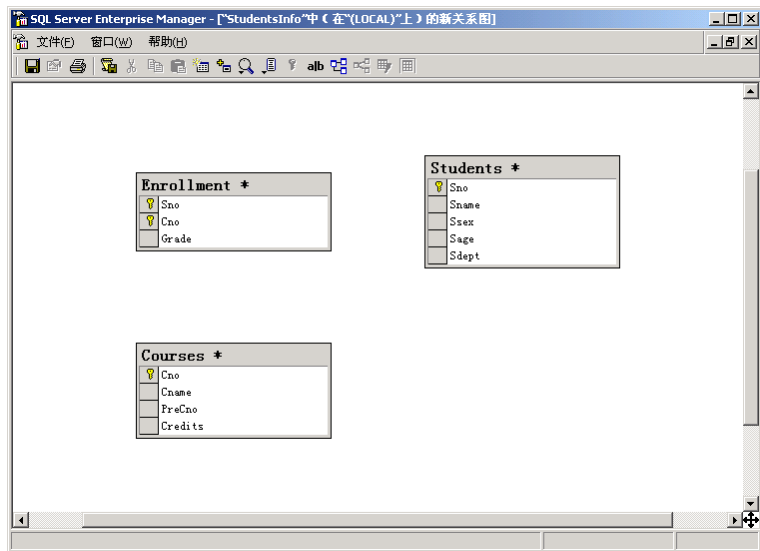


图 3.22 为关系图添加表

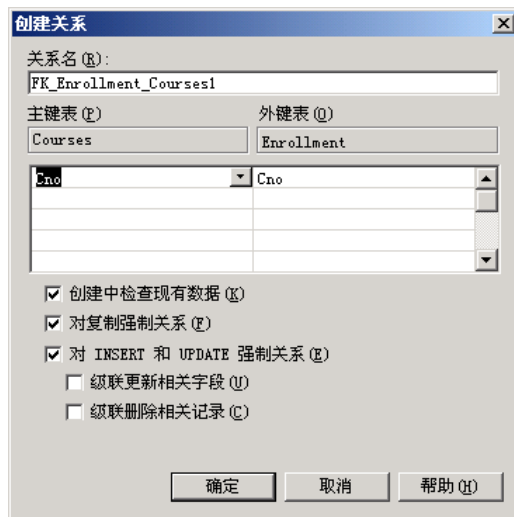


图 3.23 外部键约束建立

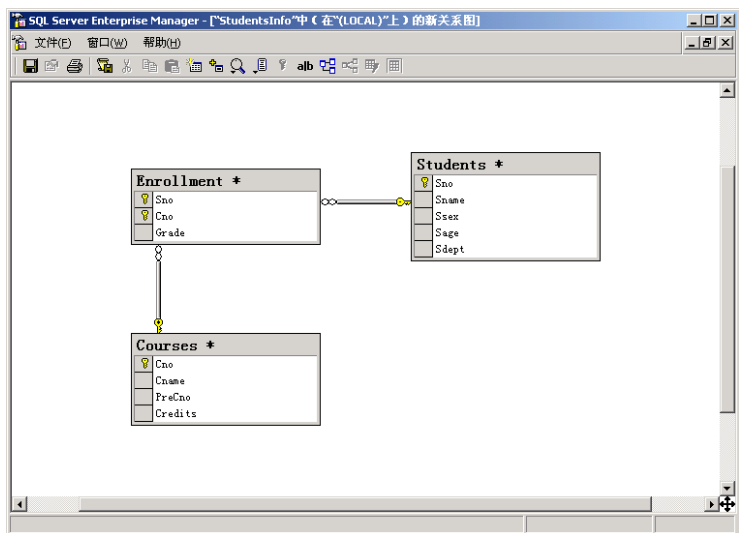


图 3.24 创建好的新关系图

五、输入、修改、删除数据

在表创建完毕后，就可以向表中输入数据以供使用。企业管理器提供了直观的方法来对数据库中表的数据进行输入、修改和删除。

在企业管理器中选择要操作的表，单击鼠标右键，然后在弹出的菜单中选择“打开表”项。其中有 3 个查看表数据的子项(见图 3.25)：

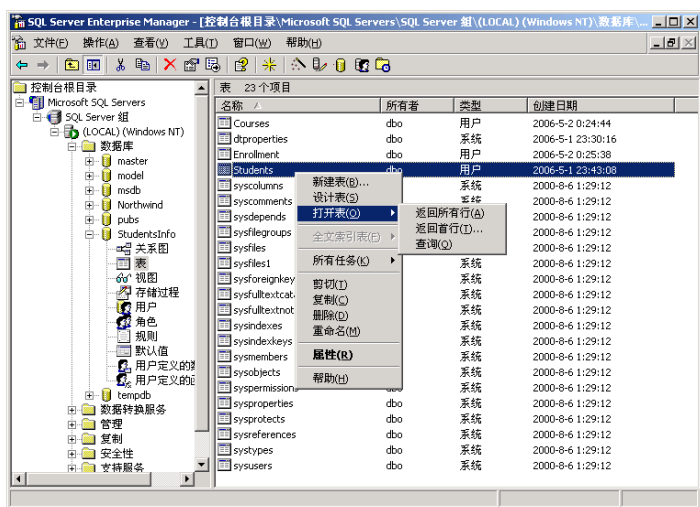

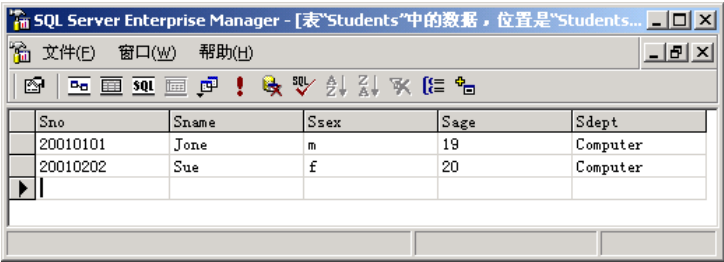


图 3.25 打开表

- 1) 返回所有行 用于显示所有表的数据；
- 2) 返回首行 用于显示前 N 条记录数据，N 可以自行指定；
- 3) 查询 用于查看某具体记录。

在此只对“返回所有行”。作详细说明。选择该选项后，系统显示表数据窗口，如图 3.26 所示。初始时，表 Students 为空，然后根据表的字段格式向表中输入数据，如“20010101’，‘Jone’，‘M’，19，‘Computer’”和“20010102’，‘Sue’，‘F’，20，‘Computer’”，单击工具栏上的“运行”图标，则系统检查数据的合法性成功后保存数据。修改数据的方法也相同：修改数据后单击“运行”图标即可。对于不要的数据，可以选中数据行后直接使用键盘上的 Del 键；或者单击鼠标右键，在弹出的菜单上选择“删除”项即可。



Sno	Sname	Ssex	Sage	Sdept
20010101	Jone	m	19	Computer
20010202	Sue	f	20	Computer

图 3.26 表数据的输入与修改

第四章 SQL Server2000 的查询分析器的使用

SQL 查询分析器是用于交互式设计和执行 Transact-SQL 语句、批处理和脚本的极好工具。用户若要使用 SQL 查询分析器，必须先了解和掌握 Transact-SQL 语句。在 SQL 查询分析器中，用户可在全文本窗口中编辑 Transact-SQL 语句，或者打开包含 Transact-SQL 语句的文本文件，执行该语句并在结果窗口中查看结果。查询分析器还在 Transact-SQL 语法中使用不同的颜色，以提高复杂语句易读性。

一、查询分析器的启动与关闭

(1) 查询分析器的启动

启动查询分析器有两种方法：

- 1) 和企业管理器一样，从系统的“开始”菜单上的 Microsoft SQL Server 程序组上选择“查询分析器”项进入。
- 2) 从企业管理器中进入。在企业管理器的菜单“工具”中，选择“SQL 查询分析器”项。如果预先没有和数据库服务器建立连接，则需要合法登录后才能使用查询分析器。使用查询分析器中的“文件”→“连接”(如图 4.1 所示)，然后出现“连接到 SQL Server”的登录窗口(如图 4.2 所示)，登录验证通过后，进入查询分析器窗口如图 4.3 所示。

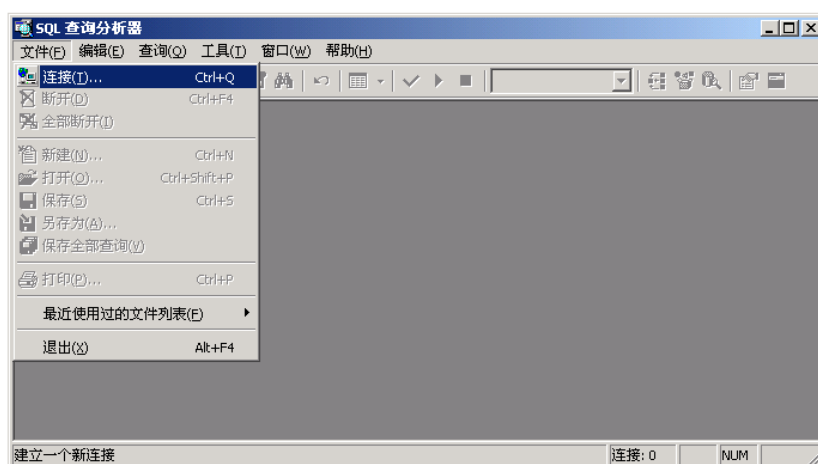


图 4.1 与 SQL Server 建立连接

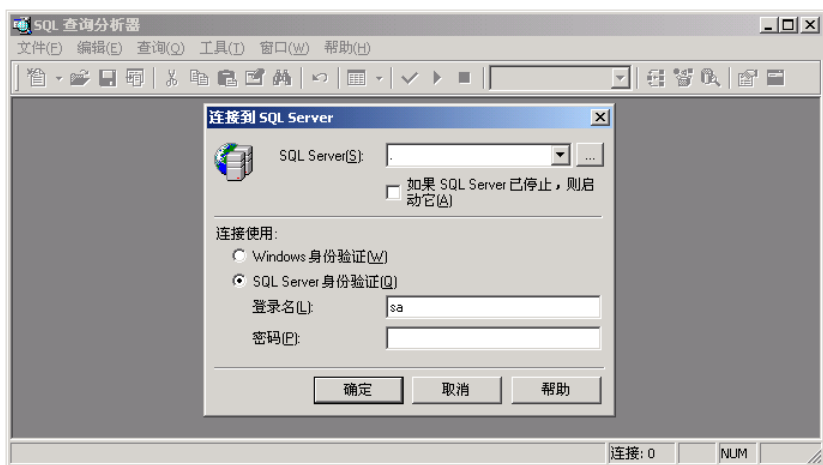


图 4.2 连接到 SQL Server 窗口

(2) 查询分析器的关闭

查询分析器的关闭方法与企业管理器相同，有两种：

- 1) 单击窗口的标题栏最右侧的控制图标。
- 2) 单击查询分析器器的菜单栏中的“文件”菜单，然后单击“退出”项。

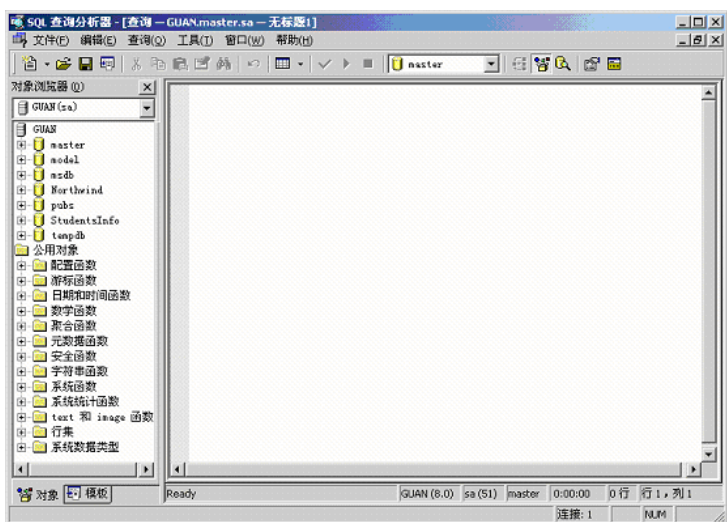


图 4.3 SQL Server 查询分析器界面

二、执行 SQL 语句

1) 设置 Transact-SQL 语句所要操作的数据库。这个设置可以通过对工具栏上的数据库下拉列表来选择，也可以使用菜单，“查询”→“更改数据库”来设置，如图 4.4 所示。将数据库更改为我们当前要操作的“StudentsInfo”

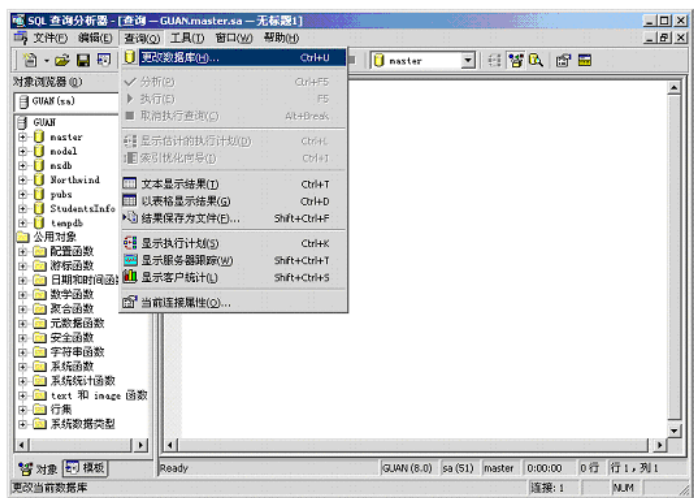



图 4.4 目标数据库的设置

2) 在 SQL 的查询分析器的文本编辑器编辑窗口中直接输入要执行的 SQL 语句(如图 4.5 所示)，查询学生信息表(Students)中的所有记录。语句输入完成后，可以选择“查询”→“分析”命令或者使用工具栏上的  图标来检查语法是否正确。如在输入“select*from student”的执行语法检查，提示在第一行的问号附近有语法错误。根据提示修改 SQL 语句，直到提示“命令已成功完成。”则表示当前语句通过语法验证。

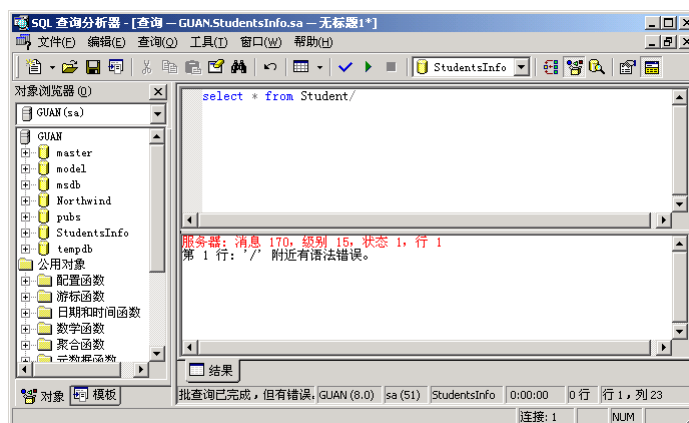



图 4.5 SQL 语句的语法检查

3) 选择菜单“查询”→“执行”命令或者单击工具栏上的  图标, 来执行 SQL 语句。如果执行成功, 则系统返回查询结果; 否则, 给出执行失败的反馈消息, 并提示命令不能成功执行的原因: 数据库中并没有被称为 Student 的表对象(如图 4.6 所示)。用户可以根据提示反复修改 SQL 语句, 直到执行成功为止(如图 4.7 所示, 将 Student 改为 Students 后执行成功)。

注意: 在执行 SQL 语句时, 也要先检查语法。

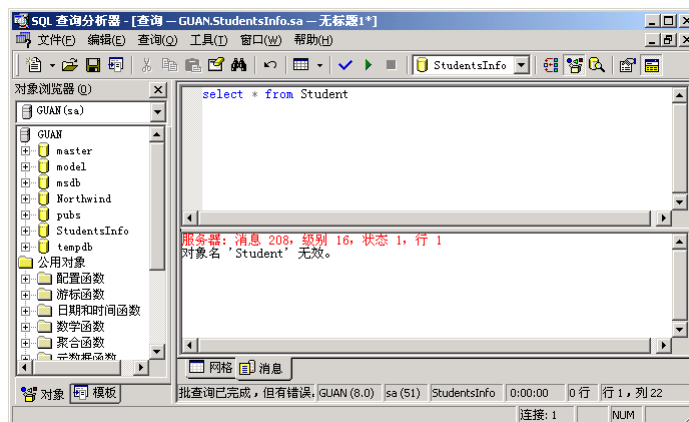


图 4.6 SQL 语句执行失败

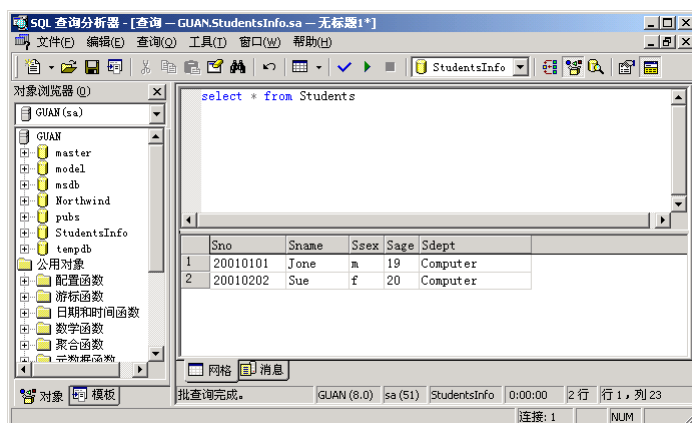


图 4.7 SQL 语句执行成功

三、查询结果的显示模式

SQL 查询分析器提供了 3 种查询结果的方法：以文本显示结果、以表的形式显示结果和将结果保存为文件。在图 4.7 中，SQL 查询的结果是以表的形式显示的。结果显示方式的设置结果的显示模式后，再重新执行 SQL 语句，则结果就按照设置的显示模式显示。

其他两种模式的结果分别如图 4.8 和图 4.9 所示。

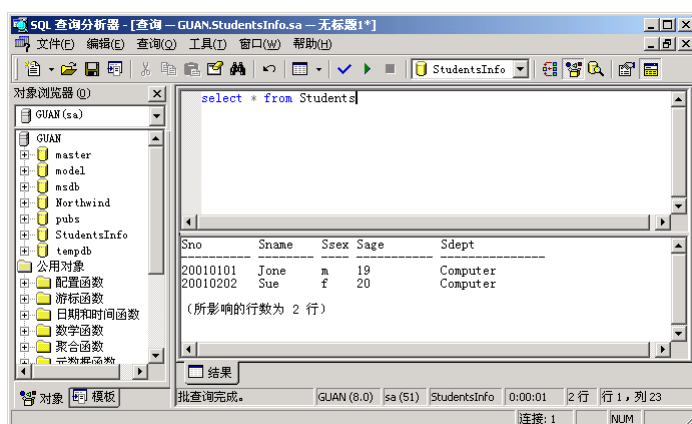


图 4.8 “以文本显示结果”模式

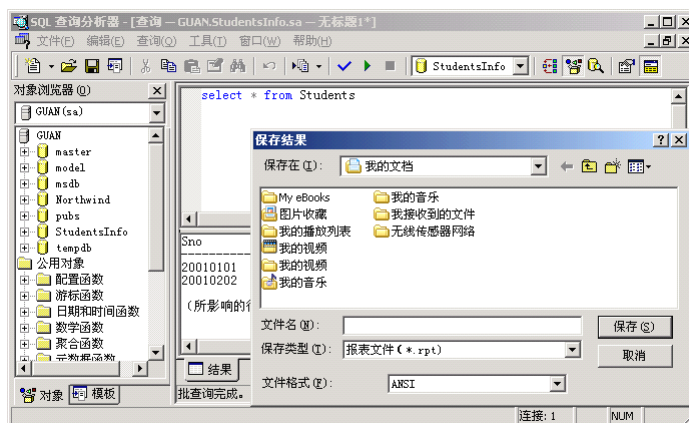


图 4.9 “将结果保存为文件”模式

四、创建和运行脚本文件

一个脚本文件包含一个或多个批处理，每个批处理由一条或多条 Transact-SQL 语句组成，每个批处理之间用“GO”分割。GO 不是 Transact-SQL 语句，而是可被 SQL Server 查询分析器、osql、isql 等与 SQL Server2000 有关的实用工具识别的命令。SQL Server 实用工具将 GO 解释为：将当前 Transact-SQL 批处理语句发送给 SQL Server 服务器，其中当前批处理语句是自上一条 GO 命令后输入的所有语句。因此，GO 意味着一个批处理的结束。

例如，下面是一个脚本文件，包含 3 个批处理：第一个批处理在第一个“GO”之前的语句，用于判断 StudentsInfo 数据库中是否存在表对象 Students，如果存在，则删除它；第二个批处理是第一个“GO”之后到第二个“GO”之前的语句，用于创建表 Students；第三个批处理是第二个“GO”之前的语句，用于查询已创建的表对象名称。

```
USE StudentsInfo
IF object_ID('dbo.students') IS NOT NULL
DROP TABLE dbo.students
GO
CREATE TABLE Students(
Sno      CHAR(10),
Sname    CHAR(8)   NOT NULL,
Sage     CHAR(1)   NOT NULL,
```

```

Sage    INT,
Sdept   CHAR(20)          DEFAULT    'Computer',
CONSTRAIN SPK PRIMARY    KEY(Sno),
CONSTRAIN CK             CHECK(Ssex='F' OR Ssex='M')
)
GO
--Display results.
SELECT table_name
FROM information_schema.tables
WHERE table_name='Students'
GO

```

脚本文件可以在 SQL Server2000 的查询分析器中直接运行，并得到结果。因此，对于开发数据库应用的程序员来说，脚本文件是高效、低价的前期准备 SQL 语句的方法，他们可以先编写简单的脚本文件来验证 SQL 语句的正确性，然后再将这些 SQL 语句插入到其他高级语言书写的程序中。脚本文件也为重复创建数据库及其对象提供了方便，因为脚本文件可以一次存储，重复执行。SQL Server2000 的企业管理器还可以为已有的数据库对象自动生成脚本文件。以便以后重复利用。SQL Server2000 的查询分析器可以创建、打开、保存并执行脚本文件，脚本文件的扩展名为.sql。

(1) 创建脚本文件

可使用 SQL 查询分析器的菜单“文件”→“新建”或者单击工具栏图标，来创建脚本文件。查询分析器打开一个新建查询的对话框，如图 4.10 所示。可以通过它打开一个空查询窗，或者使用系统提供的 SQL 模板。在此，使用新建一个新查询窗口，如图 4.11 所示。

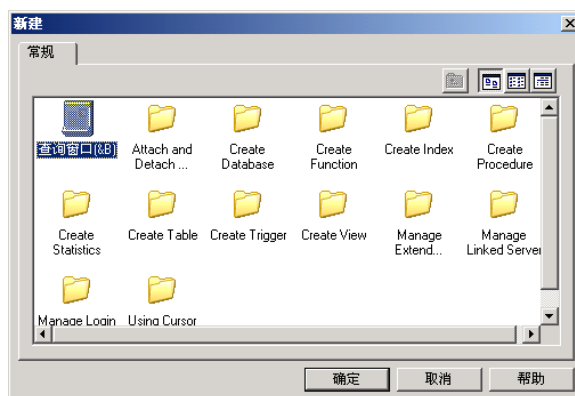


图 4.10 新建查询对话框

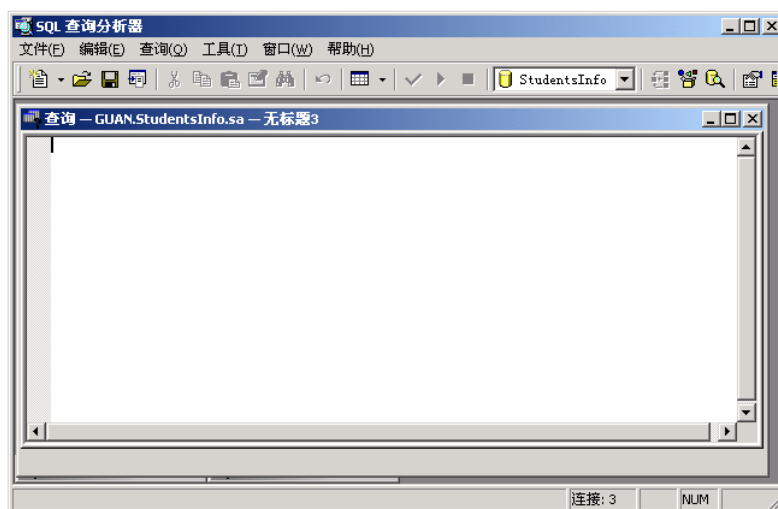



图 4.11 打开一个空白查询窗口

在空白查询入“select * from students where Sdept='Computer'”，检查语法正确后选取菜单“文件”→“保存”命令，或使用工具栏中的“保存”图标，将当前脚本存为 students_jsj.sql。这样就创建一个 SQL 的脚本文件，如图 4.12 所示。

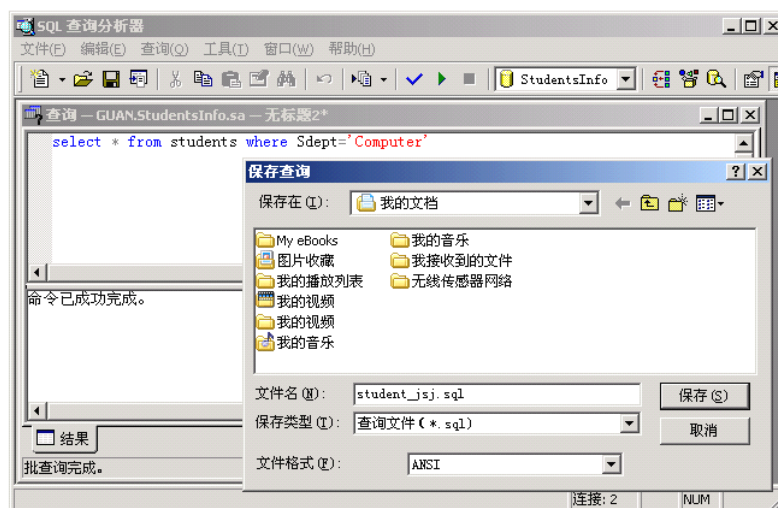



图 4.12 保存 SQL 脚本文件

(2) 执行已创建的脚本文件

选择查询分析器的菜单“文件”→“打开”命令或者使用工具栏上的  图标，可以打开已经创建好的 SQL 脚本文件，这里打开刚刚创建的脚本 students_jsj.sql 如图 4.13 所示。然后，就使用前面介绍的 SQL 语句的方法来执行这个脚本。自行选择结果集合的显示模式。这个查询返回所有系别为“Computer”(计算机系)的学生记录，如图，如图 4.14 所示。

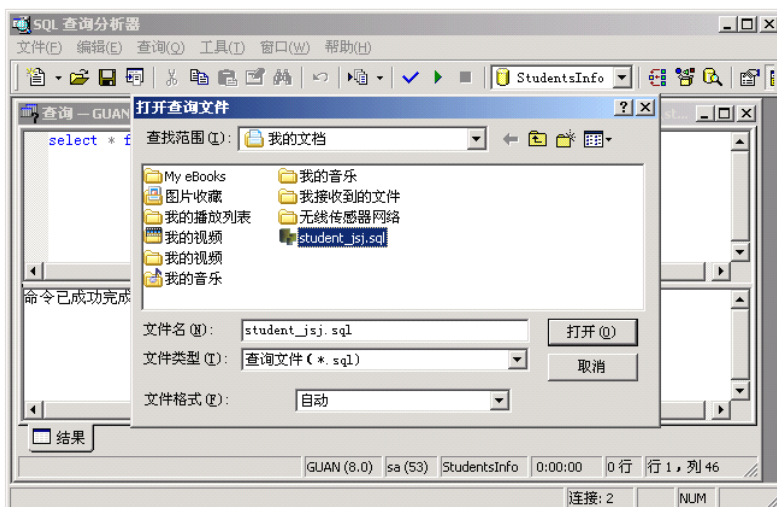


图 4.13 打开 SQL 脚本文件

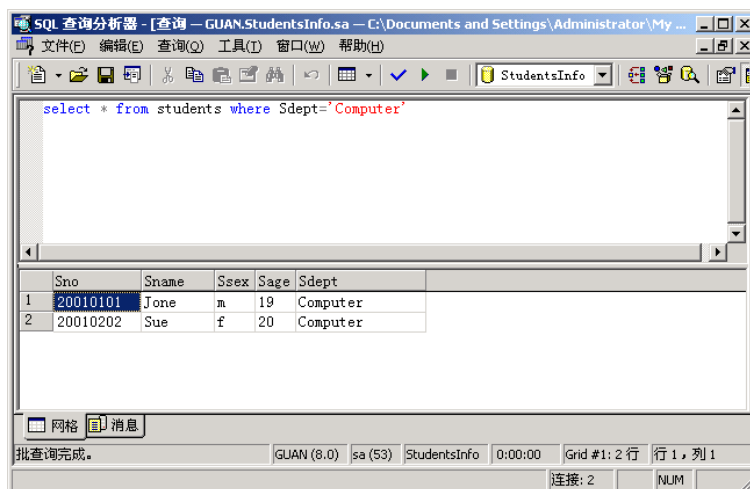


图 4.14 显示运行结果

五、查询分析器的常用快捷键

SQL 查询分析器对一些常用功能提供了键盘快捷使用方式，让熟练的用户避免频繁的菜单单击。表 4-5 列出了 SQL Server 查询分析器最常用的键盘快捷方式。

表 2-4 查询分析器中的最常用的快捷键

功能	快捷方式
编辑：复制	Ctrl+C、Ctrl+Insert
编辑：剪切	Ctrl+X、Shift+Del
编辑：粘贴	Ctrl+V、Shift+Insert
编辑：增大缩进	Tab
编辑：减小缩进	Shift+Tab
编辑：在编辑器窗格中删除至行尾	Ctrl+Del
编辑：查找	Ctrl+F
编辑：转到行号	Ctrl+G
编辑：重复上次搜索或查找下一个	F3 功能键
编辑：替换	Ctrl+H
编辑：全选	Ctrl+A
编辑：撤销	Ctrl+Z
执行查询	F5 功能键
SQL 查询分析器帮助	F1 功能键
对所选 Transact-SQL 语句的帮助	Shift+F1

第五章 SQL Server 2000 用户与权限管理

SQL Server 2000 也是通过设置不同级别的用户和分配不同的权限的方法来实现数据库的安全性的。一个用户如果要访问 SQL Server 数据库中的数据，必须经过 3 个认证过程：

- **安全身份验证** 用来确认登录用户的帐号和密码的正确性，由此来验证用户是否具有连接到 SQL Server 数据库服务器的资格，这里只验证该用户是否具有连接到数据库服务器的“连接权”。SQL Server 2000 具有用户登录帐号统一管理的功能。
- **用户帐号验证** 当用户通过安全身份验证后，即登录到数据库服务器后，若要访问具体的某个数据库，必须拥有对该数据库访问的用户帐号，才能通过用户帐号验证。SQL Server 2000 具有对数据库的用户帐号进行统一管理的功能。
- **操作许可验证** 当用户通过上述两级验证后，若要操作数据库中的数据或对象，还必须拥有相应操作的操作许可权，即必须通过操作许可验证 SQL Server 2000 管理数据库的用户帐号的操作权限。

1. 安全身份验证

SQL Server 提供了一种被称为安全身份验证的机制来确认登录用户的登录帐号和密码的正确性，由此来验证该用户是否具有连接 SQL Server 的权限。SQL Server 2000 提供了两种确认用户的验证模式，即 Windows 验证模式和混合验证模式。

(1) Windows 验证模式

Windows 验证模式利用了 Windows 本身具备的管理登录、验证用户合法性的能力，允许 SQL Server 用户登录时使用 Windows 的用户名和口令。在这种模式下，用户只需要通过 Windows 的验证，就可以连接到 SQL Server。此时，系统从用户登录到 Windows 时提供的用户名和密码中查找当前用户的登录信息，从而判断当前用户是否为合法用户。

在 SQL Server 中，推荐使用 Windows 验证模式。因为该模式能够与 Windows 操作系统的安全系统集成在一起，从而提供更多的安全功能。但是这种安全模式只能用在基于 NT 的 Windows 的操作系统上，在 Windows98 等个人操作系统上，不能使用 Windows 验证模式进行身份验证。

(2) 混合验证模式

混合验证模式接受 Windows 授权用户和 SQL 授权用户。如果不是 Windows 操作系统的用户也希望使用 SQL Server，那么应该选择混合验证模式。在该验证模式下，用户连接 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，与操作系统的登录帐号无关。SQL Server 自行执行认证处理，如果输入的登录信息与系统表 syslogins 中的记录相匹配，则登录成功。

(3) 利用企业管理器设置验证模式

用户可以根据自己的需要设置身份验证模式。在企业管理器中的设置方法如下：

- 1) 打开企业管理器，在要设置验证模式的服务器名上单击鼠标右键，然后在弹出的菜单上选择“属性”选项(见图 5-1)，则出现“SQL Server 属性”对话框，如图 5.2 所示。

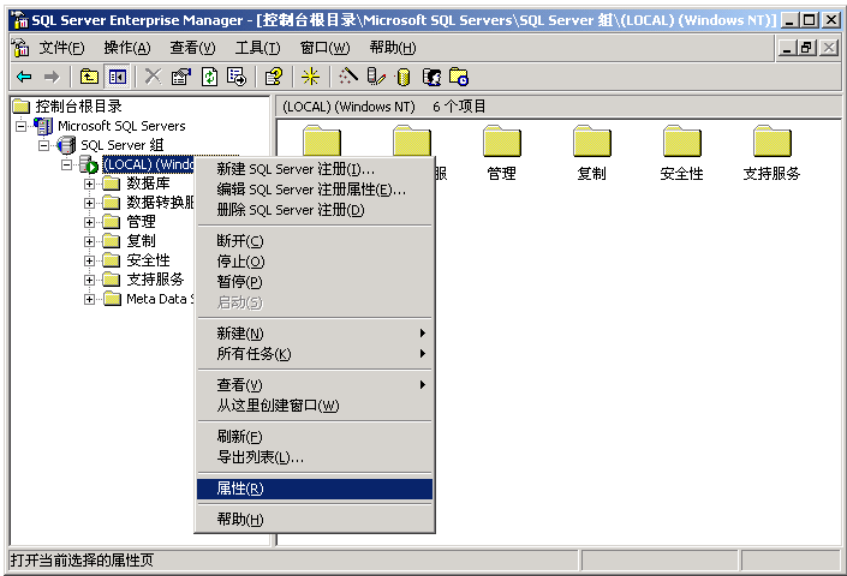


图 5.1 设置 SQL Server 安全身份验证

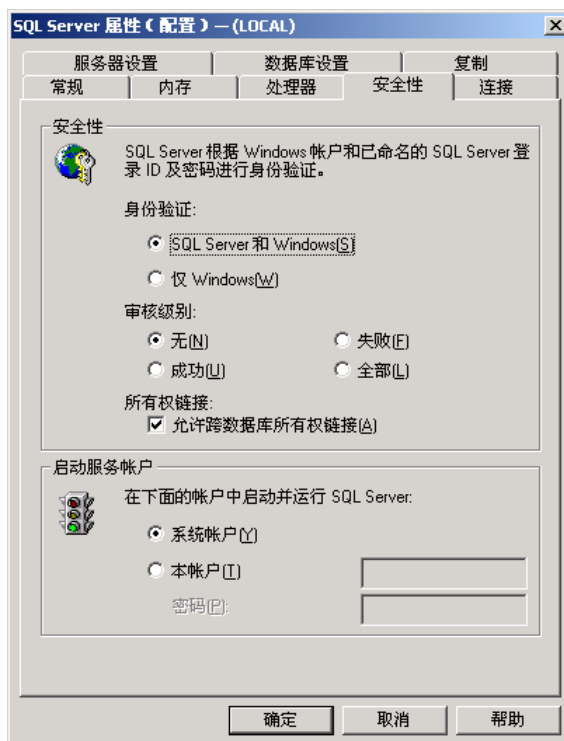


图 5.2 SQL Server 安全身份验证属性

- 2) 选择“安全性”选项卡(图 5.2)，在窗口的“安全性”选区中的“身份验证”部分有两个选项：“SQL Server 和 Windows”和“仅 Windows”。前一个选项代表混合验证模式，后一个选项代表 Windows 验证模式。用户可以选择需要的验证模式。
- 3) 单击“确定”按钮，保存设置。此时，系统会出现一个提示对话框，表示 SQL Server 在使服务器设置生效之前，必须先停止当前的所有服务，然后在更改完服务器设置后，再根据新的设置重新启动服务。因此，应该在没有用户使用服务器时修改服务器的设置。

要注意的是，用户通过了身份验证并不代表就有权访问 SQL Server 中的数据，同时他还必须通过权限确认。用户只有在具备访问数据库的权限后，才能对数据库服务器上的数据库进行权限许可范围内的各处操作，这种用户访问数据库权限的设置是通过用户帐号来实现的。

2. 登录帐号管理

一个用户必须具有有效的登录帐号，才能登录数据库服务器。SQL Server 有两个默认的登录帐号：sa、BUILTIN/administrators。其中，sa 是系统管理

员(System Administrator)的缩写,是一个特殊的登录帐号,它在数据库中对应用户帐号 dbo, 在 SQL Server 系统和所有数据库中拥有所有的权限。BUILTIN/administrators 是 SQL Server 为每一个 Windows NT 系统管理员提供的默认的登录帐号,这个帐号是 SQL Server 系统和所有的数据库中也拥有所有的权限。

登录帐号可以使用 SQL Server 提供的系统存储过程来创建,也可以通过企业管理器来创建。下面只介绍使用企业管理器创建 SQL Server 用户登录帐号的方法:

- 1) 打开企业管理器,打开需要登录的服务器的文件夹,然后再展开“安全性”文件夹。选择“登录”图标,单击鼠标右键,在弹出菜单上的选择“新建登录”项,如图 5.3 所示。

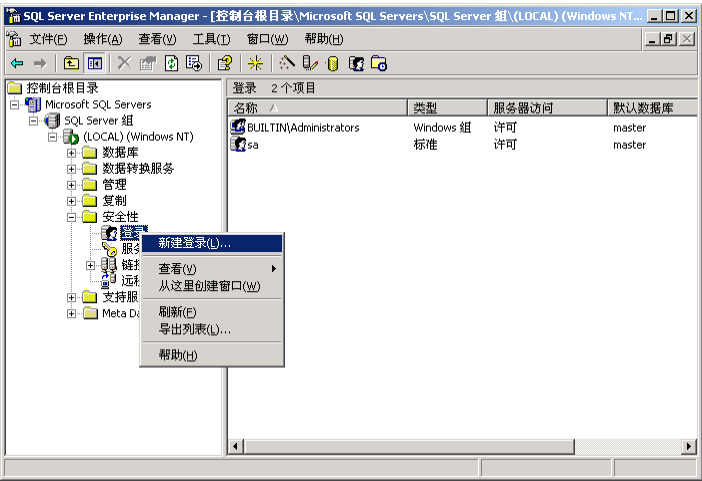


图 5.3 创建 SQL Server 登录帐号

- 2) 在“SQL Server 登录属性-新建登录”对话框(如图 5.4 所示)中进行设置。

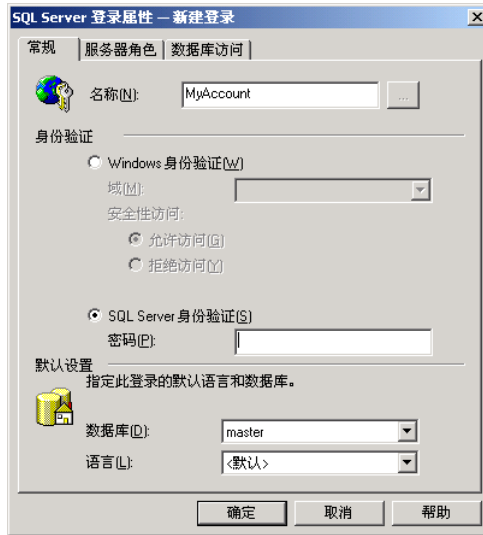


图 5.4 “常规”选项卡

- 在“常规”选项卡中，输入登录名，如 My-Account。在“身份验证”选区中选择需要的验证模式：SQL Server 身份验证。如果使用“SQL Server 身份验证”，那么要输入密码，如 My-Account。在“默认设置”选区可以设置与当前登录帐号对应的默认的数据库和语言。
- 在“服务器角色”选项卡中(如图 5.5 所示)，列出了系统的固定服务器角色。在左侧的复选框上打勾表示当前登录帐号是相应的服务器角色成员，例如选中 System Administrator、Security Administrators 和 Server Administrators。

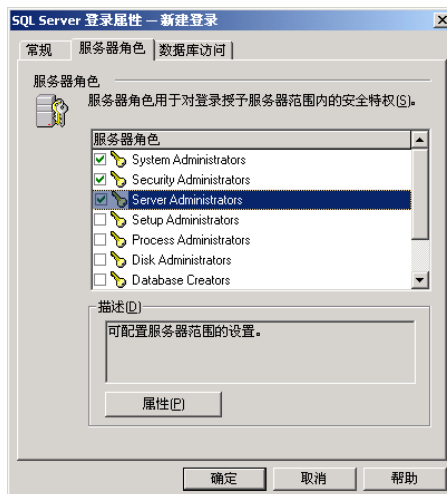


图 5.5 “服务器角色”选项卡

- 在“数据库访问”选项卡中(如图 5.6 所示),列出了当前登录帐号可以选择访问的数据库。在左侧的复选框上打勾,表示当前登录帐号可以访问对应的数据库,默认创建的访问该数据库的用户帐号与登录帐号相同,也可以输入另外的名称,例如选中 **StudentsInfo**。下部的列表框列出了相应的数据库中的数据库角色清单,从中可以指定该登录帐号关联的用户帐号所属的数据库角色。



图 5.6 “数据库访问”选项卡

- 单击“确定”按钮,完成创建。

对于已经建立好的 SQL Server 登录帐号,可以使用鼠标右键,在弹出的菜单上选择“属性”进行修改(见图 5.7)。若不再需要某个登录帐号,或者不再允许某个登录帐号访问 SQL Server,则可以将其删除,在右键弹出的菜单上选“删除”项或者直接按键盘上的 Delete 键。

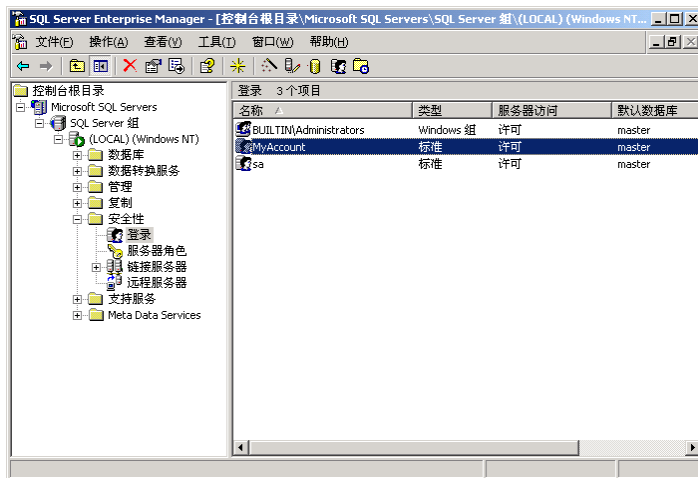


图 5.7 修改和删除登录帐号

3. 用户帐号管理

在数据库中，用户或者工作组取得合法的登录帐号，只是表明该帐号通过了 Windows 验证或者 SQL Server 安全身份验证。通过登录帐号，用户只能连接到 SQL Server 服务器上，但不能对数据库数据和数据库对象进行某种或某些操作，只有当他同时拥有了用户帐号后，才能够访问数据库。用户帐号惟一标识用户，用户对数据库的访问权限以及与数据库对象的所有关系都有通过用户帐号来控制的。用户帐号总是基于数据库的，它只能来自于服务器上已有的登录帐号，两个不同的数据库可以有相同的用户帐号。

管理数据库用户的过程实际上就是建立登录帐号与数据库用户帐号之间的映射关系的过程。默认情况下，新建立的数据库只有一个用户帐号 dbo，它是数据库的拥有者，系统默认的登录帐号 sa 自动与每一个数据库用户帐号 dbo 关联。通常，数据库用户帐号总是与某一登录帐号相关联，但是 guest 用户帐号例外。guest 用户帐号在 SQL Server 安装时被添加到 master、pubs、tempdb 和 northwind 数据库中，让那些没有属于自己用户帐号的 SQL Server 登录帐号能够访问具有 guest 用户帐号的数据库(guest 用户帐号 master 和 tempdb 中必须始终存在，其他的数据库中可以添加或删除 guest 用户。默认情况下，新建的数据库中没有 guest 用户帐号)。

利用企业管理器可以授予 SQL Server 登录帐号访问数据库的权限。创建一个新的数据库用户帐号的过程如下：

- 1) 在企业管理器中选中要访问操作的数据库，在右击弹出的菜单上选择“新建数据库用户”选项(见图 5.8)，则出现“数据库用户属性-新建用户”对话框，如图 5.9 所示。

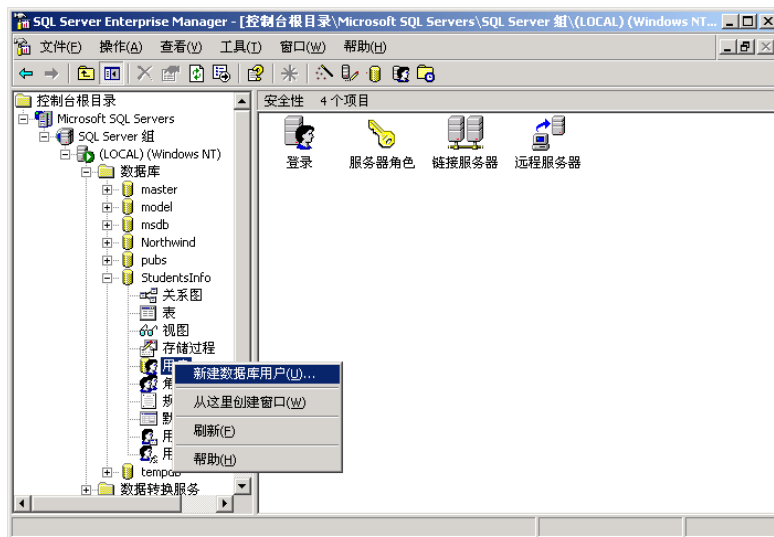


图 5.8 创建数据库用户帐户

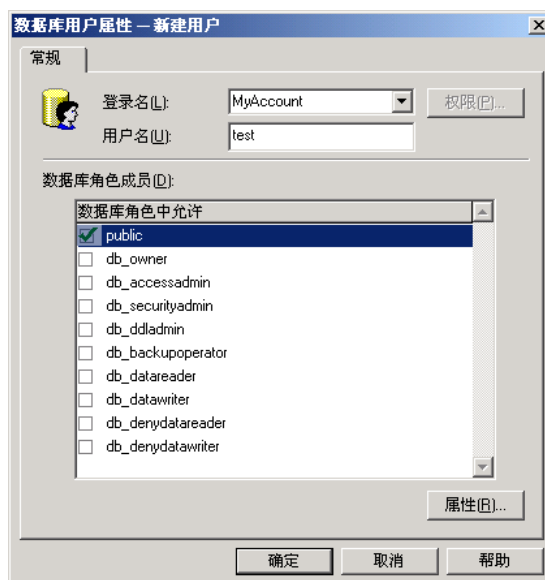


图 5.9 “新建用户”选项卡

- 2) 在“登录名”下拉列表中选择已经创建的登录帐号：My-Account，在“用户名”中输入数据库用户帐号名称，然后在数据库角色成员框中为当前用户帐号选择相关的角色。
- 3) 单击“确定”按钮，完成数据库用户帐号的创建。

要补充说明的是，在创建 SQL Server 登录帐号的时候，也可以同时创建用户帐号，即在图 5.6 的位置确定数据库的用户帐号。本例创建一个与登录

帐号 My-Account 对应的用户帐号：test。

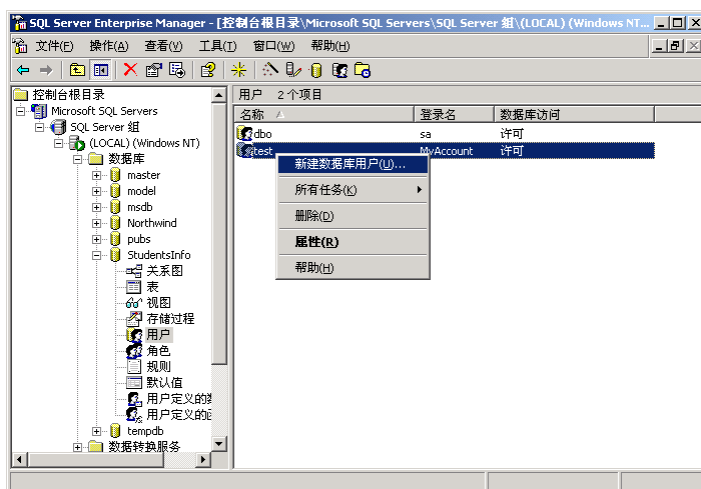


图 5.10 数据库用户的修改与删除

若要修改用户帐号，可以使用鼠标右键，在弹出的菜单上选择“属性”进行，如图 5.10 所示。若不再需要某个用户帐号，则可以将其删除，在右键弹出的菜单上选“删除”项或者直接按键盘上的 Delete 键。

4. 权限管理

权限用来限定用户可以使用的数据库对象和可以对数据库对象执行的操作。用户在登录到 SQL Server 以后，用户帐号所在的 Windows 用户组或角色所具有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、修改哪些数据库。在 SQL Server 中包括 3 种类型的权限，即对象权限、语句权限和隐含权限。

- 对象权限 是指用户对数据库中的表、视图等对象的操作权，相当于数据库操作语言(DML)的语句权限。如果用户想要对某一对象进行操作，必须具有相应操作的权限。表和视图权限用来控制用户是否有权在表和视图上执行 SELECT、INSERT、UPDATE 和 DELETE 操作。字段权限用来控制用户是否有权在单个字段上执行 SELECT、UPDATE 和 REFERENCES 操作。存储过程权限用来控制用户是否有权执行 EXECUTE 操作。
- 语句权限 相当于数据定义语言(DDL)的语句权限，表示对数据库的操作许可，通常是一些具有管理性的操作，如创建数据库、表和存储过程等。这种权限专指是否允许执行下列语句：CREATE TABLE、CREATE VIEW 等与创建数据库对象有关的操作。这种语句虽然仍包含有操作的对象，但这些对象在执行该语句之前并不存在于数据库中。因此，语句权限可针对某个 SQL 语句，而不是数据库中已经创建的特定的数据库对象。只有 sysadmin、db owner 和 db securityadmin 角色

的成员才能授予语句权限，语句权限可用的 Transact-SQL 语句及功能如表 5.1 所示。

- **隐含权限** 是指由 SQL Server 预定义的服务器角色、数据库角色、数据库拥有者和数据库对象拥有者所具有的权限。隐含权限相当于内置权限，在系统安装以后有些用户和角色不必经过授权就拥有的权限。其中，用户指数据库对象所有者，角色指固定服务器角色和固定数据库角色。只有数据库对象所有者或者固定角色的成员才能可以执行某些操作，执行这些操作的权力就成为隐含权限。

表 5.1 SQL Server 中语句权的 Transact-SQL 语句

序号	Transact-SQL	功能
1	CREATE DATABASE	创建数据库
2	CREATE TABLE	创建表
3	CREATE VIEW	创建视图
4	CREATE RULE	创建规则
5	CREATE DEFQULT	创建默认
6	CREATE PROCEDURE	创建存储过程
7	CREATE INDEX	创建索引
8	BACLUP DATABASE	备份数据库
9	BACKUP LOG	备份事务日志

因为隐含权限是 SQL Server 内置的，所以权限的设置实际上是指对象权限和语句权限的设置。

权限的管理包含如下 3 个内容：

- **授予权限** 允许用户或角色具有某种操作权。
- **收回权限** 不允许用户或角色具有某种操作权，或者收回曾经授予的权限。
- **拒绝权限** 拒绝某用户或角色具有某种操作权，即使用户或角色通过继承的方式得到了这种操作权，也不允许其执行相应的操作。

(1) 使用企业管理器管理权限

SQL Server 通过两种途径来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管理。这两种途径是：面向用户的权限设置和面向数据库对象的权限设置。

面向用户的权限设置的方法如下：

- 1) 在企业管理器中，展开指定的数据库文件夹，单击“用户”图标，则在右侧显示当前数据库的所有用户。可以在想要进行权限设置的用户上右击鼠标，然后在弹出的菜单上选择“属性”项，如图 5.11 所示。

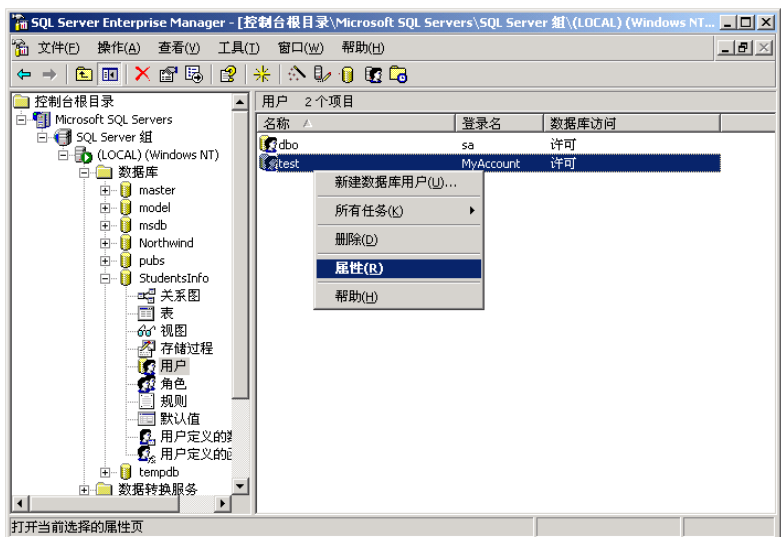


图 5.11 面向用户的权限设置

2) 系统出现与特定用户对应的“数据库用户属性”窗口(见图 5.12)。在窗口中说明了当前用户是那些数据库角色的成员。单击“权限”按钮，则弹出与数据库对应的“数据库用户属性”窗口(见图 5.13)。在该窗口中可以通过选中“列出所有对象”来列出当前数据库的所有对象，其中包括表、视图和存储过程等，同时也给出了针对该对象能够进行了的操作。通过单击其中空白方格来完成权限设置：当空白方格打勾时，表示执行了授权语句(**grant**)——授予权限，即用户能够对该对象执行此种操作；当空白方格打叉时，表示执行了否定语句(**deny**)——拒绝权限，即禁止用户对该对象执行相应的操作。从打勾变为空白，表示收回权限。如果选中“仅列出此用户具有的权限的对象”，则下面的列表框中将显示该用户有权访问的所有数据库对象以及详细的权限设置信息。这个步骤也可以通过在图 5.11 中选择“所有任务”→“管理权限...”达到。

在特定用户的“数据库用户属性”窗口(见图 5.12)，如果在“数据库角色成员”列表中选择任何一个数据库角色(在默认条件下，任何数据库用户都至少是 **public** 角色)，实际上就完成了数据库用户语句权限的设置。因为对于这些数据库固定的角色，SQL Server 已经定义了其具有那些语句权限。

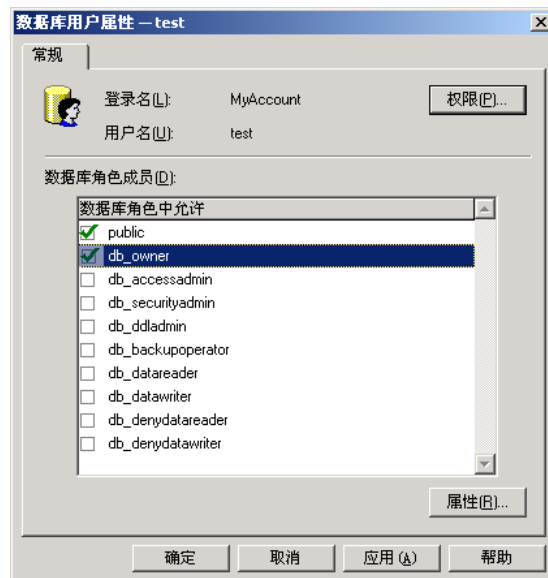


图 5.12 对用户 test 的权限设置

单击图 5.13 的“列”按钮，出现如图 5.14 所示的“列权限”窗口。在该对话框中可以设置用户对当前对象的哪些列具有的权限。

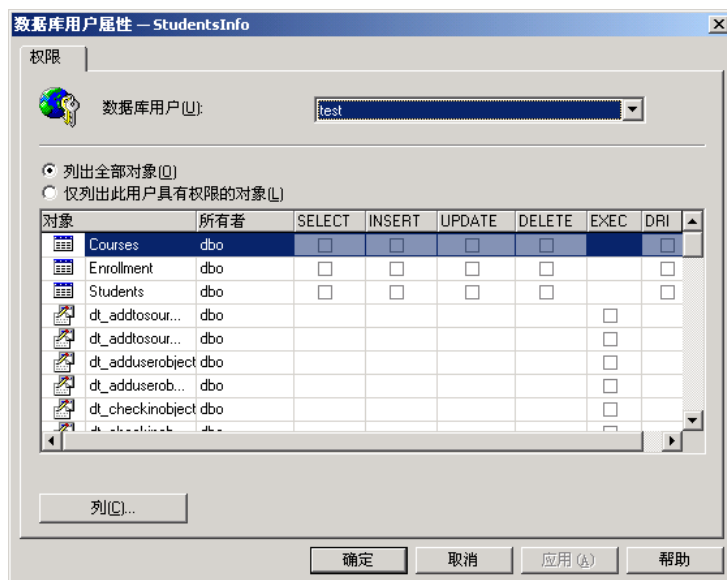


图 5.13 与数据库对应的数据库用户属性

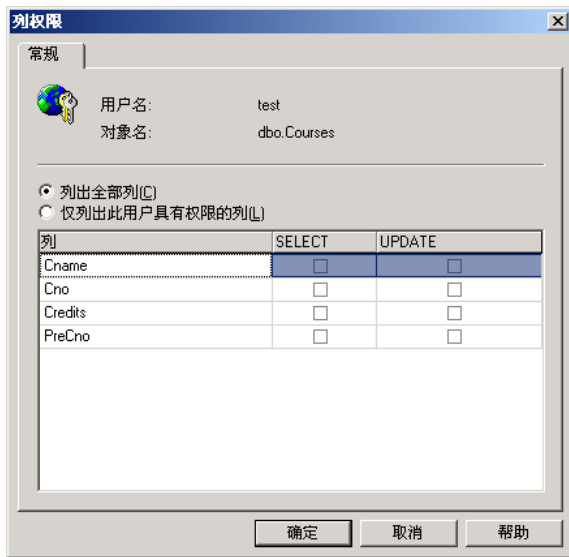


图 5.14 列权限

3) 单击“确定”按钮保存设置。

面向数据库对象的权限设置的方法如下：

1) 展开指定的服务器和指定的数据库，选择需要设置的数据库对象类型(即表、视图和存储过程等)，在右侧的列表中选择要进行了权限设置的具体对象。单击鼠标右键，从弹出菜单中选择“属性”项，如图 5.15 所示。

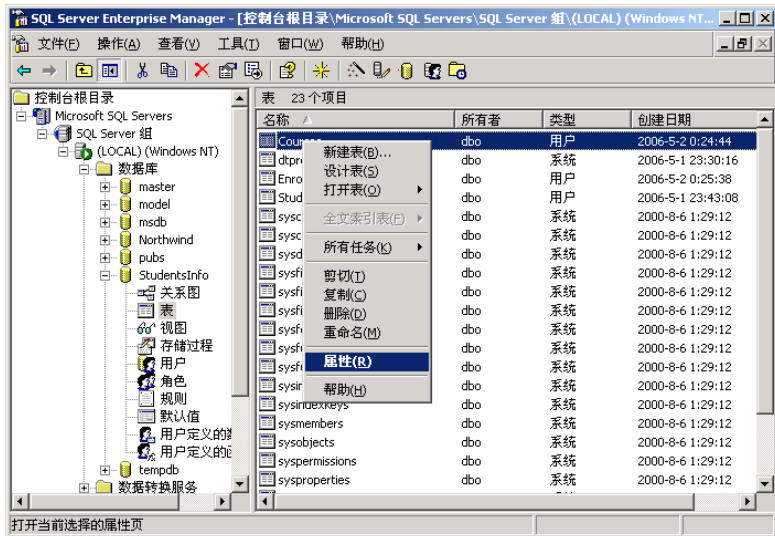


图 5.15 面向对象的权限设置

2) 系统出现与特定对象对应的属性窗口(见图 5.16)。单击“权限”按钮，

则弹出与对象对应的“对象属性”窗口(见图 5.17)。在该窗口中可以通过选中“列出全部用户/用户定义的数据库角色/public”来列出当前数据库的所有用户对该对象能够进行的操作的权限拥有情况。通过单击其中空白方格来完成权限设置：当空白方格打勾时，表示执行了授权语句(**grant**)——授予权限，即用户能够对该对象执行此种操作；当空白方格打叉时，表示执行了否定语句(**deny**)——拒绝权限，即禁止用户对该对象执行相应的操作。从打勾变为空白，表示收回权限。如果选中“仅列出此对象具有权限的用户/用户定义的数据库的角色/public”，则下面的列表框中将显示该有权访问该对象的用户以及详细的权限设置信息。这个步骤可以通过在图 5.15 中选择“所有任务”→“管理权限...”达到。

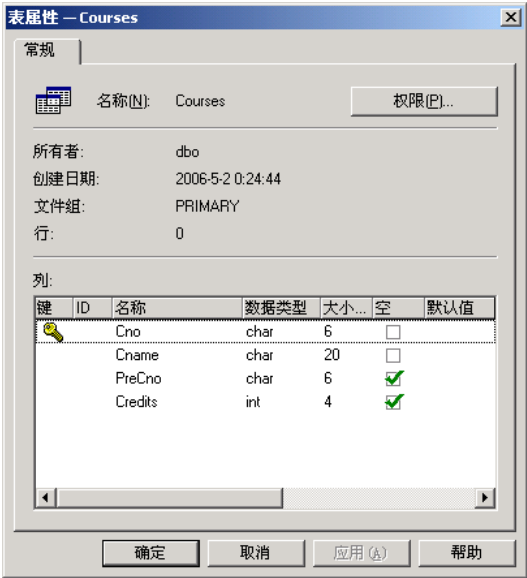


图 5.16 对表对象 courses 的权限设置

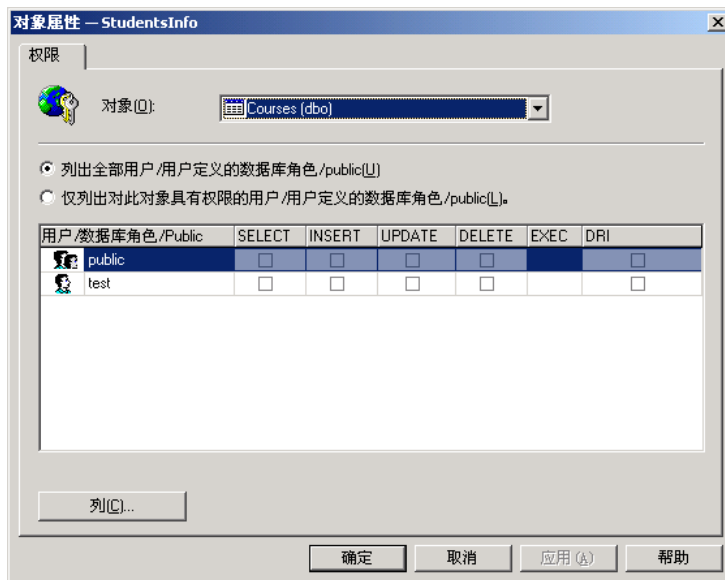


图 5.17 对象 courses 的权限设置

3) 单击“确定”按钮保存设置。

可以看出，“对象属性”对话框与前面的“数据库用户属性”对话框很类似。在“数据库用户属性”对话框中，是为某一用户设置其对当前数据库的所有对象的访问权限，而在“对象属性”对话框中是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。

(2) 使用 Transact-SQL 语句管理权限

在 Transact-SQL 语句中，用于管理权限的语句有 3 个：GRANT，用于授权；REVOKE 语句，用于收回权限；DENY 语句，用于拒绝权限。

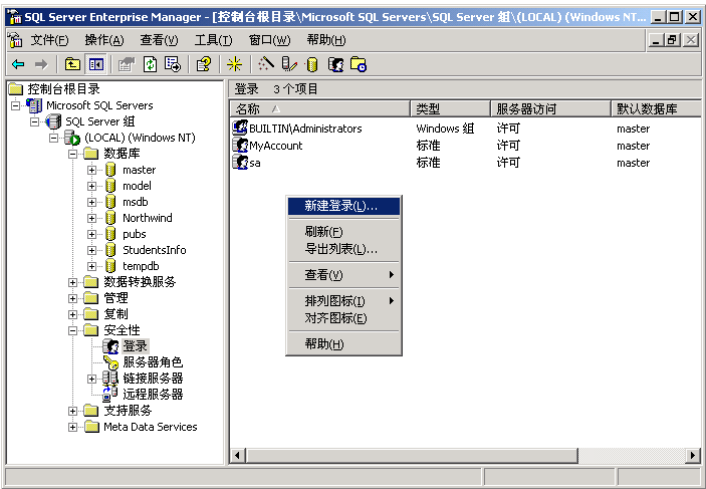
其中，语法格式如表 5.2 所示。

表 5.2 SQL Server 中管理权限的 Transact-SQL 语句

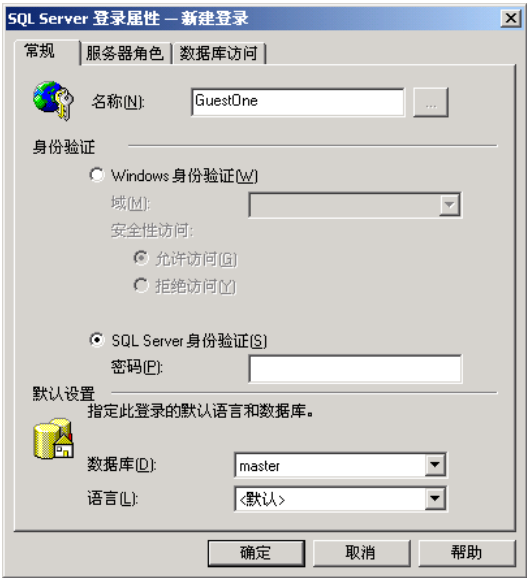
类别	语句	语法
管理对象权限	GRANT	GRANT 对象权限名[,...]ON{表名 视图名 存储过程名} TO {数据库用户名 角色名}[,...]
	REVOKE	REVOKE 对象权限名[,...]ON{表名 视图名 存储过程名} FROM {数据库用户名 角色名}[,...]
	DENY	DENY 对象权限名[,...]ON{表名 视图名 存储过程名} TO {数据库用户名 角色名}[,...]
管理语句权限	GRANT	GRANT 语句权限名[,...] TO {数据库用户名 角色名}[,...]
	REVOKE	REVOKE 语句权限名[,...] FROM {数据库用户名 角色名}[,...]

	DENY	DENY 语句权限名[,...] TO {数据库用户名 角色名}[,...]
--	------	--

例 5.1 用前面介绍的方法，首先创建一个新的登录帐号 GuestOne，采用 SQL Server 身份验证，密码为空(见图 5.18)。并替 StuentInfo 数据库创建一个与 GuestOne 对应的用户帐号：rightstest(见图 5.19)。在查询分析器中，用sa 登录(查询分析器的用法可参见附录 A)，我们用 GRANT 语句为 rightstest 分配 Students 表中 SELECT 权限(见图 5.20)。



a) 新建登录



b) 登录帐号

图 5.18 新建登录帐号 GuestOne



a) 新建用户



b) 用户帐号

图 5.19 新建 Studentsinfo 的用户帐号

GRANT SELECT ON Students To righttest

注意: 要用 Windows 合法用户以外的其他登录帐号登录, 则在 SQL Server 的服务器属性的安全性页框中必须选择“SQL Server 和 Windows”的身份验证

模式。

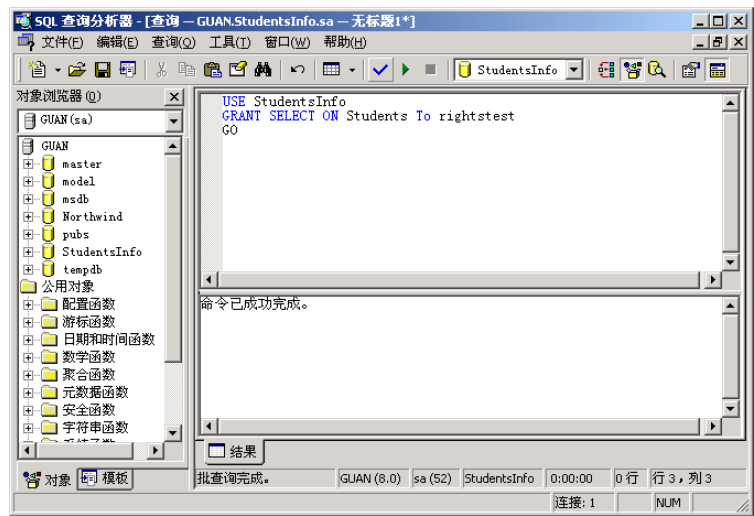


图 5.20 为用户帐号 rightstest 分配权限

在查询分析器中，用 GuestOne 登录，使用 SELECT 语句：SELECT * FROM Students, 执行成功(见图 5.21)；执行 UPDATE 语句：UPDATE Students SET Sname='John' WHERE Sname='Jone'，更新 Students 表时失败(见图 5.22)。查询分析器的消息窗同时给出语句执行失败原因：当前的用户(rightstest)不拥有在表对象 Students 上执行 UPDATE 语句的权限。

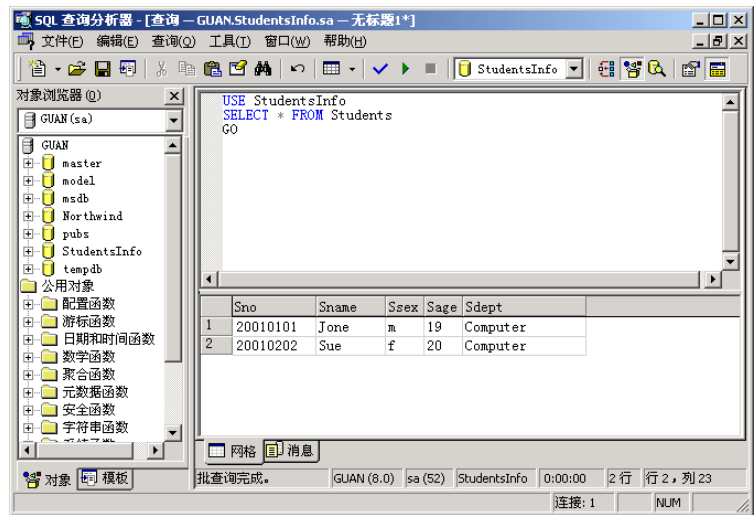


图 5.21 普通用户 rightstest 被允许执行 SELECT 语句

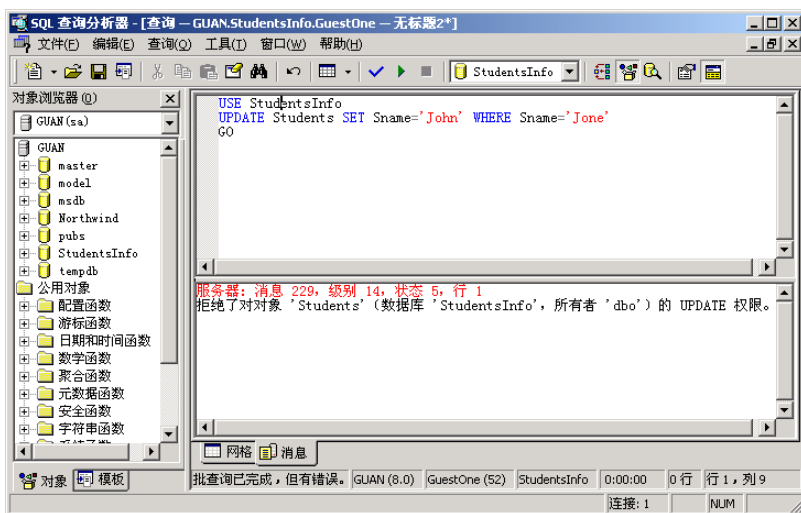


图 5.22 普通用户 rightstest 无权执行 UPDATE 语句

例 5.2 用 sa 再次登录查询分析器,收回刚刚分配给 rightstest 的 SELECT 权限: REVOKE SELECT ON Students FROM rightstest, 如图 5.23 所示。然后重新以 GuestOne 登录查询分析器, 执行语句: SELECT*FROM Students, 执行失败(见图 5.24)。查询分析器同时给出执行失败的原因: 是因为执行了 REVOKE 操作后, 当前的 rightstest 用户已经不拥有可以执行 SELECT 语句的权限了。

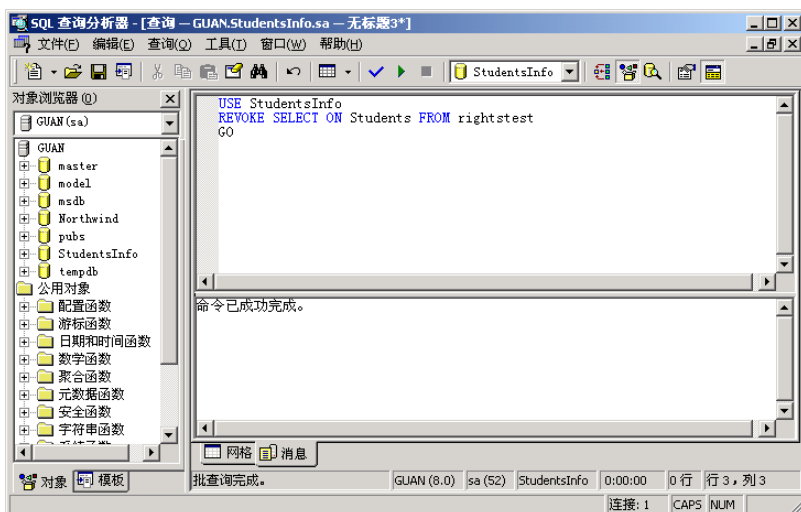


图 5.23 收回普通用户 rightstes 执行 SELECT 语句的权限

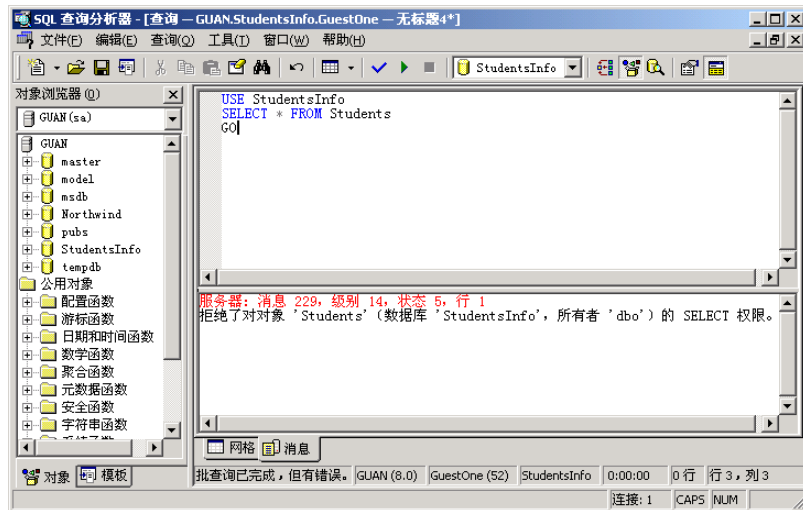


图 5.24 普通用户 rightstest 没有了执行 SELECT 语句的权限

第六章 数据库备份机制

一、为什么要进行数据库备份

尽管 SQL Server 提供了内置的安全性和数据保护,但这种安全管理主要是为了防止非法登录者或没有经过的授权用户对 SQL Server 数据库或数据造成破坏。在有些情况下,这种安全管理机制是无能为力的,比如说,病毒破坏、计算机设备硬件故障及误操作等。在这些情况发生时,很可能导致数据的丢失和破坏。为了能尽快恢复系统的正确工作并把损失降低到最低,必须对系统数据进行备份,以便在需要时能够及时恢复数据。数据库备份是指系统管理员定期地将数据库部分或全部地复制到磁带或另一个磁盘上保存起来的过程。这些复制的数据拷贝称为后备副本。当数据库遭到破坏时,可以利用后备副本进行数据库的恢复,但也只能恢复到备份时的状态。要使数据库恢复到发生故障时刻前的状态,必须重新运行自备份以后到发生故障前的所有事务(事务是指一个不可分割的数据库操作工作单元,一个事务内的动作,要么全部做完,要么全部不做,它保证了数据库操作的一致性和可恢复性)。

对 SQL Server 而言,备份针对数据库或者日志进行,数据库备份记录了在进行备份这一操作时数据库中所有数据的状态,以便在数据库遭到破坏时能够及时的将其恢复。SQL Server 备份数据库是动态的:在进行数据库备份时,SQL Server 允许其他用户继续对数据库进行操作。要执行备份操作必须先拥有对数据库备份的权限,SQL Server 只允许系统管理员、数据库所有者和数据库备份执行者备份数据库。备份是数据库系统管理的一项重要内容,也是系统管理员的日常工作。

二、SQL Server 2000 数据库备份的种类

SQL Server 2000 支持 4 种备份形式:

(1) 数据库备份(完全数据库备份)

数据库备份(Database Backup)是指对数据库的完整备份,包括所有的数据对象、数据和事务日志中的事务。这种备份方式非常简便易行,通常按照一个常规的时间间隔进行。在还原数据库时,只需要简单的操作即可完成数据库的恢复。恢复进程重写现有数据库,如果现有数据库不存在则新建一个。恢复后的数据库与备份完成时的数据库状态一致,但不包括任何未提交的事

务。

注意：与差异数据库备份和事务日志备份相比，完全数据库备份由于是对数据库的完整备份，所以这种备份类型不仅速度较慢，而且会占用大量磁盘空间，备份的创建频率通常比差异数据库或事务日志备份低；另一方面，由于完全数据库备份不能频繁的创建，使其不能最大程度地恢复丢失的数据，它只能用作系统失败时恢复数据库的基础。例如：假设系统在每晚 0 点进行完全数据库备份，而服务器在备份前 1 小时崩溃了，那么，过去 23 小时的数据都将无法根据备份来恢复。

这种备份方式应该与下面的其他几个备份方法相互结合，才能最大程度地对数据库数据进行保护。如果数据库的数据不是十分重要，或者数据库中的数据可以很容易地重建，亦或者数据库是只读的，那么可以单独采用完全数据库备份。

(2) 差异数据库备份(增量备份)

差异备份(Differential Database Backup)是指将最近一次数据库备份(完全数据库备份)以来发生的数据变化备份起来，因此差异数据库备份实际上是一种增量数据库备份。对于一个经常进行数据操作的数据库而言，需要在完全数据库的备份的基础上，进行差异备份。差异数据库备份比完全数据库备份需要的磁盘空间小而且备份速度快，因此可以更经常地备份。通过增加差异备份的次数，可以减少丢失数据的危险。

使用差异数据库备份只能将数据库还原到差异数据库备份完成时的那一点，若要恢复到精确的故障点，必须使用事务日志备份。

在下列情况下，可考虑使用差异数据库备份：

- 1) 最近一次完全数据库备份后数据库中只有较少的数据发生了改变；
- 2) 希望进行频繁的备份，但不希望进行频繁的完全数据库备份；
- 3) 希望在还原数据库时花费最少的时间前滚事务日志备份。

(3) 事务日志备份

事务日志备份(Transaction Log Backup)是对数据库发生的事务进行备份，包括从上次事务日志备份、差异备份和完全数据库备份后，数据库已经执行完成的所有事务。它可以在相应的数据库备份的基础上，将数据库恢复到特定的即时点或恢复到故障点时的状态。

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完全数据库备份(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，即每天进行一次完全数据库备份，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。

在下列情况下，可考虑使用事务日志备份：

- 1) 不允许在最近一次完全数据库备份之后发生数据丢失或损坏现象。

- 2) 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。
- 3) 准备把数据库恢复到发生故障的前一点。
- 4) 数据库变化较为频繁。

差异备份和事务日志备份需要的时间相对于完全数据库备份而言都少，但它们之间有一个重要的差别：事务日志备份含有自上次备份以来某一行修改几次的所有修改，它记录操作的过程；而差异备份只含有该行的最后一次修改，它记录动作的结果。

(4) 数据库文件和文件组备份

文件或文件组备份(File and File Group Backup)是指数据库文件或文件夹进行备份，但其不像完全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，因为其仅为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。

当 SQL Server 系统备份文件或文件组时，指定需要备份的文件，最多指定 16 个文件或文件组。

三、SQL Server 2000 数据库备份的方法

在进行备份前必须先指定或创建备份设备。备份设备是指 SQL Server 中存储数据库、事务日志或文件及文件组备份的存储介质，可以是本地的磁盘和远程服务器上硬盘、磁带或者命名管道。当建立一个备份设备时，需要给其分配一个逻辑名和一个物理名。物理名是操作系统用来标识备份设备的名称。逻辑名是用来标识物理备份设备的别名。逻辑设备名称永久地存储在 SQL Server 的系统表中。使用备份设备逻辑名比使用物理名简单。

逻辑名最多 30 个字符并且必须遵守 SQL Server 的命名约定。备份数据库时，可以使用物理名或逻辑名，然后执行备份操作对数据库进行备份。

下面通过创建选课数据库 StudentsInfo 的备份来说明数据备份的方法。

1. 创建备份设备

创建备份设备(backup device)有两种方法：使用企业管理器和使用系统存储过程 `sp addu-mpdevice`。

(1) 使用企业管理器创建备份设备

- 1) 在企业管理器中，选择要创建备份设备的服务器，双击展开“管理”文件夹、右击“备份”，在弹出的菜单中选择“新建

备份设备”项，如图 6.1 所示。

- 2) 进入“备份设备属性”窗口，如图 6.2 所示。在名称栏中输入备份设备名称，该名称是备份设备的逻辑名称。这里假定为 StudentsInfo 数据库创建的备份设备逻辑名称为 StudentsInfoBK。

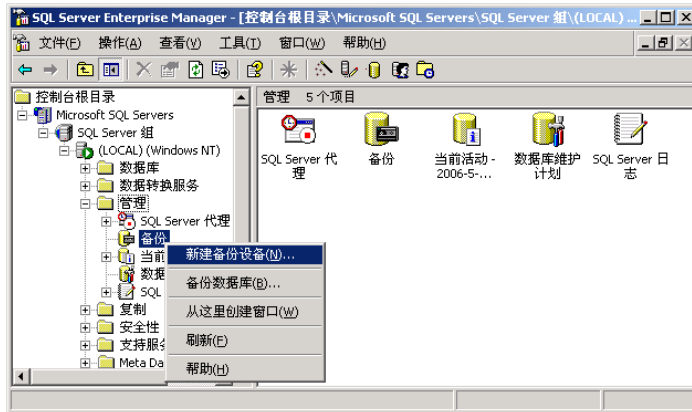


图 6.1 新建备份设备

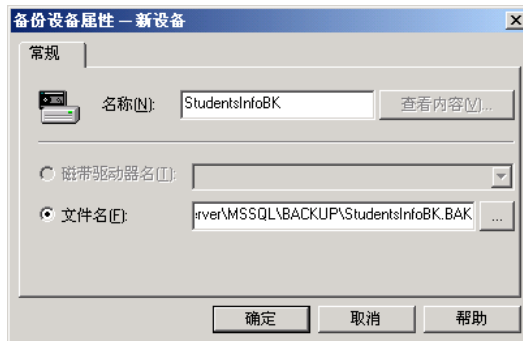



图 6.2 备份设备属性设置

- 3) 选择备份设备的类型，以确定备份设备的物理名称。如果选择“文件名”表示使用硬盘做备份，只有正在创建的备份是硬盘文件时，该选项才有效。如果选择“磁带驱动器”表示使用磁带设备，只有正在创建的备份设备是与本地服务器相连的磁带设备时，该选项才有效(如果像图 6.2 上，没有列出磁带设备，则表明本地计算机无法检测到磁带设备)。

这里，选择“文件名”。然后使用下列方法之一来确定备份设备的物理名

称：①输入备份设备所使用的文件名。它应该是一个完整的路径和文件名；

②单击  浏览按钮，显示“备份设备位置”对话框，再选择备份设备所使用的本地计算机上的物理文件。默认情况下，SQL Server 根据输入的设备逻辑名自动在系统目录 BACKUP 下生成同名的物理名称。

4) 单击“确定”按钮，完成创建备份设备的操作。

(2) 使用系统存储过程创建备份设备

SQL Server 的系统存储过程 `sp addumpdevice` 可以用来创建备份设备。

其语法形式如下：

`sp addumpdevice[@devtype=]` 备份设备类型', `[@logicalname=]` 备份设备的逻辑名称'

其中：

- 备份设备类型 值为 `disk`(以硬盘文件作为备份设备)、`pipe`(命名管道备份设备)或者 `tape`(磁带备份设备)之一。
- 备份设备的逻辑名称 是用户命名的备份设备的逻辑名称。
- 备份设备的物理名称 是备份设备对应的物理备份设备文件全名。

例 6.1 为学生选课数据库 `StudentsInfo` 创建一个本地的备份设备。

USE `StudentsInfo`

EXEC `sp addumpdevice 'disk', 'studentsInfoBK', 'C:\Program Files\Microsoft SQL Server\MSSQL\BACKUP\StudentsInfoBK.BAK'`

GO

删除备份设备可以释放备份空间。在企业管理器中选中要删除的备份设备，单击鼠标右键，在弹出的菜单上选择“删除”项即可；或者使用系统存储过程 `sp_dropdevice` 来删除备份设备。语法形式如下：

`sp dropdevice[@logicalname=]` 是指数据库设备或备份设备的逻辑名称'`[,@delfile=]` 应该删除物理备份设备文件'

2. 执行备份

SQL Server 系统提供了 3 种数据库备份操作的方法：企业管理器、备份向导和 BACKUP 语句。

(1) 使用企业管理器进行备份

1) 在企业管理器中选中“数据库”文件夹，单击鼠标右键，在弹出的菜单上选择“所有任务”→“备份数据库”命令(图 6.3)；或者在企业管理器的菜单上依次选择“操作”→“所有任务”→“备份数据库”命令。

2) 进入 SQL Server 备份对话框(见图 6.4)。在“常规”页框中分 5 个设置部分：

① 选择要备份的数据库，输入备份操作的名称和描述信息，在此选

StudentsInfo。描述忽略；

- ② 根据备份的需要选择备份的类型，在此使用默认选项“完全数据库备份”；
 - ③ 备份介质的设置，可以通过“添加”按钮操作；
 - ④ “追加到媒体”表示将备份内容添加到当前备份的后面，“重写现有媒体”表示备份内容会覆盖原来的备份文件；
 - ⑤ “调查”复选框可以按计划执行数据库的备份，而不是立即执行。
- 3) 单击“添加”按钮，设置备份设备(见图 6.5)。用户可以选择“文件名”，根据设备的物理名称来指定；或者选择“备份设备”，通过逻辑名称来确定。这里选择“备份设备”，然后在下拉框中选择刚才创建的设备名称：StudentsInfoBK，单击“确定”按钮，返回图 6.4。

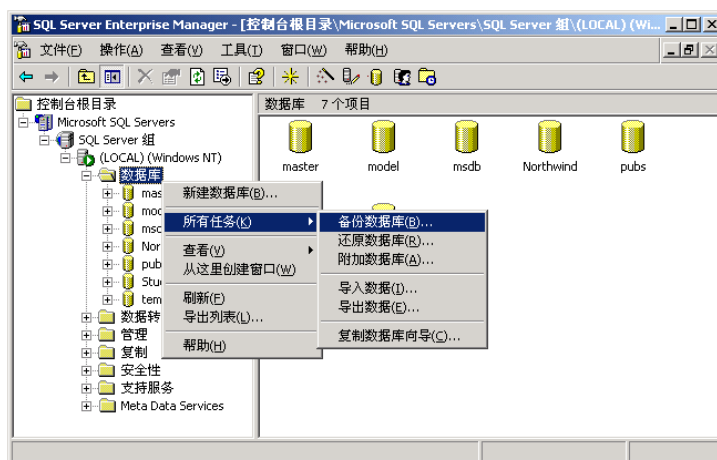


图 6.3 备份数据库

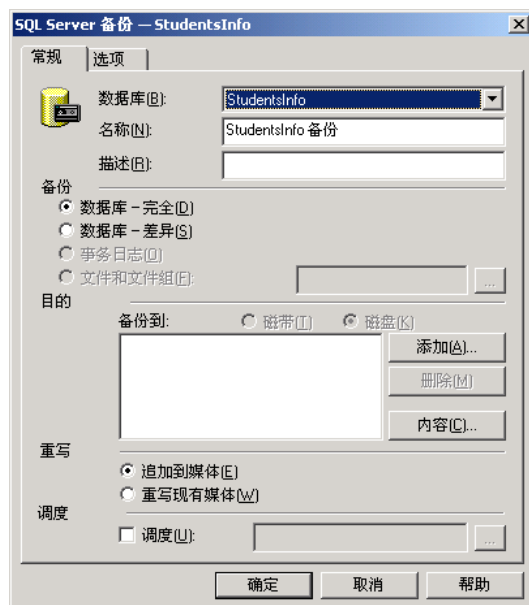


图 6.4 SQL Server 备份设置—“选项”选项卡

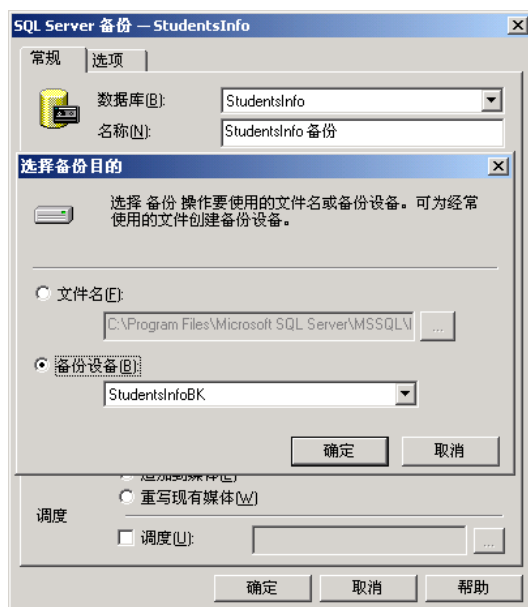


图 6.5 选择备份目的对话框

- 4) 如果不想立即进行备份，则选择图 6.4 的“调度”。可以使用



按钮在弹出的编辑调度窗口(见图 6.6)中设置调度计

划，默认的调度计划为“第 1 调度”。对于周期性的备份动作，可以通过编辑反复出现的调度作业来安排具体的备份动作的发生时间和频率，可以在单击图 6.6 中的“更改”按钮后出现的编辑窗口中设置(见图 6.7)。

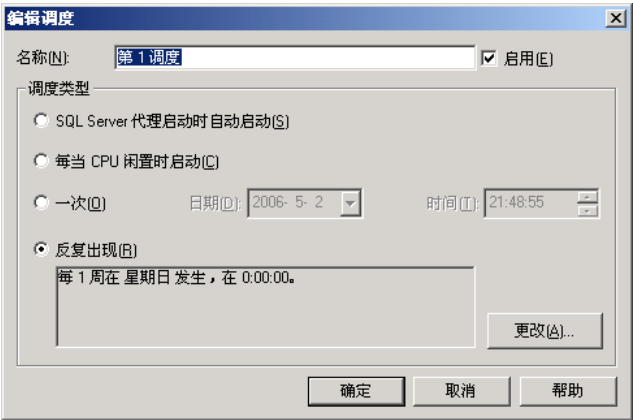


图 6.6 “编辑调度”对话框

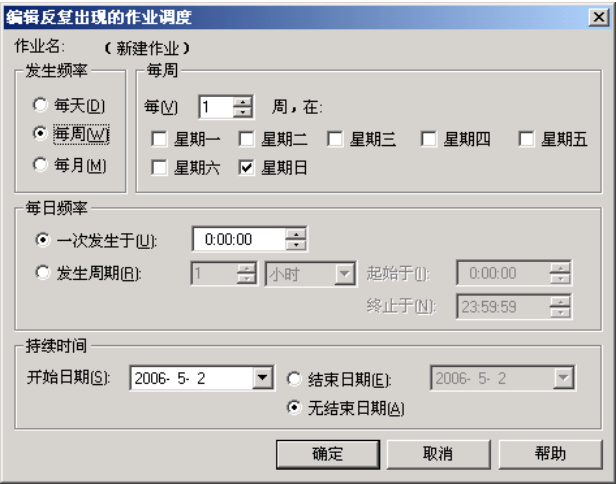


图 6.7 编辑反复出现的作业调度

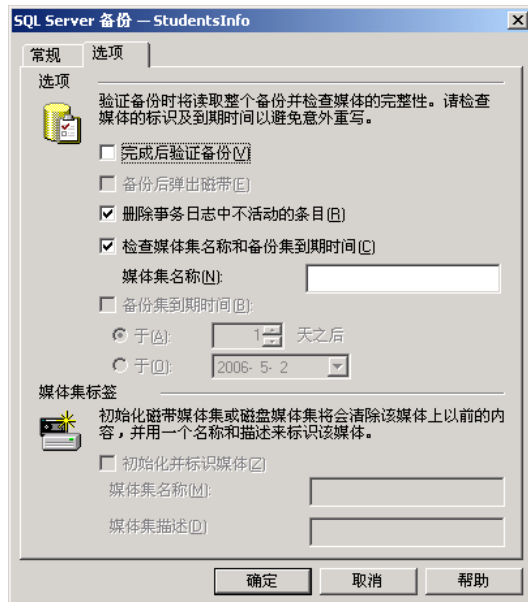


图 6.8 SQL Server 备份设置—“选项”选项卡

- 5) 可以在进入 SQL Server 备份对话框的“选项”选项卡中进行备份的附加设置，如图 6.8 所示。“完成后验证备份”表示备份完成后 SQL Server 阅读全部备份，检查备份介质是否可读，保证备份的完整性。“检查媒体集名称和备份集到期时间”表示执行备份前，检查备份介质集的名称和失效日期，以防意外重写。“媒体集名称”指定要检查备份集的名称，如果仅指定备份集到期时间，则将其保留为空。如果是第一次使用备份媒体，或者要更改现有的媒体标签，则在“媒体集标签”框下选择“初始化并标识媒体”复选框，然后输入媒体集名称和媒体集描述。只有在重写媒体时才能对其进行初始化和标识设置。
- 6) 单击“确定”按钮开始数据库备份。备份完成后，系统给出提示，如图 6.9 所示。

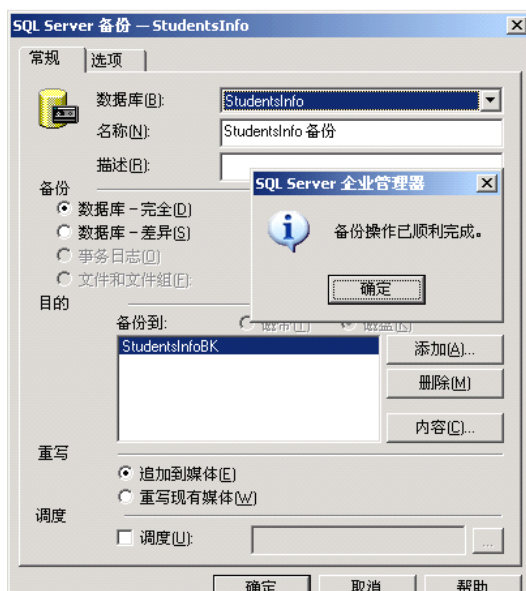


图 6.9 SQL Server 备份操作完成

(2) 使用备份向导进行备份

- 1) 在企业管理器中选中要创建备份的服务器，然后依次选择“工具”→“向导”→“管理”命令，在出现的向导选择对话框中选择“备份向导”(见图 6.10)，单击“确定”按钮。
- 2) 出现如图 6.11 所示的数据库备份向导欢迎窗体。单击“下一步”按钮，在出现的窗口上选择要备份的数据库 StudentsInfo(见图 6.12)，再单击“下一步”按钮。

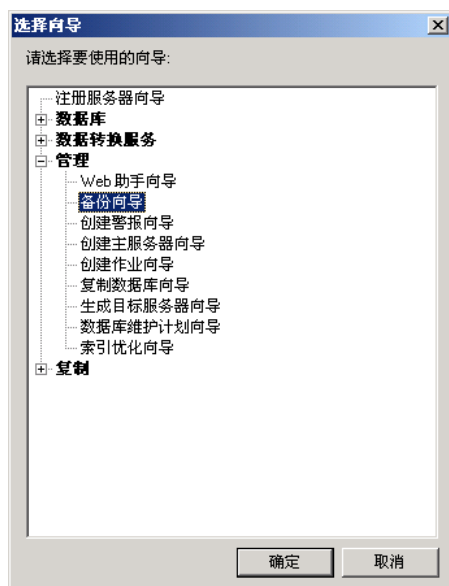


图 6.10 选择向导对话框

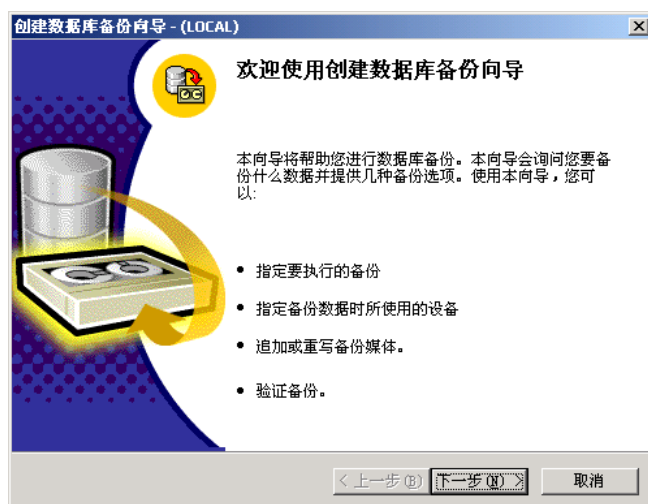


图 6.11 备份向导的欢迎界面

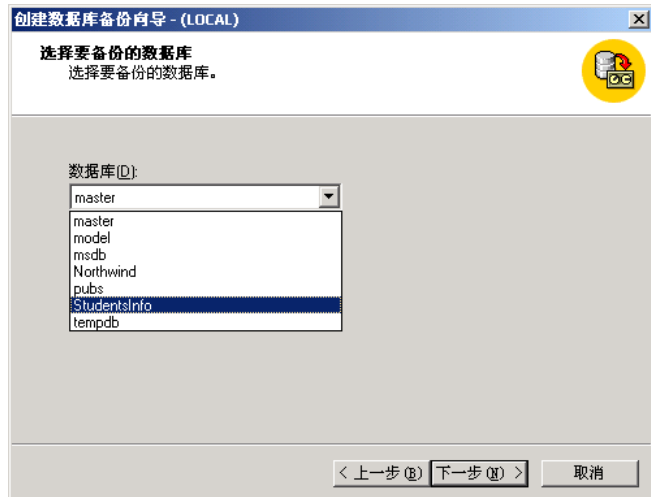


图 6.12 选择要备份的数据库

- 3) 在如图 6.13 的窗口中设置备份名称和备份的描述，可以直接单击“下一步”按钮进入备份类型选择窗口(见图 6.14)，选择“备份整个数据库”，单击“下一步”按钮。

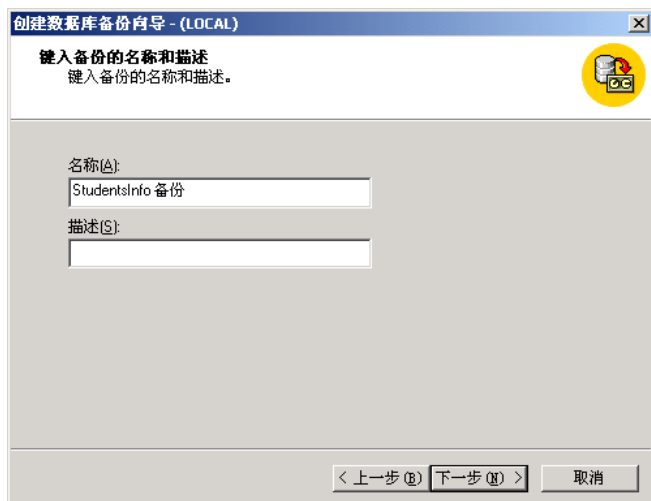


图 6.13 输入备份数据库名称和描述

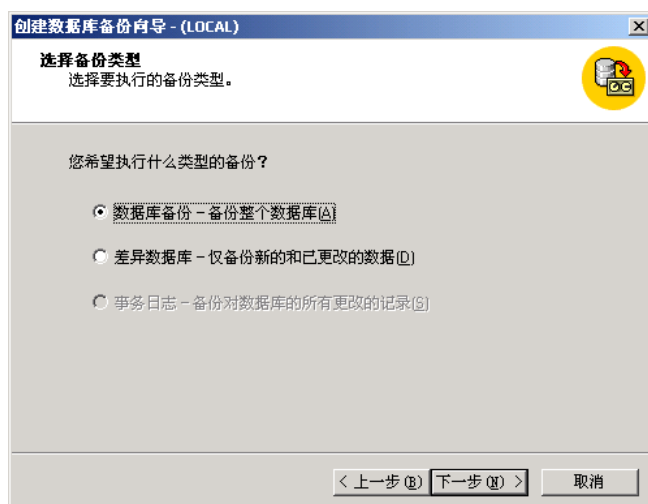


图 6.14 选择备份类型

- 4) 在图 6.15 中选“备份设备”，在其下拉列表中选择刚才创建的设备名称：Students-InfoBK；由于前面已经使用企业管理器创建了一个同名备份，所以在属性选区中选“重写媒体设备”。单击“下一步”按钮，出现初始化媒体设置，再次单击“下一步”按钮，忽略该媒体设置。

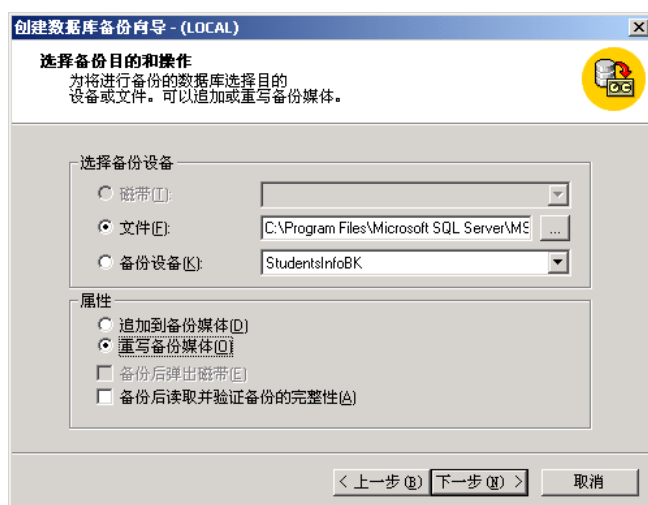


图 6.15 选择备份设备和设置属性

- 5) 在图 6.16 中设置备份验证和调度属性。单击“下一步”按钮，出现图 6.17 的备份完成窗口。单击“完成”按钮后，SQL

Server 创建备份并给出完成提示。

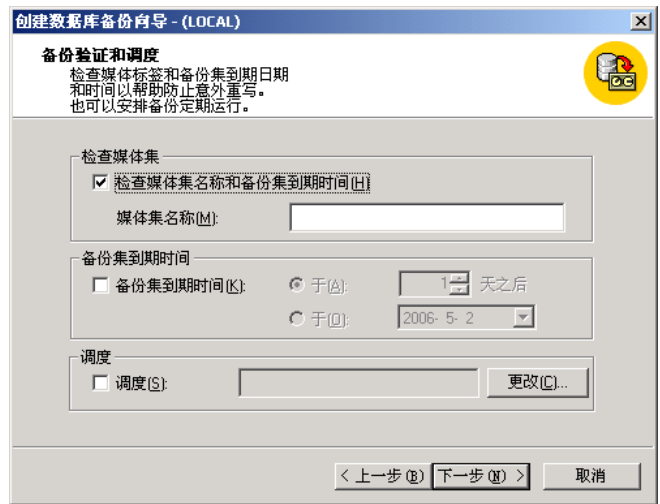


图 6.16 备份验证和高度

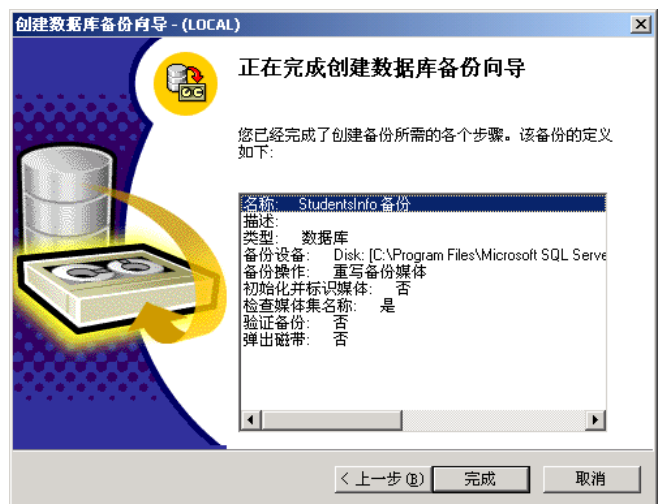


图 6.17 备份完成

(3) 使用 BACKUP 语句进行备份

使用 Transact-SQL 语句进行备份时，需要使用 BACKUP DATABASE 语句和 BACKUP LOG 语句备份整个数据库、事务日志，或者备份一个或多个文件，或者备份文件组。

其中 BACKUP DATABASE 语句的最简单的语法形式如下：

BACKUP DATABASE 要备份的数据库名称 TO 备份设备

例 6.2 为学生选课数据 StudentsInfo 创建备份到设备 StudentsInfoBK 上。

```
USE StudentsInfo
BACKUP DATABASE StudentsInfo To StudentsInfoBK
GO
```

第七章 数据库恢复机制

一、为什么要进行数据库恢复

数据库备份以后，一旦系统发生崩溃或者执行了错误的数据库操作，就可以从备份文件中恢复数据库。恢复数据库是指将数据库从错误状态恢复到正确状态(最近的正确时刻)的过程。从某种意义上讲，数据库的恢复比数据库的备份更加重要，因为数据库备份是在正常的工作环境和数据库的正常状态下进行的，而数据库恢复是在非正常状态下进行的，比如硬件故障、软件瘫痪以及误操作。

数据库系统在恢复过程中，自动执行安全检查、重建数据库结构以及完成填写数据库内容的任务。安全性检查是恢复数据库时必须执行的过程，它可以防止用户使用错误的数据库备份文件或者不兼容的数据库备份去覆盖已经存在的数据库。SQL Server 恢复数据库时，根据数据库备份文件自动创建数据库结构，并且恢复数据库中的数据。

二、SQL Server 2000 数据库恢复的种类

SQL Server 2000 支持 3 种数据库恢复模式

1) **简单恢复 (Simple Recovery)** 指在进行数据库恢复时仅使用了完全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故障发生时的状态。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。

2) **完全恢复 (Full Recovery)** 指通过使用完全数据库备份和事务日志备份将数据库恢复到发生故障的时刻，因此几乎不造成任何数据丢失，这种恢复模式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故障发生时刻则必须对数据库故障发生前正处于运行状态的事务进行备份。

3) **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所需要的存储空间。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。

三、SQL Server 2000 数据库恢复的方法

恢复数据库之前，首先要保证所使用备份文件的有效性，并且在备份文件中包含所要恢复的数据内容。在 SQL Server 企业管理器中可以直接看到备份文件的属性信息，包括文件名、备份时间和备份的数据库名称等。Transact-SQL 还提供了更详细的查看备份文件信息的语句。其次，应该删除故障数据训，以便删除对故障数据库的任何引用；最后，由于数据库的恢复动作是静态的：不允许其他用户在数据库恢复时对数据库进行访问和操作。所以，必须限制用户对数据库的访问为“单用户”模式。用户可以使用企业管理器选中要恢复的数据库，在弹出的菜单中选择“StudentInfo 属性”对话框中设置数据库访问属性(见图 7.1)，或者利用系统存储过程 sp dboption 设置数据库为单用户模式。

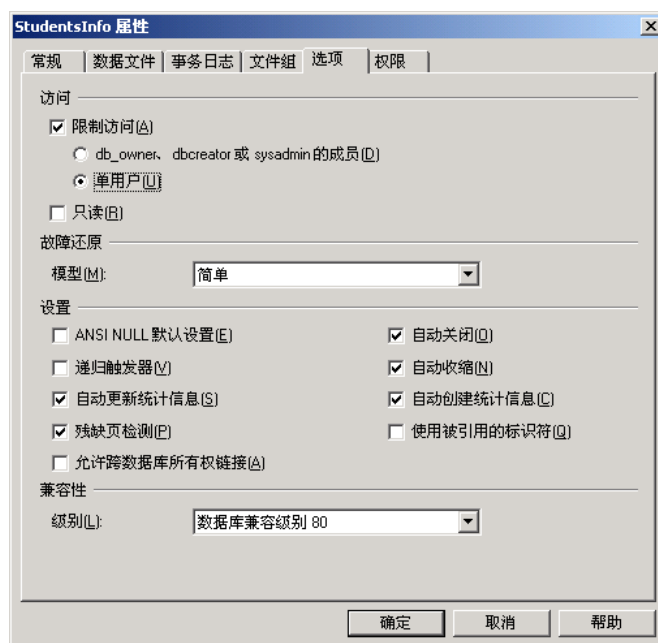


图 7.1 设置访问数据库属性

1. 普通数据库恢复

SQL Server 系统提供了两种数据库恢复的操作方法：使用企业管理器和使用 RESTORE 语句。

(1) 使用企业管理器恢复数据库

- 1) 在企业管理器中选中“数据库”文件夹，单击鼠标右键，在弹出的菜单上选“所有任务”→“还原数据库”(见图 7.2)；或者在企业管理器的菜单上依次选择“操作”→“所有任务”→“还原数据库”命令。
- 2) 在还原数据库对话框中进行还原数据库的设置。

在“常规”选项卡(如图 7.3 所示)中选择要恢复的数据库；设置用于恢复数据库的备份类型，默认为“数据库”，则在参数栏中会显示备份历史。可以在“要还原的第一个备份”下拉列表中选择要使用的备份，默认情况使用最近一次的备份。例如，选中“从设备”还原的选项，则要进行设备设置(如图 7.4 所示)，单击“选择设备”按钮，在弹出的设备选择对话框(见图 7.5)中单击“添加”按钮，将还原设备设置为前面创建的“StudentsInfoBK”，图 7.6 所示。

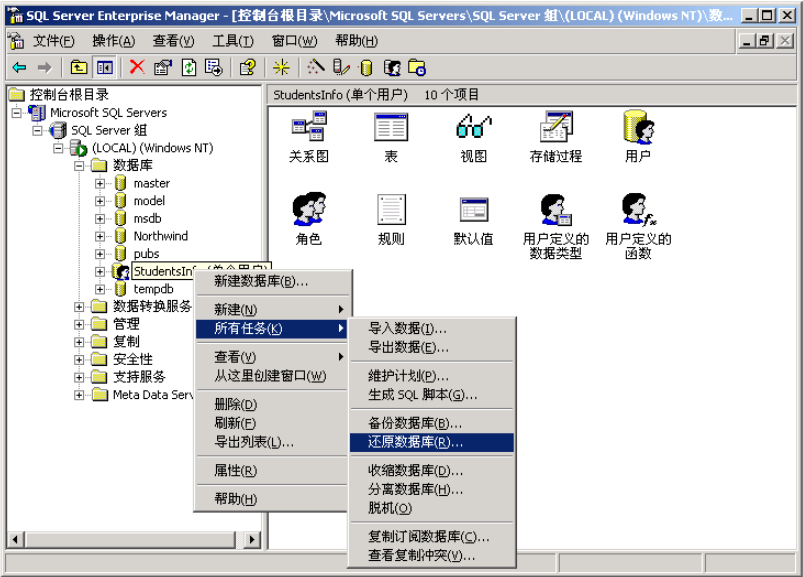


图 7.2 还原数据库

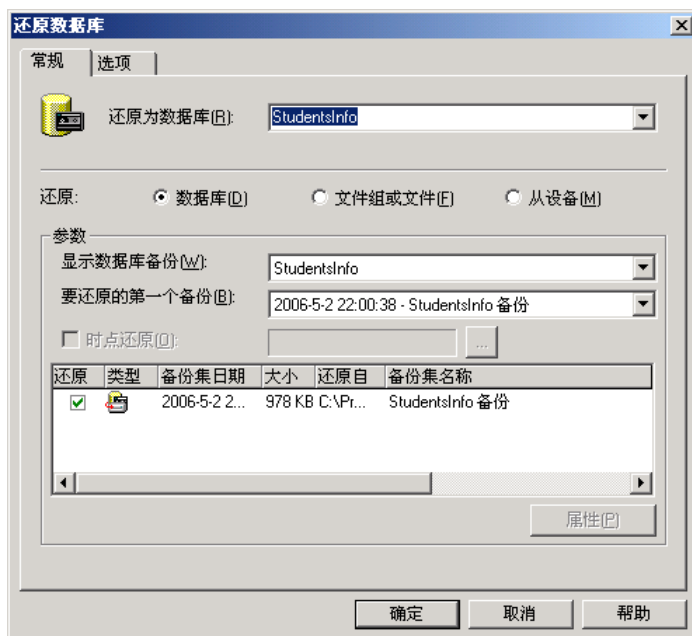


图 7.3 “还原数据库”对话框-“常规”选项卡

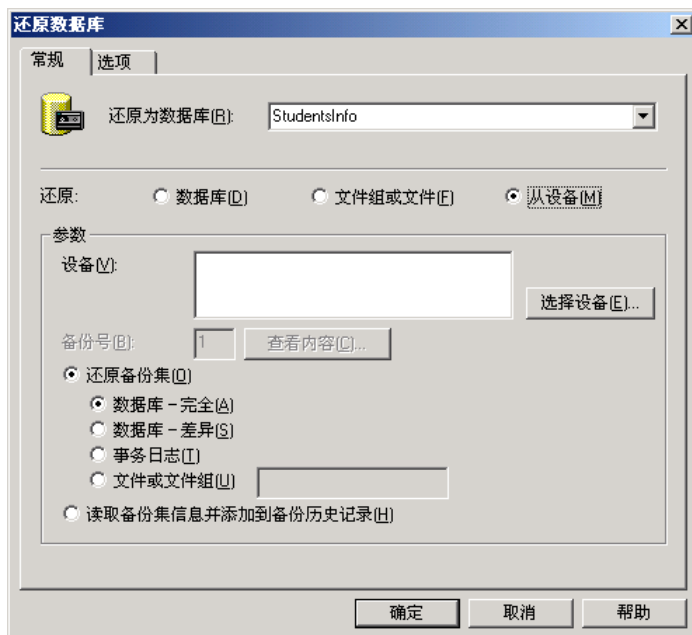


图 7.4 “从设备还原”对话框

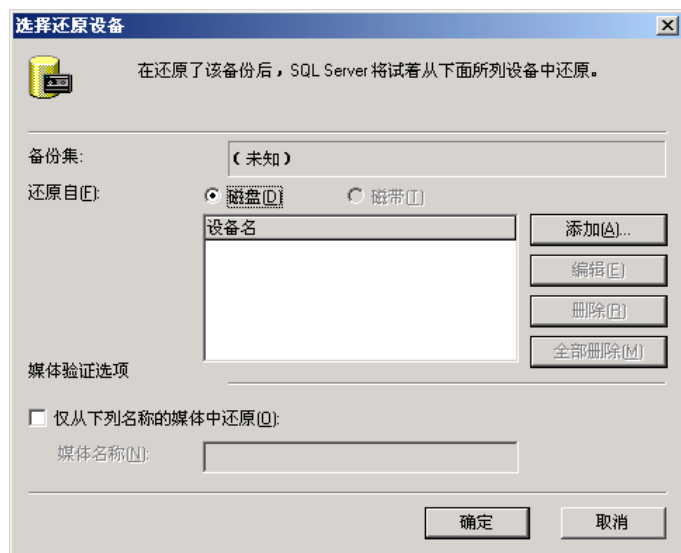


图 7.5 “还原选择设备”对话框

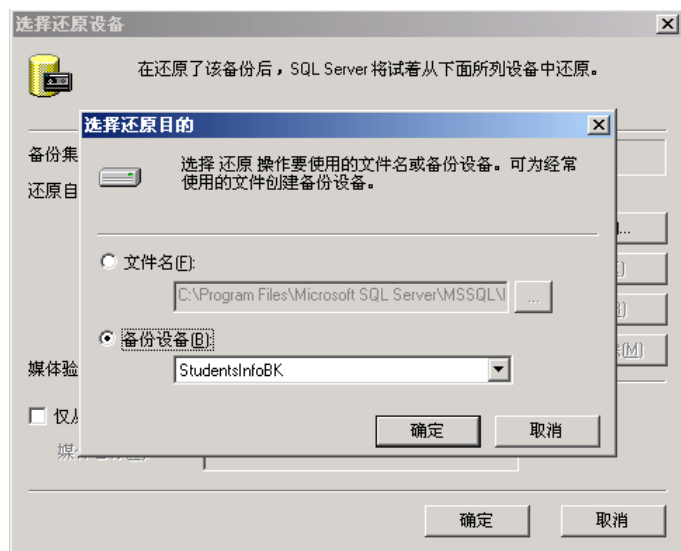


图 7.6 还原目的选择

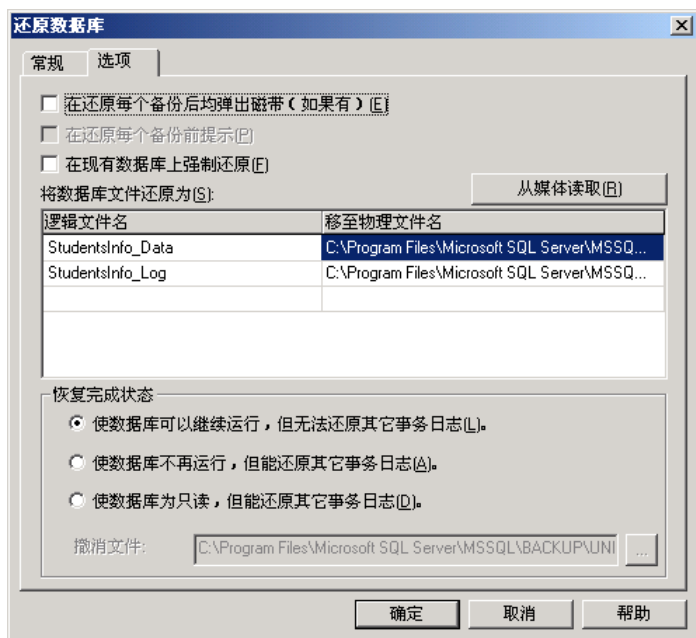


图 7.7 “选项”选项卡

在“选项”选项卡中进行其他项的设置(如图 7.7 所示), 各选项含义如下:

- “在还原每个备份后均弹出磁带(如果有)”表示如果使用的备份是磁带, 则在恢复数据库完成后, 将磁带从磁带机中弹出。
- “在还原每个备份前提示”表示要求系统在恢复数据库前进行备份信息的提示。
- “在现有的数据库上强制还原”表示要恢复的数据库已经存在, 使用备份覆盖已存在的数据库。本例中, 选项该项, 是因为 StudentsInfo 数据库已经存在。
- “恢复完成状态”选项含义如下:
- “使数据库可以继续运行, 但无法还原其他事务日志”表示当前的恢复动作是最后一次, 执行完成后, 无法恢复其他事务日志, 数据库仍然可以使用。
- “使数据库不再运行, 但能还原其他事务日志”表示当前的恢复动作不是最后一次, 恢复完成后, 数据库仍然不能使用, 还要继续恢复。
- “使数据库为只读, 但能还原其他事务日志”表示当前的恢复完成后, 数据库只能作为只读数据库, 而且还可以继续恢复其他事务日志, 此时可以指定一个撤销文件, 用于取消对数据库中的更改。

3) 单击“确定”按钮, 则执行恢复数据库操作, 完成后给出还

原完成提示(见图 7.8)。

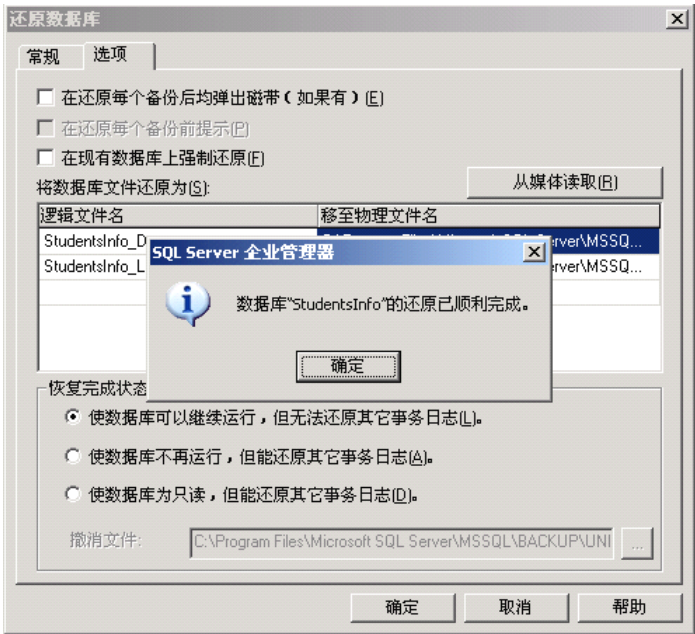


图 7.8 数据库恢复完成

(2) 使用 RESTORE 语句恢复数据库

Transact-SQL 中的 RESTORE 用于恢复数据库。最简单的语法形式如下:
RESTORE DATABASE 要恢复的数据库名称 FROM 备份设备

例 7.1 使用完全数据库备份 StudentsInfoBK 恢复学生选课数据库 StudentsInfo。

```
USE StudentsInfo
RESTORE DATABASE StudentsInfo FROM StudentsInfoBK
GO
```

2. 系统数据库恢复

在备份用户数据库时需要备份系统数据库，这使得在系统或数据库发生故障(如果硬盘发生故障)时可以重建系统。需要经常备份的系统数据库包括 master 和 msdb。如果 model 数据库被修改过，也需要进行定期备份。

系统数据库的备份与恢复操作与用户数据库的备份与恢复操作相似，这里不再重复。

第八章 练习

设有一图书馆数据库，其中包括 3 个表，即图书表、读者表和借阅表。三个表的结构如表 8.1、表 8.2 和表 8.3。

表 8.1 图书表结构

列名	说明	数据类型	约束说明
书号	图书唯一的编号	定长字符串，长度为 20	主键
书名	图书的名称	变长字符串，长度为 50	空值
作者	图书的编著者名	变长字符串，长度为 30	空值
出版社	图书的出版社	变长字符串，长度为 30	空值
单价	出版社确定的图书的单价	浮点型，FLOAT	空值

表 8.2 读者表结构

列名	说明	数据类型	约束说明
读者号	读者唯一的编号	定长字符串，长度为 10	主键
姓名	读者姓名	定长字符串，长度为 8	非空值
性别	读者性别	定长字符串，长度为 2	非空值
办公电话	读者办公电话	定长字符串，长度为 8	空值
部门	读者所在部门	定长字符串，长度为 30	空值

表 8.3 借阅表结构

列名	说明	数据类型	约束说明
读者号	读者唯一的编号	定长字符串，长度为 10	外码，引用读者表的主键
书号	图书唯一的编号	定长字符串，长度为 20	外码，引用图书表的主键
借出日期	借出图书的日期	定长字符串，长度为 8，为‘yyyymmdd’	非空值
归还日期	归还图书的日期	定长字符串，长度为 8，为‘yyyymmdd’	空值
主键为：(读者号，图书号)			

数据如表 8.4、表 8.5 和表 8.6 所示。

表 8.4 图书表数据

书号	书名	作者	出版社	单价
TN913.2/530	21 世纪的电信网	盛友招	人民邮电出版社	7.5
TP31.13/CM3	数据库原理及应用	苗雪兰	机械工业出版社	28
TP31.132/ZG1	XML 数据库设计	尹志军	机械工业出版社	38
TP316/ZW6	操作系统	吴庆菊	科学出版社	5
TP316/ZY1	操作系统	沈雪明	电子工业出版社	31
TP391.132.3/ZG5	网络数据库精粹	李智等	机械工业出版社	45

表 8.5 读者表数据

读者号	姓名	性别	电话
001973	王平	男	88320701
001974	张丽	女	88320701
001975	王辉	男	88320701
001976	李建	女	88320701
001977	程淡	男	88320701

表 8.6 借阅表数据

读者号	书号	借出日期	归还日期
001973	TP31.132/ZG1	2006-03-27	2006-05-28
001973	TP316/ZW6	2006-03-27	2006-05-28
001975	TP31.132/ZG1	2006-03-27	2006-05-28
001974	TP31.13/CM3	2006-03-27	2006-05-28
001975	TP316/ZW6	2006-03-27	2006-05-28
001974	TP31.132/ZG1	2006-03-28	2006-05-28

试完成以下各题。

- (1) 用 SQL 语句创建图书馆数据库。
- (2) 用 SQL 语句创建上述 3 个表。
- (3) 基于图书馆数据库的 3 个表，用 SQL 语言完成以下操作：
 - 1) 给图书表增加一列“ISBN”数据类型为 CHAR(10)。
 - 2) 为刚添加的 ISBN 列增加缺省值约束，约束名为 ISBNDEF，缺省值为‘7111085949’。
 - 3) 为读者表中‘办公电话’一列添加一个 CHECK 约束，要求电

话号码的前五位是‘88320’约束名为 CHECKDEF。

- 4) 删除图书表中 ISBN 列增加的缺省值约束。
 - 5) 删除读者表中的“办公电话”列的 CHECK 约束。
 - 6) 删除图书表中新增的列 ISBN。
- (4) 基于图书馆数据库的 3 个表，用 SQL 语言完成以下数据更新操作：
- 1) 向读者表中加入一个新读者，该读者的信息为：’’001978’,’王小平’,’男’,’88320701’’
 - 2) 向借阅表插入一个借阅记录，表示读者‘王小平’借阅了一本书，图书号为‘TP316/ZW6’，借出日期为当天的日期，归还日期为空值。
 - 3) 读者‘王小平’在借出上述图书 10 天之后归还该书。
 - 4) 当读者‘王小平’按期归还图书时，删除上述借阅记录。
- (5) 针对以上 3 个表，用 SQL 语言完成以下单表查询。
- 1) 查询全体图书的图书号、书名、作者、出版社和单价。
 - 2) 查询全体图书的信息，其中单价打 8 折，并设置该列的别名为‘打折价’。
 - 3) 显示所有借阅者的读者号，并去掉重复行。
 - 4) 查询所有单价在 20~30 元之间的图书信息。
 - 5) 查询所有单价不在 20~30 元之间的图书信息。
 - 6) 查询机械工业出版社、科学出版社、人民邮电出版社的图书信息。
 - 7) 查询不是机械工业出版社、也不是科学出版社出版的图书信息。
 - 8) 查找姓名的第二个字符是‘建’并且只有两个字符的读者的读者号及姓名。
 - 9) 查找姓名以‘王’开头的读者的读者号及姓名。
 - 10) 查找姓名以‘王’、‘张’或‘李’开头的读者的读者号及姓名。
 - 11) 查找姓名不是以‘王’、‘张’或‘李’开头的读者的读者号及姓名。
 - 12) 查询无归还日期的借阅信息。
 - 13) 查询有归还日期的借阅信息。
 - 14) 查询单价在 20 元以上、30 元以下的机械工业出版社出版的图书名及单价。
 - 15) 查询机械工业出版社或科学出版社出版的图书名、出版社及单价。

- 16) 查询读者的总人数。
 - 17) 查询借阅了图书的读者的总人数。
 - 18) 查询机械工业出版社图书的平均价格、最高价格、最低价格。
 - 19) 查询借阅图书超过 2 本的读者号、总本数。并按借阅本数值从大到小排序。
- (6) 针对以上三个表,用 SQL 语言完成以下各项多表连接查询。
- 1) 查询读者的基本信息及其借阅情况。
 - 2) 查询读者的读者号、姓名、借阅的图书名、借出日期及归还日期。
 - 3) 查询借阅了机械工业出版社出版,并且书名中包含‘数据库’三个字的图书的读者,并显示读者号、姓名、书名、出版社、借出日期及归还日期。
 - 4) 查询至少借阅过 1 本机械工业出版社出版的图书的读者的读者号、姓名、书名及借阅本数,并按借阅本数多少降序排列。
 - 5) 查询与‘王小平’的办公电话相同的读者的姓名。
 - 6) 查询办公电话为‘88320701’的所有读者的借阅情况,要求包括借阅了图书的读者和没有借阅的读者,显示他们的读者号、姓名、书名及借阅日期。
- (7) 针对以上 3 个表,用 SQL 语言完成以下各项子查询、组合查询:
- 1) 查询所有单价小于平均单价的书号、书名及出版社。
 - 2) 查询‘科学出版社’的图书中单价比‘机械工业出版社’最高单价还高的图书书名及单价。
 - 3) 查询‘科学出版社’的图书中单价比‘机械工业出版社’最低单价高的图书书名及单价。
 - 4) 查询已被借阅过并已归还的图书信息。
 - 5) 查询从未被借阅过的图书信息。
 - 6) 查询正在借阅的图书信息。
 - 7) 查询或者借阅了机械工业出版社出版的书名中含有‘数据库’三个字的图书,或者借阅了科学出版社出版的书名中含有‘数据库’三个字的图书的读者姓名及书名。
 - 8) 查询借阅了机械工业出版社出版的书名中含有‘数据库’的三个字的图书,并且也借阅了科学出版社出版的书名中含有‘数据库’三个字的图书的读者姓名及书名。
 - 9) 查询借阅了机械工业出版社出版的书名中含有‘数据库’三个字的图书,但没有借阅科学出版社出版的书名中含有‘数

据库’三个字的图书的读者姓名及书名。

第二部分 综合设计实验

实验一 课程成绩管理数据库

一、实验目的

1. 理解数据库系统的三级模式结构与两级映像机制；
2. 体会数据库的数据独立性特征；
3. 掌握 ER 模型的使用；
4. 掌握将 ER 模型转换为关系模型的方法；
5. 了解数据库设计的完整过程；
6. 掌握数据库的建立和重构方法；
7. 掌握数据添加、数据删除和数据更新的方法；
8. 掌握各种数据查询方法；
9. 了解数据库的安全性控制机制。

二、实验内容`

1. 课程成绩管理的业务要求
 - 1) 与课程有关的信息有：课程编号、课程名称、学分；
 - 2) 与学生有关的信息有：学号、姓名、班级；
 - 3) 与教师有关的信息有：工作证编号、姓名、所在学院；
 - 4) 与班级有关的信息有：班级名称；
 - 5) 一名学生属于一个班级，一个班级有多名学生；
 - 6) 一个班级可以开设多门课程，一门课程可能为多个班级开设；
 - 7) 一名学生可以选修为其所在班级开设的多门课程，为一个班级开设的一门课程可以有多名学生选修；
 - 8) 一名教师可以主讲为多个班级开设的多门课程，一门课程

可以有多名教师为多个班级主讲，为一个班级开设的一门课程只能由一名教师主讲；

- 9) 一名学生选修的一门课程只能有一个成绩，这个成绩只能由主讲该生所在班级该课程的教师评定；
- 10) 对各种数据的格式要求如表 9.1 所示；
- 11) 开设 3 门课程，具体信息如 表 9.2 所示；
- 12) 共有 6 名学生，具体信息如表 9.3 所示；
- 13) 共有 5 名教师，具体信息如表 9.4 所示；
- 14) 共有 2 个班级，具体信息如表 9.5 所示；
- 15) 教务员负责建立和维护课程、班级、教师和学生的档案，为各班开设课程，为各个班的各门课程分配主讲教师，按班级打印各门课程每个学生的成绩并计算平均成绩。各班课程开设及主讲教师分配情况如表 9.6 所示；

表 9.1 各种数据格式要求

数据	类型	长度	其他说明
课程编号	字符	6	课程的唯一标识
课程名称	字符	20	无
学分	正整数	2	无
学号	字符	8	学生的唯一标识
姓名	字符	20	无
班级名称	字符	16	班级的唯一标识
工作证编号	字符	6	教师的唯一标识
所在学院	字符	20	无
成绩	非负整数	3	不大于 100

表 9.2 课程数据表

课程编号	课程名称	学分
050304	计算机科学导引	4
120509	建筑力学	6
910201	世界科技史	3

表 9.3 学生数据表

学号	姓名	班级
40533123	诸葛飞扬	信息 0519
40533099	雷铭	信息 0519
40533103	贺舞	信息 0519
40525009	卫小虎	建筑 0501
40525011	邵剑青	建筑 0501
40525024	欧阳雨晨	建筑 0501

表 9.4 教师数据表

工作证编号	姓名	所在学院
032011	向华泽	信息科学与技术学院
032036	梅鹤	信息科学与技术学院
011221	张天啸	文法学院
011224	秦放	文法学院
021121	萧雁鸣	建筑科学与工程学院

表 9.5 班级数据表

班级名称
信息 0519
建筑 0501

表 9.6 各班课程开设及主讲教师分配情况

班级	课程	主讲教师
信息 0519	计算机科学导引	向华泽
信息 0519	世界科技史	张天啸
建筑 0501	计算机科学导引	梅鹤
建筑 0501	建筑力学	萧雁鸣
建筑 0501	世界科技史	秦放

- 16) 学生可以选课和查看成绩；
- 17) 教师负责给出成绩、更正成绩和按班级打印各门课程每个学生的成绩并计算平均成绩；
- 18) 学生选课情况如表 9.7 所示；

19) 学生各门课程成绩如表 9.8 所示;

表 9.7 学生选课情况

学生	课程	班级
诸葛飞扬	计算机科学导引	信息 0519
诸葛飞扬	世界科技史	信息 0519
雷铭	计算机科学导引	信息 0519
雷铭	世界科技史	信息 0519
贺舞	计算机科学导引	信息 0519
贺舞	世界科技史	信息 0519
卫小虎	计算机科学导引	建筑 0501
卫小虎	建筑力学	建筑 0501
卫小虎	世界科技史	建筑 0501
邵剑青	建筑力学	建筑 0501
邵剑青	世界科技史	建筑 0501
欧阳雨晨	计算机科学导引	建筑 0501
欧阳雨晨	建筑力学	建筑 0501

表 9.8 学生各门课程成绩

学生	课程	成绩
诸葛飞扬	计算机科学导引	85
诸葛飞扬	世界科技史	77
雷铭	计算机科学导引	69
雷铭	世界科技史	88
贺舞	计算机科学导引	75
贺舞	世界科技史	89
卫小虎	计算机科学导引	84
卫小虎	建筑力学	90
卫小虎	世界科技史	79
邵剑青	建筑力学	80
邵剑青	世界科技史	72
欧阳雨晨	计算机科学导引	81
欧阳雨晨	建筑力学	83

2. 实验任务

- 1) 根据业务要求建立数据库
- 2) 根据业务要求建立该数据库的用户
- 3) 根据业务要求以不同的用户身份使用该数据库

三、实验步骤

1. 使用 ER 模型进行概念设计。
2. 将得到的 ER 模型转换成关系模型。
3. 定义数据库模式(全局模式)。
4. 定义数据库外模式(子模式)。
5. 定义数据库用户：
 - 1) 教务员；
 - 2) 学生；
 - 3) 教师。
6. 以不同用户的身份使用数据库：
 - 1) 以教务员的身份输入课程、学生、教师的数据，并为课程分配教师；
 - 2) 以学生的身份选课；
 - 3) 以教师的身份给出成绩；
 - 4) 以学生的身份查看成绩；
 - 5) 以教师的身份更改成绩；
 - 6) 以教务员的身份更改成绩；
 - 7) 以教师的身份打印成绩单；
 - 8) 以教务员的身份打印成绩单。
7. 重新分配用户权限，收回教师的更改成绩权限，分配给教务员更改成绩权限。
8. 以不同用户的身份使用数据库：
 - 1) 以教师的身份更改成绩；
 - 2) 以教务员的身份更改成绩；
 - 3) 以教师的身份打印成绩单；
 - 4) 以教务员的身份打印成绩单。
9. 修改数据库模式，在教师信息中增加“职称”一项(字符类型，长度为 6)，并且删除“所在学院”一项。
10. 以不同的身份使用数据库：

- 1) 以学生的身份查看成绩；
- 2) 以教师的身份打印成绩单；
- 3) 以教务员的身份增加一名教师。

四、思考题

1. 数据库系统的三级模式结构与两级映像机制在 SQL server 2000 中是如何实现的？
2. 本实验中如何发挥数据库的数据独立性优点？
3. 在数据库设计过程中 ER 模型起到什么作用？
4. 本实验中数据库的安全性控制机制如何发挥作用？

五、实验报告内容及要求

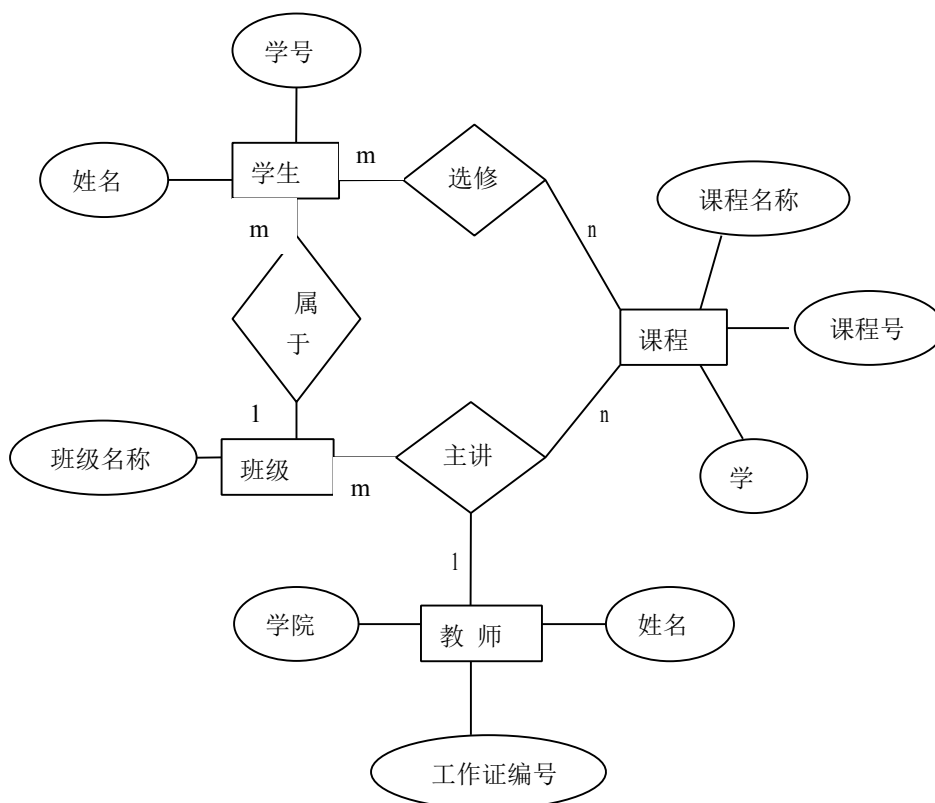
1. 本实验的数据库设计及建立的完整过程及各环节的产品；
2. 本实验所建立的数据库的三级模式结构与两级映像机制及实现；
3. 本实验所建立的数据库的数据独立性分析；
4. 本实验所建立的数据库的安全性控制机制；
5. 实验结论。

附录一：实验报告格式

1. 实验报告要求用 A4 纸张；
2. 一级标题使用黑体 4 号字，二级标题使用黑体小 4 号字，三级标题使用加黑宋体小 4 号字，正文使用宋体小 4 号字，不得使用斜体字，页面行距取 1.5 倍；
3. 所有的图表均要注明图号或表号。

附录二：实验参考文档

1. 课程成绩管理数据库系统的 ER 模型如附图 1 所示。



附图 1 课程成绩管理数据库系统的 ER 模型

2. 关系模型

实体“学生”的关系模式：学生（学号，姓名，班级），该关系模式包含联系“属于”所对应的关系模式。

实体“课程”的关系模式：课程（课程号，课程名称，学分）。

实体“教师”的关系模式：教师（工作证编号，姓名，学院）。

联系“选修”的关系模式：选修（学号，课程号，分数）。

联系“主讲”的关系模式：主讲（工作证编号，课程号，班级）。

文档中各模式的英文表示分别为：

Student(Sno,Sname,Ssex,Class)；

```
Course(Cno,Cname,Cridet) ;
Teacher(Tno,Tname,Tdept) ;
Enrollment(Sno,Cno,Grade) ;
TeachCourse(Tno,Cno,Class).
```

3. 数据库模式

1) 创建 TeachManagement 的脚本文件

```
USE master
GO
CREATE DATABASE TeachManagement
ON
( NAME = TeachManagement_dat,
  FILENAME = 'd:\program files\microsoft sql server\mssql\data\
TeachManagement_dat.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = 'TeachManagement_log',
  FILENAME = 'd:\program files\microsoft sql server\mssql\data\
TeachManagement_log.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
GO
```

2) 创建 TeachManagement 中各个表的脚本文件

```
USE TeachManagement
if exists (select * from dbo.sysobjects where id = object_id
(N'[dbo].[FK_Enrollment_Course]') and OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [dbo].[Enrollment] DROP CONSTRAINT
FK_Enrollment_Course
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id
(N'[dbo].[FK_Enrollment_Student]') and OBJECTPROPERTY(id,
N'IsForeignKey') = 1)
ALTER TABLE [dbo].[Enrollment] DROP CONSTRAINT
```

FK_Enrollment_Student
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Course]')
and OBJECTPROPERTY(id, N'IsUSERTable') = 1)
drop table [dbo].[Course]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Enrollment]') and OBJECTPROPERTY(id,
N'IsUSERTable') = 1)
drop table [dbo].[Enrollment]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Student]') and OBJECTPROPERTY(id, N'IsUSERTable')
= 1)
drop table [dbo].[Student]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[TeachCourse]') and OBJECTPROPERTY(id,
N'IsUSERTable') = 1)
drop table [dbo].[TeachCourse]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Teacher]') and OBJECTPROPERTY(id, N'IsUSERTable')
= 1)
drop table [dbo].[Teacher]
GO

CREATE TABLE [dbo].[Course] (
[Cno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL PRIMARY
KEY ,
[Cname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
[Credit] [int] NULL

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Enrollment] (
[Sno] [char] (8) COLLATE Chinese_PRC_CI_AS NOT NULL,
[Cno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,
[Grade] [int] NULL,
PRIMARY KEY ([Sno], [Cno])
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Student] (
[Sno] [char] (8) COLLATE Chinese_PRC_CI_AS NOT NULL PRIMARY
KEY,
[Sname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
[Class] [char] (16) COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[TeachCourse] (
[Class] [char] (16) COLLATE Chinese_PRC_CI_AS NOT NULL,
[Cno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL,
PRIMARY KEY ([Class], [Cno])
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Teacher] (
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL PRIMARY
KEY ,
[Tname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
[Tdept] [char] (20) COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY]
GO
```

4. 数据库外模式

创建 **Grade** 视图，供学生和教师查询。
成绩（班级，姓号，课程名称，学分，成绩）

Grade(Class,Sname,Cname,Credit,Grade)

创建 Grade 视图的脚本文件如下：

```
CREATE VIEW dbo.Grade
AS
SELECT  dbo.Student.Class,  dbo.Student.Sname,  dbo.Course.Cname,
        dbo.Course.Credit,
        dbo.Enrollment.Grade
FROM  dbo.Enrollment INNER JOIN
        dbo.Course ON dbo.Enrollment.Cno = dbo.Course.Cno INNER JOIN
        dbo.Student ON dbo.Enrollment.Sno = dbo.Student.Sno
```

5. 定义数据库用户：

1) 教务员 (Dean)

定义数据库用户方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

为该用户分配权限的脚本文件如下：

```
USE TeachManagement
GO
GRANT ALL PRIVILEGES ON Student, Course, Teacher, TeachCourse TO
Dean
GO
```

2) 学生 (Student)

定义数据库用户方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

为该用户分配权限的脚本文件如下：

```
USE TeachManagement
GO
GRANT SELECT, INSERT(Sno,Cno), UPDATE(Sno,Cno) ON Enrollment
TO Student
GO
```

3) 教师 (Teacher)

定义数据库用户方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

为该用户分配权限的脚本文件如下：

```
USE TeachManagement
GO
GRANT SELECT, UPDATE(Grade) ON Enrollment TO Teacher
GO
```

6. 以不同用户的身份使用数据库:

- 1) 以教务员的身份输入课程、学生、教师的数据, 并为课程分配教师;

在查询分析器中以教务员身份登录, 进行下面操作。以不同身份登录的方法见第一部分第5章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Course]')
and OBJECTPROPERTY(id, N'IsUSErTable') = 1)
```

```
delete from [dbo].[Course]
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Student]') and OBJECTPROPERTY(id, N'IsUSErTable')
= 1)
```

```
delete from [dbo].[Student]
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Teacher]') and OBJECTPROPERTY(id, N'IsUSErTable')
= 1)
```

```
delete from [dbo].[Teacher]
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[TeachCourse]') and OBJECTPROPERTY(id,
N'IsUSErTable') = 1)
```

```
delete from [dbo].[TeachCourse]
```

```
GO
```

```
INSERT INTO Course (Cno,Cname,Credit) VALUES ('050304','计算机科学
导引','4')
```

```
INSERT INTO Course (Cno,Cname,Credit) VALUES ('120509','建筑力学
','6')
```

```
INSERT INTO Course (Cno,Cname,Credit) VALUES ('910201','世界科技史
','3')
```

```
GO
```

```
INSERT INTO Student(Sno,Sname,Class)VALUES('40533123','诸葛飞扬','
信息 0519')
```

```
INSERT INTO Student(Sno,Sname,Class)VALUES('40533099','雷铭','信息
0519')
```

```
INSERT INTO Student(Sno,Sname,Class)VALUES('40533103','贺舞','信息
```

0519')

INSERT INTO Student(Sno,Sname,Class)VALUES('40525009','卫小虎','建筑 0501')

INSERT INTO Student(Sno,Sname,Class)VALUES('40525011','邵剑青','建筑 0501')

INSERT INTO Student(Sno,Sname,Class)VALUES('40525024','欧阳雨晨','建筑 0501')

GO

INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('032011','向华泽','信息科学与技术学院')

INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('032036','梅鹤','信息科学与技术学院')

INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('011221','张天啸','文法学院')

INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('011224','秦放','文法学院')

INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('021121','萧雁鸣','建筑科学与工程学院')

GO

INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('建筑 0501','050304','032036')

INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('建筑 0501','120509','021121')

INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('建筑 0501','910201','011224')

INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('信息 0519','050304','032011')

INSERT INTO TeachCourse(Class,Cno,Tno)VALUES('信息 0519','910201','011221')

GO

2) 以学生的身份选课;

在查询分析器中以学生身份登录, 进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

USE TeachManagement

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Enrollment]') and OBJECTPROPERTY(id, N'IsUSERTable') = 1)

```
delete from [dbo].[Enrollment]
```

```
GO
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533123','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533123','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533099','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533099','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533103','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40533103','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525009','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525009','120509')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525009','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525011','120509')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525011','910201')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525024','050304')
```

```
INSERT INTO Enrollment(Sno,Cno)VALUES('40525024','120509')
```

```
GO
```

3) 以教师的身份给出成绩；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
GO
```

```
UPDATE Enrollment SET Grade='85' WHERE Sno='40533123' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='77' WHERE Sno='40533123' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='69' WHERE Sno='40533099' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='88' WHERE Sno='40533099' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='75' WHERE Sno='40533103' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='89' WHERE Sno='40533103' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='84' WHERE Sno='40525009' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='90' WHERE Sno='40525009' AND  
Cno='120509'
```



```
UPDATE Enrollment SET Grade='79' WHERE Sno='40525009' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='80' WHERE Sno='40525011' AND  
Cno='120509'
```

```
UPDATE Enrollment SET Grade='72' WHERE Sno='40525011' AND  
Cno='910201'
```

```
UPDATE Enrollment SET Grade='81' WHERE Sno='40525024' AND  
Cno='050304'
```

```
UPDATE Enrollment SET Grade='83' WHERE Sno='40525024' AND  
Cno='120509'
```

```
GO
```

4) 以学生的身份查看成绩；

在查询分析器中以学生身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
SELECT * FROM Grade
```

```
GO
```

5) 以教师的身份更改成绩；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Enrollment SET Grade='90' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

6) 以教务员的身份更改成绩；

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Enrollment SET Grade='95' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

7) 以教师的身份打印成绩单；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

这里使用了游标，可以依次打印各个班级的成绩，并有一些相应的格式上的设置。当然也可以不使用游标，分别对各班进行打印。

在“查询分析器”下，将查询结果的显示模式设置为“文本显

示结果”（设置方法见第一部分第4章），运行如下脚本：

```
USE TeachManagement
GO
SET NOCOUNT ON
DECLARE @Class char(16), @Average float
PRINT '----- 各班成绩打印 -----'
PRINT "
DECLARE print_cursor CURSOR FOR
SELECT DISTINCT Class FROM Student
ORDER BY Class
OPEN print_cursor
FETCH NEXT FROM print_cursor
INTO @Class
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT '-----'+@Class+'班成绩-----'
    -- Get average
    SELECT @Average=AVG(Grade) FROM Enrollment, Student WHERE
(Enrollment.Sno = Student.Sno) AND (Class=@Class)
    --Print grade and average
    SELECT * FROM Grade WHERE Class=@Class
    PRINT @Class+'班平均成绩:'+STR(@Average)
PRINT "
    -- Get the next Class
    FETCH NEXT FROM print_cursor
    INTO @Class
END
CLOSE print_cursor
DEALLOCATE print_cursor
GO
```

8) 以教务员的身份打印成绩单。

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第5章 SQL Server 2000 用户与权限管理。

在“查询分析器”下，将查询结果的显示模式设置为“文本显示结果”（设置方法见第一部分第4章），运行第7)小题中的成绩打印脚本。

7. 重新分配用户权限，收回教师的更改成绩权限，分配给教务员更改成绩

权限。

在查询分析器中以管理员(sa)身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
REVOKE UPDATE(Grade) ON TABLE Grade FROM Teacher
```

```
GRANT UPDATE(Grade) ON Grade TO Dean
```

8. 以不同用户的身份使用数据库：

1) 以教师的身份更改成绩；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Grade SET Grade='90' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

2) 以教务员的身份更改成绩；

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
UPDATE Grade SET Grade='95' WHERE Sno='40533123' AND  
Cno='050304'
```

```
GO
```

3) 以教师的身份打印成绩单；

在查询分析器中以教师身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在“查询分析器”下，将查询结果的显示模式设置为“文本显示结果”（设置方法见第一部分第 4 章），运行 6 题第 7) 小题中的成绩打印脚本。

4) 以教务员的身份打印成绩单。

在查询分析器中以教务员身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在“查询分析器”下，将查询结果的显示模式设置为“文本显示结果”（设置方法见第一部分第 4 章），运行 6 题第 7) 小题中的成绩打印脚本。

9. 修改数据库模式，在教师信息中增加“职称”一项（字符类型，长度为 6），并且删除“所在学院”一项。

在查询分析器中以管理员(sa)身份登录，进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在教师信息中增加“职称”一项的 SQL 语句如下：

```
USE TeachManagement
```

```
ALTER TABLE Teacher ADD Position CHAR(6)
```

```
GO
```

在 SQL 中，由于不能删除表的某一属性列，采用下面的替代方案，即先将所需列复制至备份表中，再删除原表。

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[TeacherTemp]') and OBJECTPROPERTY(id,  
N'IsUserTable') = 1)
```

```
drop table [dbo].[TeacherTemp]
```

```
GO
```

```
CREATE TABLE [dbo].[TeacherTemp] (  
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,  
[Tname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,  
[Position] [char] (6) COLLATE Chinese_PRC_CI_AS NULL  
) ON [PRIMARY]
```

```
GO
```

```
INSERT INTO TeacherTemp(Tno,Tname) SELECT Tno,Tname FROM  
Teacher
```

```
GO
```

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[Teacher]') and OBJECTPROPERTY(id, N'IsUserTable')  
= 1)
```

```
drop table [dbo].[Teacher]
```

```
GO
```

```
CREATE TABLE [dbo].[Teacher] (  
[Tno] [char] (6) COLLATE Chinese_PRC_CI_AS NOT NULL ,  
[Tname] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,  
[Position] [char] (6) COLLATE Chinese_PRC_CI_AS NULL  
) ON [PRIMARY]
```

```
GO
```

```
INSERT INTO Teacher SELECT Tno,Tname FROM TeacherTemp
```

```
GO
```

```
if exists (select * from dbo.sysobjects where id =  
object_id(N'[dbo].[TeacherTemp]') and OBJECTPROPERTY(id,  
N'IsUserTable') = 1)
```

```
drop table [dbo].[TeacherTemp]
```

GO

10. 以不同的身份使用数据库:

1) 以学生的身份查看成绩;

在查询分析器中以学生身份登录, 进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
SELECT * FROM Grade
```

GO

2) 以教师的身份打印成绩单;

在查询分析器中以教师身份登录, 进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

在“查询分析器”下, 将查询结果的显示模式设置为“文本显示结果”(设置方法见第一部分第 4 章), 运行 6 题第 7) 小题中的成绩打印脚本。

3) 以教务员的身份增加一名教师。

在查询分析器中以教务员身份登录, 进行下面操作。以不同身份登录的方法见第一部分第 5 章 SQL Server 2000 用户与权限管理。

```
USE TeachManagement
```

```
INSERT INTO Teacher(Tno,Tname,Tdept)VALUES('032005','李新','信息科学与技术学院')
```

GO