

引 言

一、SQL Server 2008 的体系结构

微软的 SQL Server 2008 是微软公司在 2008 年正式发布的一个客户/服务器方式的关系数据库管理系统。它是一个功能全面的整合的数据平台，提供的服务器组件包括数据库引擎（Database Engine），用于存储、处理和保护数据的核心服务；分析服务（Analysis Service），为商业智能应用程序提供联机分析处理和数据挖掘功能；集成服务（Integration Service），用于清理、聚合、合并、复制数据的转换以及管理 SSIS 包；报表服务（Reporting Services），用来创建和管理包含关系数据源和多维数据源中的数据的表格、矩阵、图形和自由格式的报表。

SQL Server 2008 的管理工具主要包括：

(1) SQL Server Management Studio，用于编辑和执行查询、以及启动标准向导任务；它主要有两个工具：对象管理器和查询分析器。对象管理器用图形化的方式管理数据库；查询分析器可以运行 Transact SQL 脚本。

(2) SQL Server Profiler，提供用于监视 SQL Server 数据库引擎实例或 Analysis Services 实例的图形用户界面。

(3) 数据库引擎优化顾问，可以协助创建索引、索引视图和分区的最佳组合。

(4) SQL Server Business Intelligence Development Studio，用于 Analysis Server、Integration Services 和 Reporting Services 项目在内的商业解决方案的集成开发环境。

(5) Reporting Services 配置管理器，提供报表服务器配置的统一的查看、设置和管理方式。

(6) SQL Server 安装中心，安装、升级到或更改 SQL Server 2008 实例中的组件。

二、SQL Server 2008 语言的基本知识

1、数据类型

在计算机中数据有两种特征：类型和长度。数据类型决定了数据的取值、范围和存储格式。数据类型可以是系统提供的系统数据类型，也可以是用户自定义的数据类型。

在 SQL Server 2008 提供了丰富的系统数据类型，列在下表中。

SQL Server 2008 提供的数据类型分类

| 分 类 | 数 据 类 型 |
|-------------|--|
| 整数型 | bigint, int, smallint, tinyint |
| 精确数值型 | decimal, numeric |
| 浮点型 | float, real |
| 货币型 | money, smallmoney |
| 位型 | Bit |
| 字符型 | char, varchar |
| Unicode 字符型 | nchar, nvarchar |
| 文本型 | text, ntext |
| 二进制型 | binary, varbinary |
| 日期时间型 | datetime, smalldatetime, date, time, datetime2, datetimeoffset |
| 时间戳型 | timestamp |
| 图像型 | image |
| 其他 | cursor, sql_variant, table, uniqueidentifier, xml, hierarchyid |

下面分类讲述各种数据类型：

(1) 整数型：用于存储整数数据，它们的表示数范围逐渐缩小。

(2) 精确数值型：由整数部分和小数部分构成，其所有的数字都是有效位，能够以完整的精度存储十进制数。

(3) 浮点型：用于存储十进制小数浮点数值的数据。这种类型不能提供精确标示数据的精度，使用该类型来存储数值时，可能会损失一些精度，所以它可用于处理取值范围大且对精确度要求不太高的数值量。

(4) 货币数据类型：用于存储货币值，在使用货币数据类型时应在数据前加上货币符号，系统才能辨识其为哪国的货币。如“\$”表示美元，如果不加货币符号则默认为“¥”。money使用8个字节存储有4位小数的十进制数；smallmoney使用4个字节存储4位小数的十进制。

(5) 位型：相当于其他语言中的逻辑型数据，它只存储0和1，长度为1字节。当为bit类型数据赋0时，其值为0，而赋非0时，其值为1。

(6) 字符型：字符数据类型是使用最多的数据类型，它可以用来存储各种字母、数字、符号、特殊符号。一般情况下使用字符类型数据时须在其前后加上单引号。

(7) Unicode字符型：用于支持国际上非英语语种的字符数据的存储和处理。SQL Server的Unicode字符型可以存储Unicode标准字符集定义的各种字符。

(8) 文本型：这类数据类型用于存储大量的字符。如较长的备注、日志信息、一个人的简历等。

(9) 二进制型：用于存储二进制数据，其定义形式为binary(n)，n表示数据的长度取值为1到8000。在使用时必须指定binary 类型数据的大小至少应为1个字节。varbinary(n)用于存储变长二进制数据。

(10) 日期时间类型：用于存储日期和时间的结合体。日期输入格式：2000-6-22。时间输入格式：在输入时间时必须按小时分钟秒毫秒的顺序来输入，在其间用冒号隔开。但可将毫秒部分用小数点。当使用12 小时制时用am 和pm分别指定时间是午前或午后。如3:5:7.2pm表示 下午3 时5 分7 秒200 毫秒。

(11) 时间戳型：timestamp类型是一个二进制格式数据，其值实际上反映了系统对该记录修改的相对顺序，一个表只能有一个timestamp列。每当对该表加入新行或修改已有行时，都由系统自动将一个计数器值加到该列。

(12) 图形数据类型：image数据类型用于存储大量的二进制数据（Binary Data）。通常用来存储图片、照片等。

(13) 其他数据类型：cursor：游标数据类型，用于创建游标变量或定义存储过程的输出参数；sql_variant：是一种存储SQL Server支持的各种数据类型（除text、ntext、image、timestamp和sql_variant外）；table：用于存储结果集的数据类型，结果集可以供后续处理；uniqueidentifier：是唯一标识符类型，系统将为这种类型的数据产生唯一标识值，它是一个16字节长的二进制数据；xml：用于存储xml文档和片段的一种类型；hierarchyid：是一种长度可变的系统数据类型，可使用hierarchyid标识层次结构中的位置。

2、变量

Transact-SQL 中可以使用两种变量一种是局部变量，另外一种是全球变量。

(1) 局部变量：是用户可自定义的变量它的作用范围仅在程序内部在程序中通常用来储存从表中查询到的数据，或当作程序执行过程中暂存变量使用局部变量。必须以@开头，而且必须先用DECLARE 命令说明后才可使用。其说明形式如下：

DECLARE @变量名 变量类型 [@变量名 变量类型]

其中变量类型可以是SQL Server 2000 支持的所有数据类型。

(2) 全局变量：是SQL Server 系统内部使用的变量。其作用范围并不局限于某一程序，而是任何程序均可随时调用。全局变量通常存储一些SQL Server 的配置设定值和效能统计数据。用户可在程序中用全局变量来测试系统的设定值或Transact-SQL 命令执行后的状态值。

3、注释符

在Transact-SQL 中可使用两类注释符。ANSI 标准的注释符-- 用于单行注释；与C 语言相同的程序注释符号即/*.....*/。/* 用于注释文字的开头，*/用于注释文字的结尾。可在程序中标识多行文字为注释。

4、运算符

(1) 算术运算符：包括 +（加），—（减乘），/（除），%（取余）

(2) 比较运算符：包括 >（大于），<（小于），=（等于），>=（大于等于），<=（小于等于），!=（不等于），!>（不大于），!<（不小于）。其中 != !> !< 不是ANSI 标准的运算符。

(3) 逻辑运算符：包括 AND（与），OR（或），NOT（非）。

(4) 连接运算符：连接运算符 + 用于连接两个或两个以上的字符或二进制串、列名或者串和列的混合体。将一个串加入到另一个串的末尾。其语法如下：

<expression1>+<expression2>

例：

```
select '月薪最高的员工是'+ e_name + ' 月薪为'+ convert(varchar(10)) e_wage
from employee
where e_wage=select max(e_wage)
              from employee
```

运行结果如下：

月薪最高的员工是:张三 月薪为:8000.00

实验一 数据库和关系表

一、实验目的

- 1、熟悉对象资源管理器的基本环境
- 2、掌握对象资源管理器的操作方法，掌握数据库和表的基本操作
- 3、熟悉应用 SQL 语言创建数据库、关系表以及修改表结构

二、预备知识

1、数据库

SQL 术语中，数据库是一个容器，包含了相关的基本表、视图、索引、存储过程以及其它对象。在创建这些对象之前，必须有首先创建一个存储它们的数据库。只有系统管理员或者被系统管理员授权的用户才能创建数据库。

当我们打开 SQL Server 系统的“对象资源管理器”时，“数据库”夹中已经存在四个系统数据库：

(1) master 数据库

master 数据库记录 SQL Server 系统的所有系统级别信息。这些系统信息主要有：

- 所有的登录账号
- 系统设置信息
- SQL Server 的初始化信息
- 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、事务日志文件的位置以及这些文件的其它信息。

(2) tempdb 数据库

tempdb 数据库保存所有的临时表、临时存储过程等临时对象。临时对象就是不属于任何数据库的数据对象。需要注意的是，当退出 SQL Server 时，用户在 tempdb 数据库中创建的所有对象都将被删除。每次 SQL Server 启动时，tempdb 数据库都将被重建，回复到系统设定的初始状态。因此，千万不要将 tempdb 数据库作为数据的最终存储位置。

(3) model 数据库

model 数据库是所有用户数据库和 tempdb 数据库的创建模板。当创建数据库时，系统会将 model 数据库中的内容复制到新建的数据库中去。由此可见，利用 model 数据库的模板特性，通过更改 model 数据库的设置，复制常用数据库对象，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。

(4) msdb 数据库

msdb 数据库存储计划信息以及备份和恢复相关的信息。SQL Server Agent 使用它来执行安排工作和警报、记录操作员等操作。主要包括如下几方面的信息：

- 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

Resource 数据库是只读数据库，它包含 SQL Server 2008 中的所有系统对象。SQL Server 系统对象在物理上持续存在于 Resource 数据库中，但在逻辑上，它们出现在每个数据库的 sys 架构中。

由于这些系统数据库都是用来存储系统信息的，所以通常是不允许修改的。

SQL Server 2008 还提供了两个实例数据库：pubs 和 Northwind。这两个实例数据库与系统数据库一样也存储在 Microsoft SQL Server 的数据库文件夹中。pubs 实例数据库存储了一个虚构的图书出版公司的基本情况。Northwind 实例数据库包含了一个名为 Northwind Traders 的虚拟食品进出口公司的销售数据。

这两个数据库可以作为 SQL Server 的学习工具，SQL Server Books Online 中的实例基本上都是基于这两个数据库来讲解的。与系统数据库不同的是，这两个数据库是可以更新的，如果改变了数据库的内容或不小心删除了实例数据库，想将其回复到初始状态，可以用 SQL Server Analyzer 运行“安装目录\MSSQL\Install”中的 Instpubs.sql 或 Instnwnd.sql 程序来重建 pubs 或 Northwind 数据库。

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。

数据库文件是存放数据库数据和数据库对象的文件。一个数据库可以有一个或多个数据库文件，一个数据库文件只属于一个数据库。当有多个数据库文件时，有一个文件被定义为主数据库文件 Primary Database File，扩展名为 mdf。它用来存储数据库的启动信息和部分或全部数据。一个数据库只能有一个主数据库文件，其它数据库文件被称为次数据库文件 Secondary Database File，扩展名为 ndf，用来存储主文件没存储的其它数据。

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 和 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

2、基本表

在关系数据库中，使用基本表存储数据。在一个 SQL Server 数据库中，可以创建多达两万亿个基本表。

SQL 语言中创建表的语句为 create table 语句，其格式如下：

```
create table <表名> (<列名><数据类型>[列级完整性约束]
[, <列名><数据类型>[列级完整性约束]]...
[, <表级完整性约束>];
```

注：

完整性约束条件被存入系统的数据字典中。约束条件涉及表的多个属性列，则定义在表级，否则定义在列级或表级。

3、实例描述

本书采用的实例是一个学生—课程数据库（Stu_Cou），该数据库中包括三个表：

- 学生表：Student（Sno, Sname, Ssex, Sbirth, Spre, Sclass, Sdept）
Student 由学号（Sno）、姓名（Sname）、性别（Ssex）、出生日期（Sbirth）、已修学分（Spre）、班级（Sclass）、所在系（Sdept）七个属性组成，其中 Sno 为主码。
- 课程表：Course（Cno, Cname, Ctype, Cpno, Ccredit）
Course 由课程号（Cno）、课程名（Cname）、课程类型（Ctype）、先修课号（Cpno）、学分（Ccredit）五个属性组成，其中 Cno 为主码。

- 学生选课表：SC (Sno, Cno, Grade)
由学号 (Sno)、课程号 (Cno)、成绩 (Grade) 三个属性组成，主码为 (Sno, Cno)。

三、实验示例

1、使用“管理工具 (SSMS)”

(1) 创建数据库

步骤 1：如图 1-1 所示，选中“数据库”后按右键在快捷菜单中选择“新建数据库”。

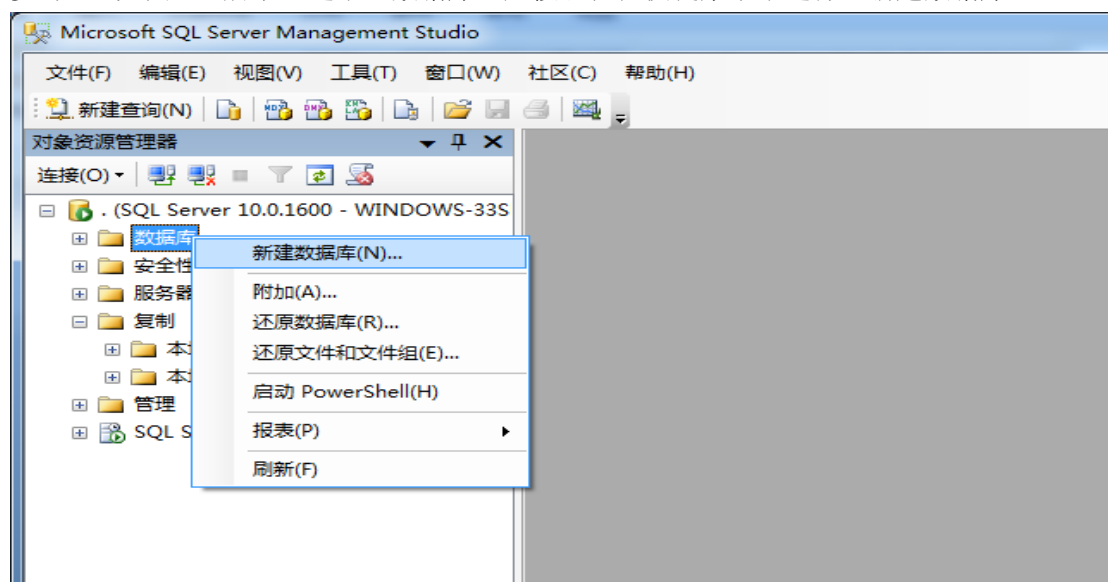


图 1-1 创建数据库（一）

步骤 2：如图 1-2 所示，在“名称”后的文本框中输入要创建的数据库名，例如“Stu_Cou”，单击确定按钮后，“Stu_Cou”数据库就创建成功了。同时数据库“Stu_Cou”相应的数据文件、日志文件都自动创建完成了。可以通过点击“数据文件”、“事务日志”部分来查看。

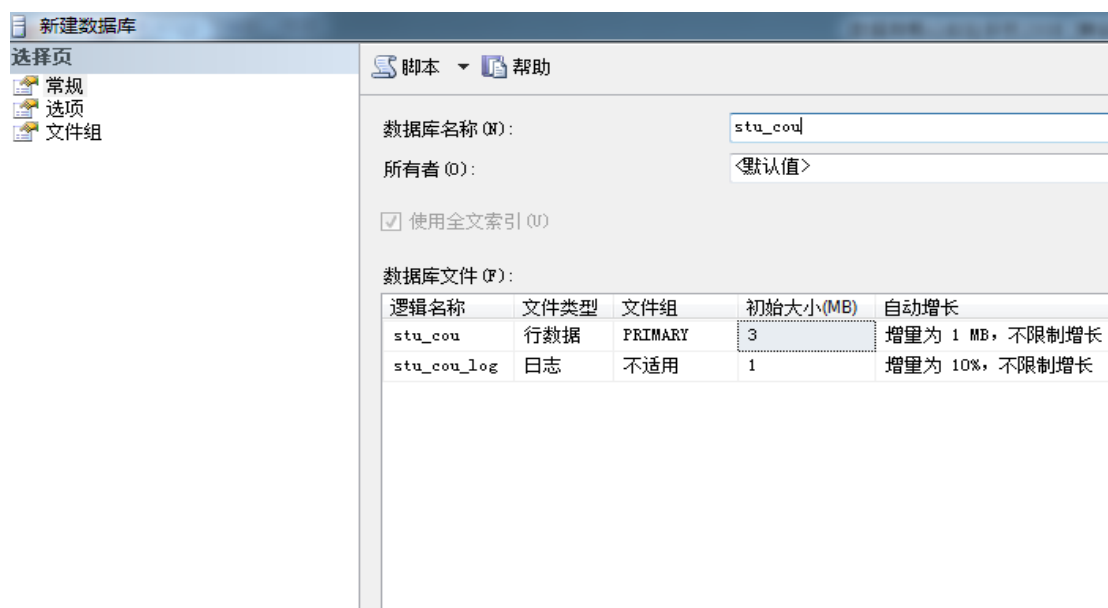


图 1-2 创建数据库（二）

另外当不需要数据库“Stu_Cou”时，就需要删除该数据库，删除数据库的操作很简单，

即选中数据库 “Stu_Cou”，然后右击 “删除” 即可。

(2) 创建数据库表

在数据库 Stu_Cou 中创建三个表，分别为 Student 表、Course 表、SC 表。这三个表中的示例数据及表结构如下：

Student 表（学生表）

| Sno | Sname | Ssex | Sbirth | Spre | Sclass | Sdept |
|-------|-------|------|-----------|------|--------|-------|
| 95001 | 李勇 | 男 | 1977-5-1 | 23 | 计 9501 | 计算机 |
| 95002 | 刘晨 | 女 | 1976-4-8 | 25 | 计 9502 | 计算机 |
| 95003 | 李鸣 | 女 | 1977-2-2 | 26 | 英 9501 | 英语 |
| 95004 | 王敏 | 女 | 1977-3-5 | 20 | 英 9501 | 英语 |
| 95016 | 金瑾 | 女 | 1978-11-1 | 23 | 化 9501 | 化学 |
| 95018 | 汪劲 | 男 | 1977-4-6 | 21 | 社 9502 | 社会 |
| 96001 | 张立 | 男 | 1978-6-2 | 18 | 数 9601 | 数学 |
| 96002 | 柳风 | 男 | 1979-8-18 | 16 | 数 9602 | 数学 |

Student 表结构：

| 列名 | 含义 | 类型 | 允许空 |
|--------|------|---------------|-----|
| Sno | 学号 | char(6) | × |
| Sname | 姓名 | varchar(10) | × |
| Ssex | 性别 | char(2) | √ |
| Sbirth | 出生日期 | datetime | √ |
| Spre | 已修学分 | Real | √ |
| Sclass | 班级 | char(6) | √ |
| Sdept | 所在系 | varchar(20) | √ |

Course 表（课程表）

| Cno | Cname | Ctype | Cpno | Ccredit |
|-----|-----------|-------|------|---------|
| 1 | 数据库_设计 | 必修 | 5 | 4 |
| 2 | 数学 | 必修 | | 2 |
| 3 | 信息系统 | 选修 | 1 | 4 |
| 4 | 操作系统 | 必修 | 6 | 3 |
| 5 | 数据_结构 | 必修 | 7 | 4 |
| 6 | 数据_处理 | 选修 | | 2 |
| 7 | PASCAL 语言 | 选修 | 6 | 4 |
| 8 | 概率统计 | 必修 | | 3 |
| 9 | 计算机网络 | 必修 | 1 | 4 |

Course 表结构

| 列名 | 含义 | 类型 | 允许空 |
|-------|------|-------------|-----|
| Cno | 课程号 | char(6) | × |
| Cname | 课程名称 | varchar(20) | × |

| | | | |
|---------|------|---------|---|
| Ctype | 课程类型 | char(8) | √ |
| Cpno | 先修课号 | char(6) | √ |
| Ccredit | 学分 | Int | √ |

SC 表（学生选课表）

| Sno | Cno | Grade |
|-------|-----|-------|
| 95001 | 1 | 92 |
| 95001 | 2 | 80 |
| 95002 | 1 | |
| 95002 | 2 | 71 |
| 95002 | 3 | 52 |
| 95016 | 2 | 88 |
| 95018 | 6 | 80 |
| 96001 | 1 | |
| 96002 | 8 | 82 |

SC 表结构

| 列名 | 含义 | 类型 | 是否允许空 |
|-------|-----|---------|-------|
| Sno | 学号 | char(6) | × |
| Cno | 课程号 | char(6) | × |
| Grade | 成绩 | int | √ |

如何应用“管理工具 SSMS”在数据库 Stu_Cou 中创建基本表的结构，以及如何在基本表中输入数据呢？下面以创建 Student 为例介绍创建的步骤，其它两个表的创建可以如法炮制，就不一一赘述了。大家自行创建。**切记建好的数据库自己保存**

步骤 1：如图 1-3 所示，选择“Stu_Cou”→ 表，然后右键在快捷菜单中选择“新建表”。

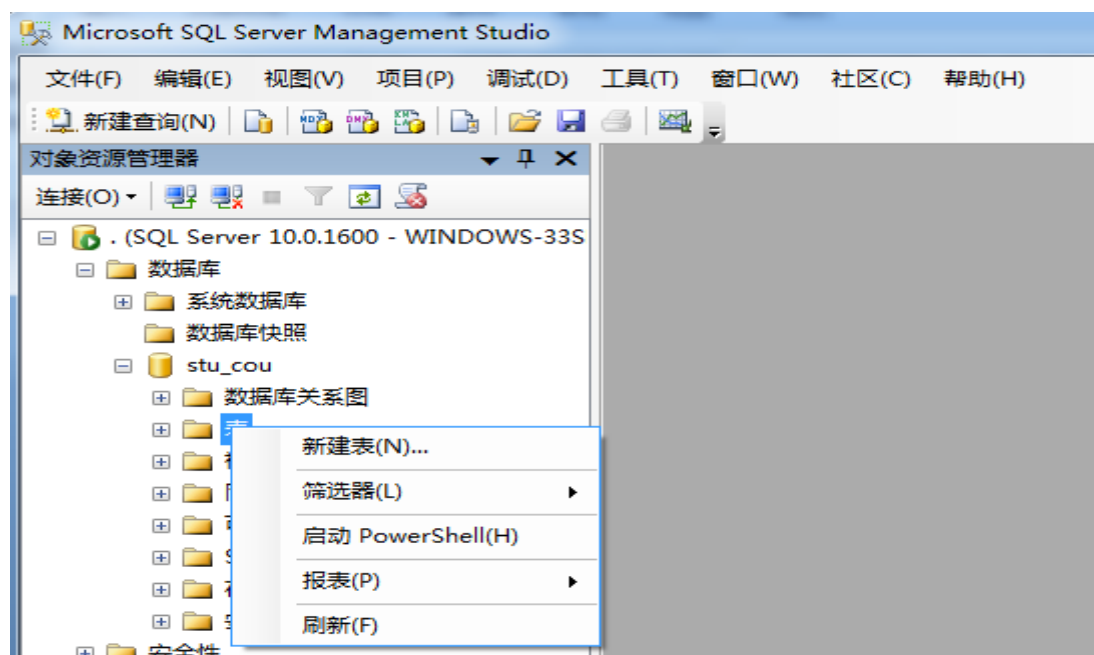


图 1-3 创建表 Student（一）

步骤 2：在图 1-4 的窗口中，按照输入表 Student 的各列列名、数据类型、长度和约束等，

然后在保存 Student 表。（设置主键点击图 1-4 中的”小钥匙”形状即可，如果确定外键，则右击该行，在出现窗体中对应关系确认即可）

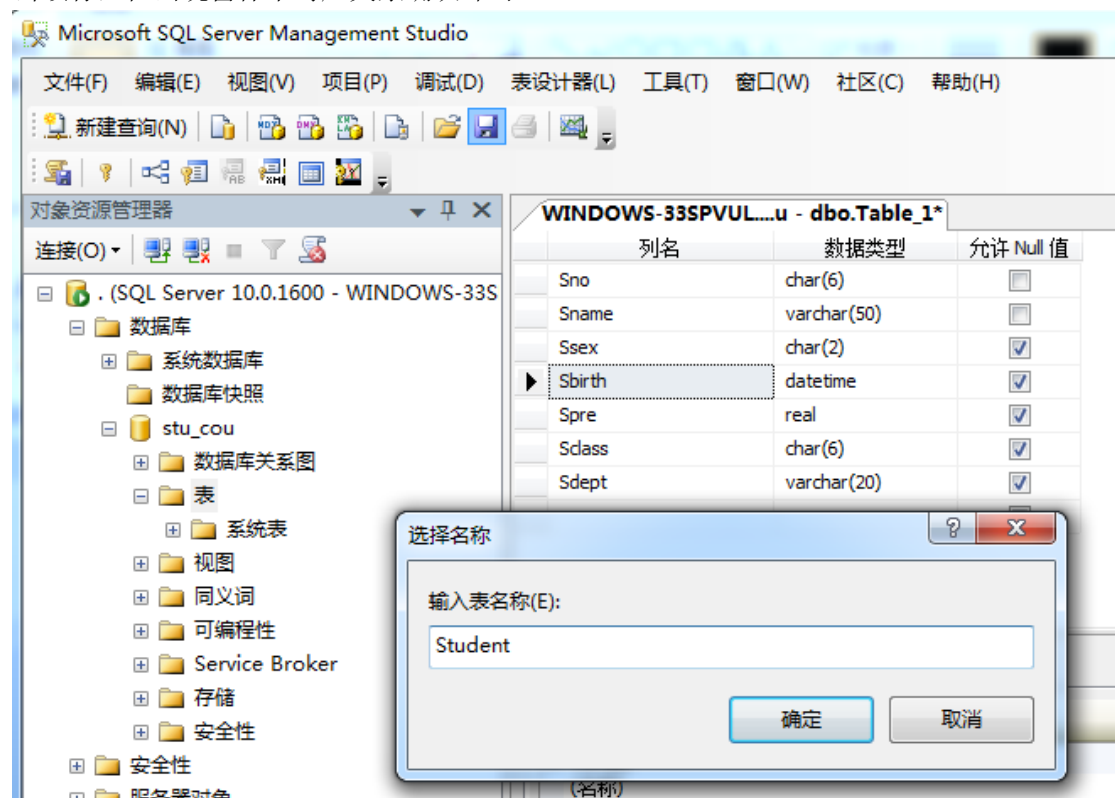


图 1-4 创建表 Student（二）

步骤 3：如图 1-5 所示，选中 Student 表，然后右键在快捷菜单中选择“打开表”——→“返回所有行”，最后在打开的窗口中输入表 Student 的全部数据即可。

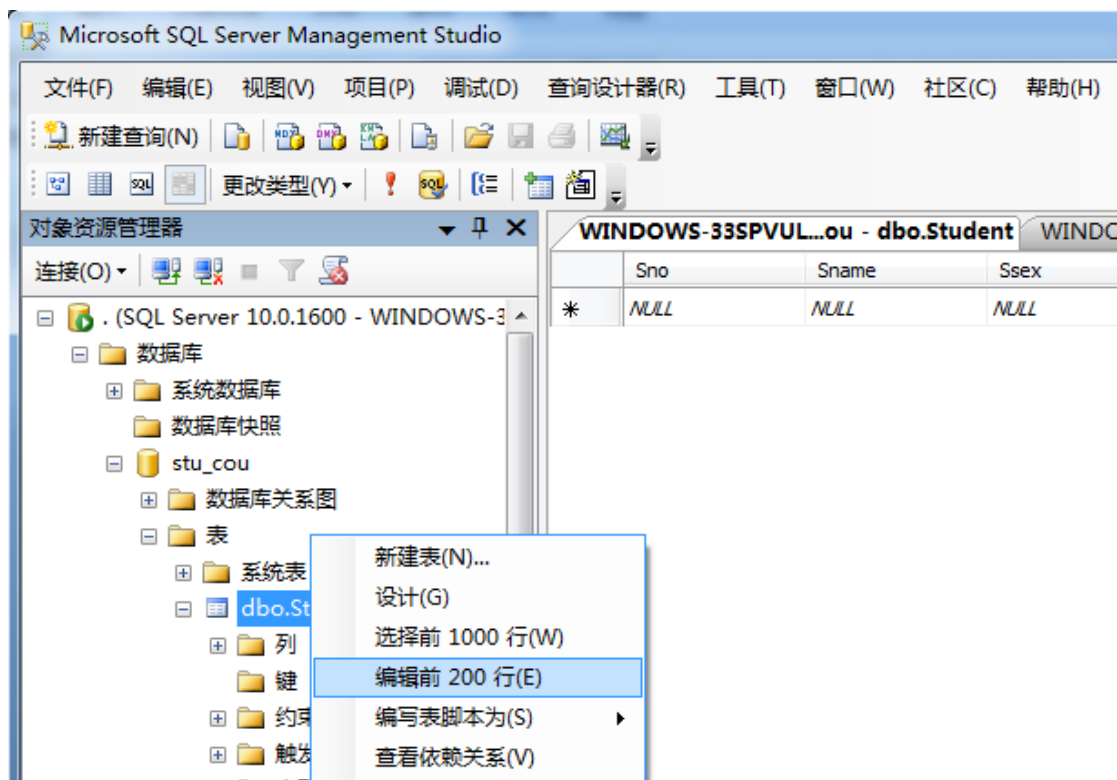


图 1-5 在 Student 中输入数据

另外当基本表 Student 不需要时，就必须删除表 Student。删除表 Student 的操作步骤是选中表 Student，然后右击“删除”即可。

(3) 数据库表的结构

如果要修改数据库表的结构，先选中表 Student，然后选择“表设计器”，在“设计表”窗口中直接修改后保存即可。如图 1-6。

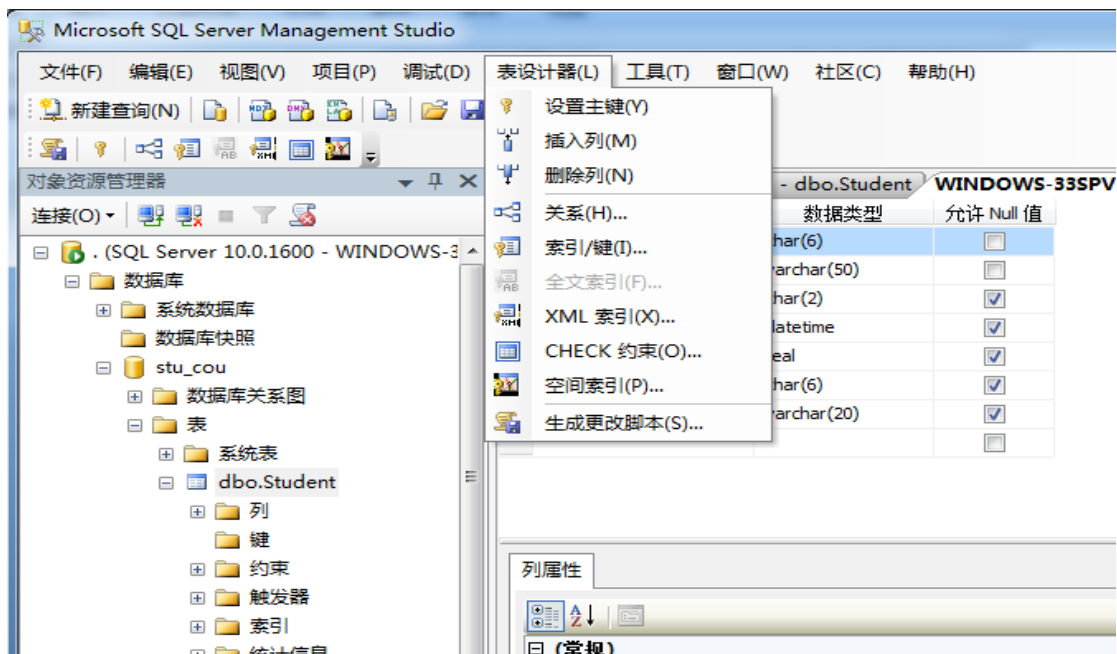


图 1-6 修改 Student 的结构

四、习题

如下表，完成下面习题

学生表：Student1 (Sno, Sname, Ssex, Sbirth, Spre, Sclass, Sdept)

Student 由学号 (Sno)、姓名 (Sname)、性别 (Ssex)、出生日期 (Sbirth)、已修学分 (Spre)、班级 (Sclass)、所在系 (Sdept) 七个属性组成，其中 Sno 为主码。

课程表：Course1 (Cno, Cname, Ctype, Cpno, Ccredit)

Course 由课程号 (Cno)、课程名 (Cname)、课程类型 (Ctype)、先修课号 (Cpno)、学分 (Ccredit) 五个属性组成，其中 Cno 为主码。

学生选课表：SC1 (Sno, Cno, Grade)

由学号 (Sno)、课程号 (Cno)、成绩 (Grade) 三个属性组成，主码为 (Sno, Cno) 和外码。

1、请应用“管理工具 SSMS”在数据库 Stu_Cou 中建立表 Course、表 SC，并输入数据。

实验二 备份与还原

一、实验目的

- 1、掌握应用“管理工具”进行数据库备份
- 2、掌握应用“管理工具”进行数据库还原

二、预备知识

对于操作人员的意外操作或蓄意的破坏性操作、破坏性病毒的攻击、自然灾害等原因所引起的系统故障，定期进行数据备份是保证系统安全的一项重要措施。在意外情况发生时，可以依靠备份数据来恢复数据库。下面我们介绍如何利用“管理工具”对数据库进行备份，以及在系统发生故障时怎样从备份数据中恢复数据库。

三、实验示例

【例 2-1】为数据库 Stu_Cou 创建备份文件

(1) 在“管理工作”中选择数据库“Stu_Cou”，右键在快捷菜单中选择“任务”——“备份...”。如图 2-1 所示：

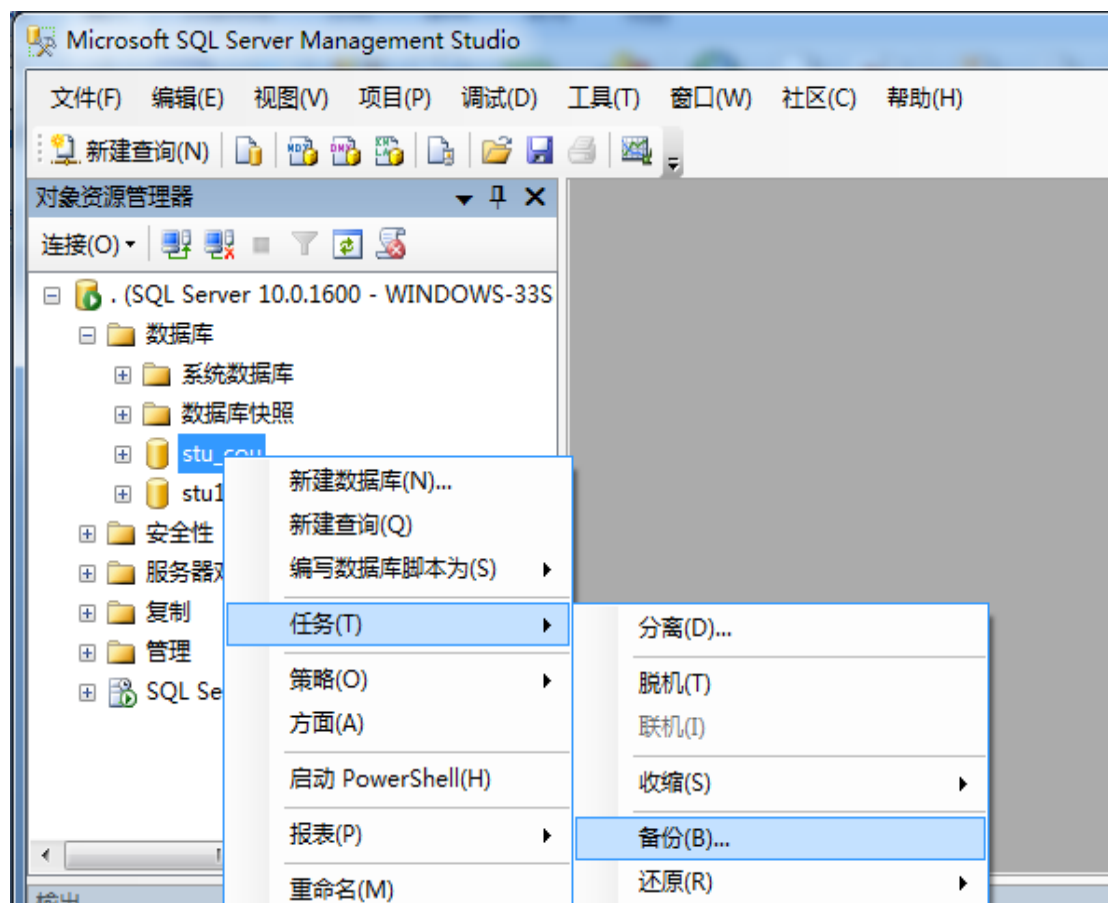


图 2-1 备份数据库（一）

(2) 在图 2-2 中，备份区选择“数据库-完整”，目标区选择“添加”按钮。

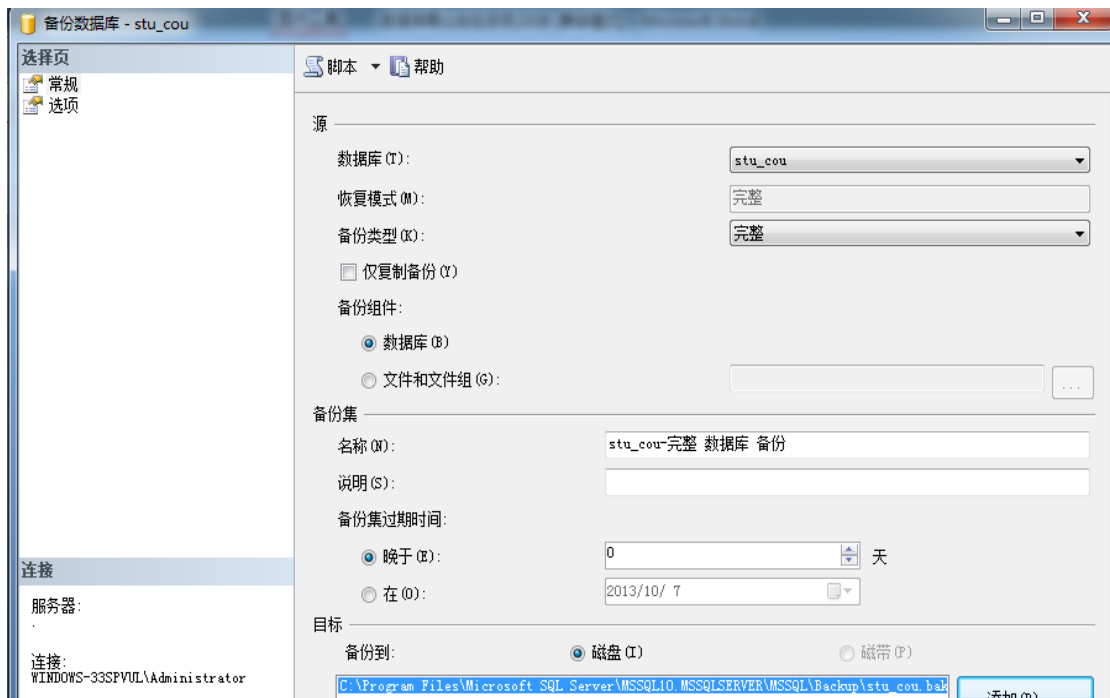


图 2-2 备份数据库（二）

（3）在图 2-3 的窗口中选择“文件名”，并选择“浏览”键寻找备份文件的路径

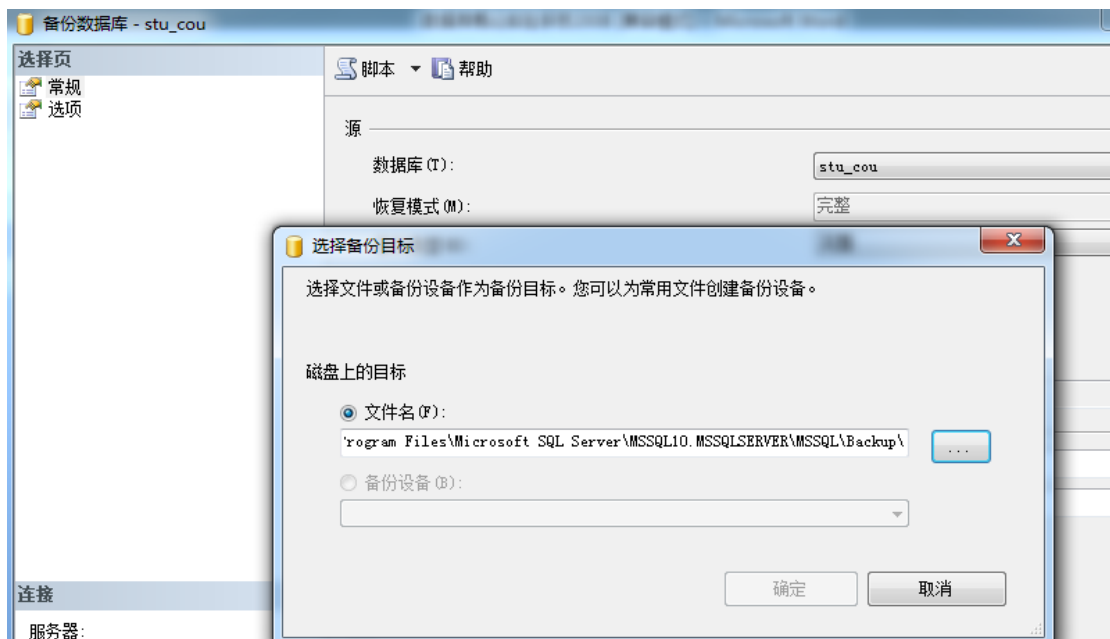


图 2-3 备份数据库（三）

（4）在图 2-4 的窗口中，首先选择备份文件所在的路径，例如“D:\”；然后在“文件名”后填写“Stu”，即备份文件的名称。逐步确定后，数据库 Stu_Cou 的备份完成。

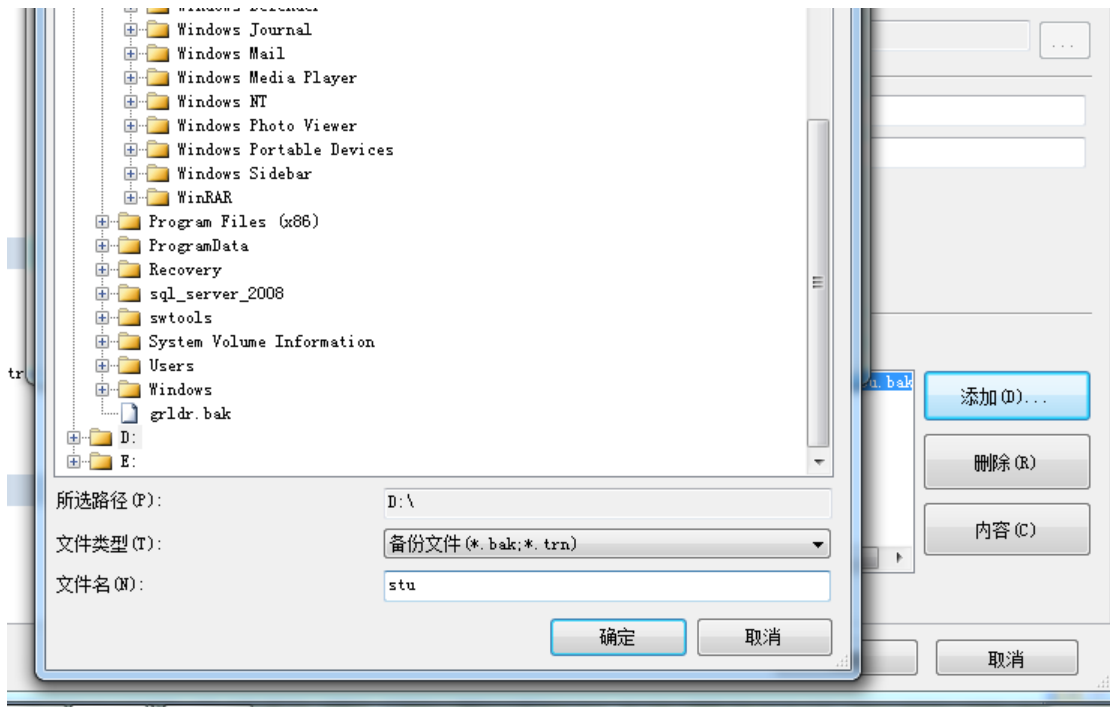


图 2-4 备份数据库（四）

【例 2-2】根据数据库 Stu_Cout 的备份文件 Stu，还原意外丢失的数据库 Stu_Cou。

(1) 在“管理工具”中选择数据库，右键在快捷菜单中选择“还原数据库”。如图 2-5 所示：

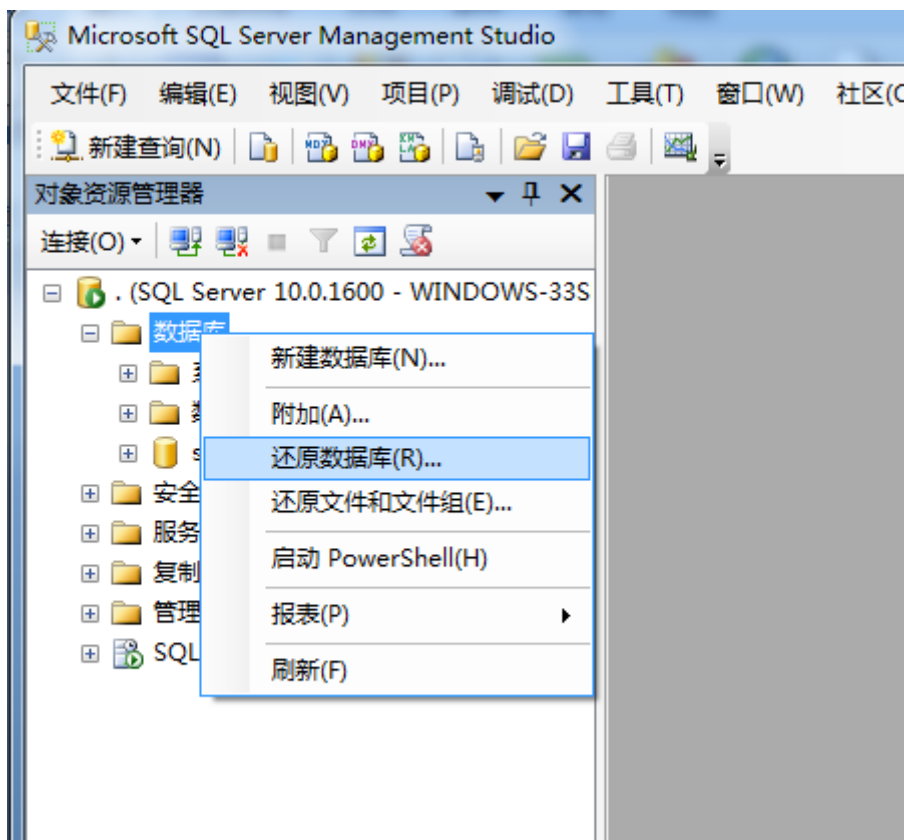


图 2-5 还原数据库（一）

(2) 在图 2-6 的窗口中，“还原为数据库”后填写 Stu_Cou；还原的源区选择“源设备”。

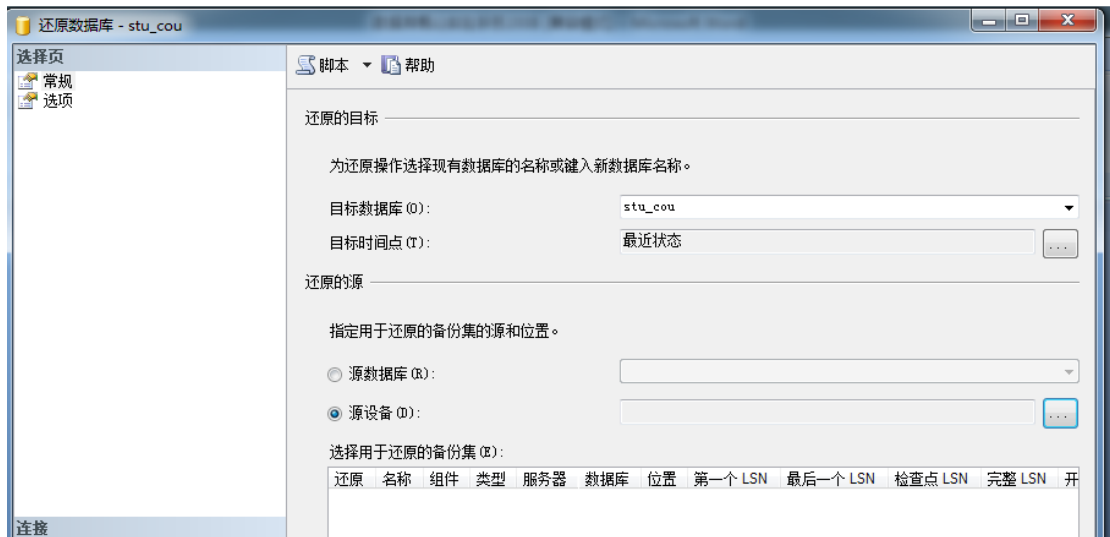


图 2-6 还原数据库（二）

(3) 在图 2-7 中选择“添加”按钮。

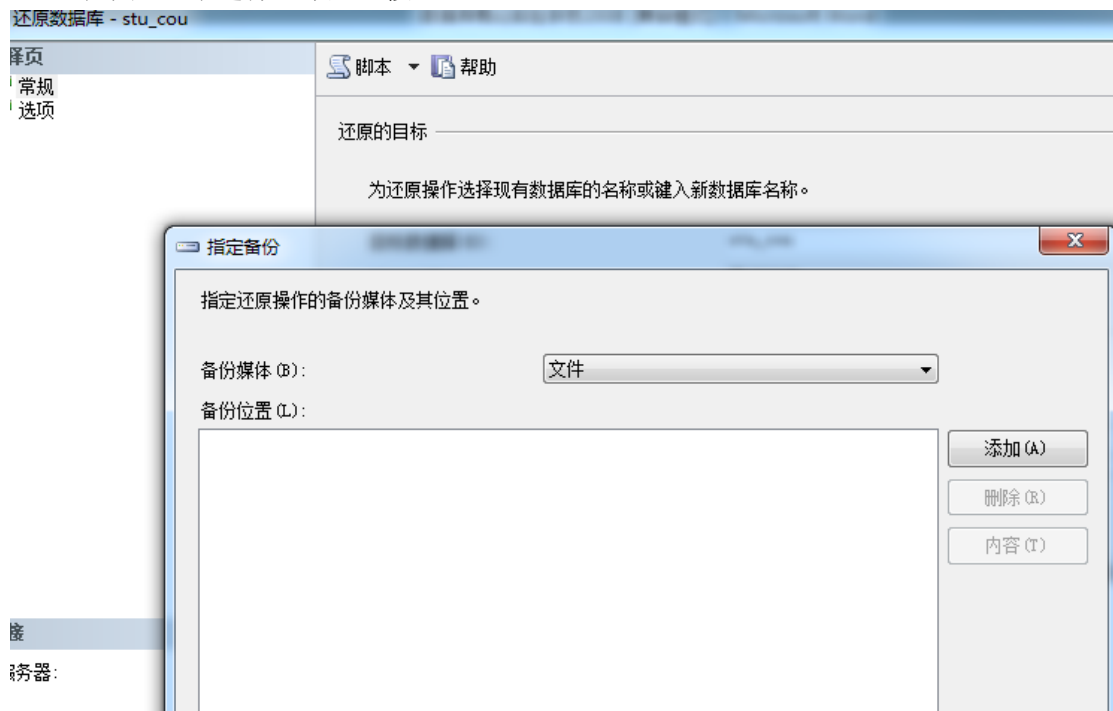


图 2-7 还原数据库（三）

(4) 在图 2-8 的窗口中，文件名里填写原数据库备份文件的路径，例如“D:\Stu”，或者通过浏览得到相应的路径。最后逐步确定后，数据库 Stu_Cou 的还原完成。

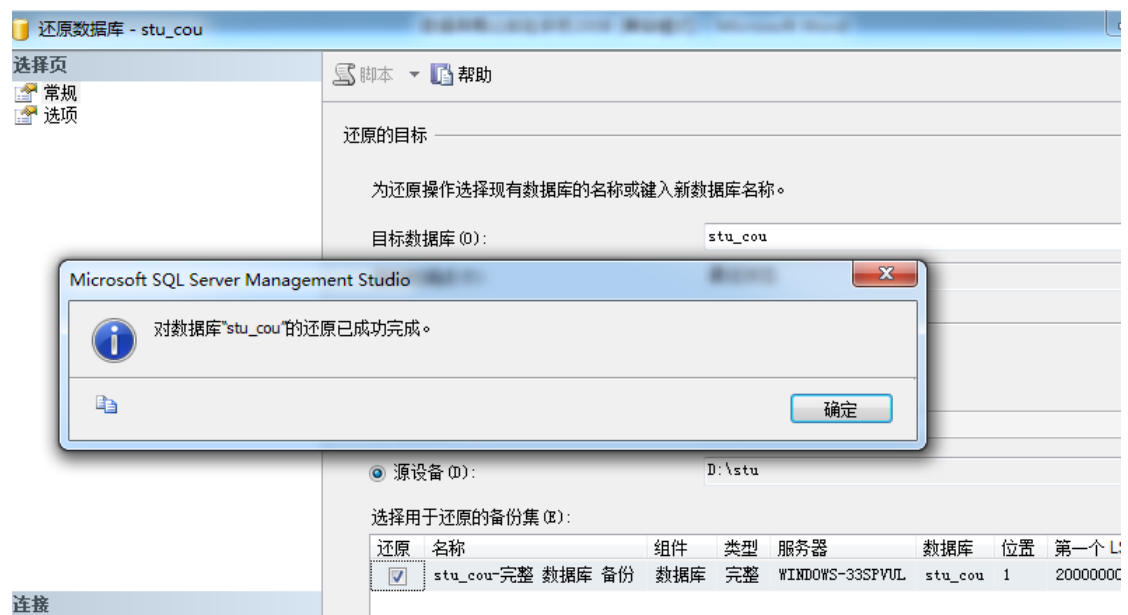


图 2-8 还原数据库（四）

四、习题

- 1、新建数据库“spj”（自己设计些表）。
- 2、利用“资源管理器”对该数据库进行备份，得到备份文件“spj”。
- 3、删除数据库“spj”，从备份文件“spj”恢复数据库 spj。