

实验9—数据库完整性和安全性

说明：本次实验主要学习数据库完整性和安全性。

一、实验目的

1. 理解关系数据库完整性的基本概念；
2. 掌握SQL Server的6类约束：NOT NULL、PRIMARY KEY、CHECK、FOREIGN KEY、DEFAULT和UNIQUE，能够在创建表时用相应的约束准确地描述实体完整性、参照完整性和用户定义完整性；
3. 掌握增加和删除约束的方法；
4. 加深对数据库安全性的理解；
5. 掌握SQL Server中有关登录、（数据库）用户和权限的管理方法。

第一部分 数据库完整性

SQL Server的完整性机制

SQL Server中实施数据完整性的两种形式：

- **声明式数据完整性**。将数据应符合的条件融入到对象定义中，这样SQL Server会自动确保符合事先指定的约束条件。
- **程序式数据完整性**。如果约束条件及其实施均通过程序代码完成，则这种完整性实施方式称为程序式数据完整性。其**特点**是可实现更复杂的条件约束。在实现中**可利用触发器**。

CREATE TABLE 创建基本表语句可定义的约束类型：

列级完整性约束：

- ① **NOT NULL** 限制列取值不能为空。
- ② **DEFAULT** 指定列的默认值。
- ③ **UNIQUE** 限制列的取值不能重复。
- ④ **CHECK** 限制列的取值范围。
- ⑤ **PRIMARY KEY** 指定本列为主码。
- ⑥ **FOREIGN KEY** 指定本列为外码。

表级完整性约束：

- ① **UNIQUE** 限制列的取值不能重复。
- ② **CHECK** 限制列的取值范围。
- ③ **PRIMARY KEY** 指定主码。
- ④ **FOREIGN KEY** 指定外码。

CHECK约束

- CHECK约束用来检查字段值所允许的范围。
- 语法格式：

[CONSTRAINT <约束名>] CHECK(<条件>)

其中，

<条件>是逻辑表达式，称为**CHECK约束表达式**，其构成与WHERE子句中逻辑表达式的构成相同。

【例】 以下的表定义中限定Age字段只能取整数值，并且范围只能为0~100。

```
CREATE TABLE Client
```

```
( ...
```

```
    Age INT CONSTRAINT age_check CHECK (Age>=0 AND  
Age<=100),
```

```
    ...
```

```
)
```



约束名

UNIQUE约束

- UNIQUE约束用于指明某列或多个列组合的取值必须唯一。
- 若对列指明了UNIQUE约束，则系统自动为其建立索引。
- UNIQUE约束定义的语法格式为：

[CONSTRAINT <约束名>] UNIQUE

【例】 以下的表定义中为Name字段定义UNIQUE约束。该约束确保没有同名记录。

```
CREATE TABLE Client
```

```
( ...
```

```
    Name CHAR(8) CONSTRAINT name_unique UNIQUE,
```

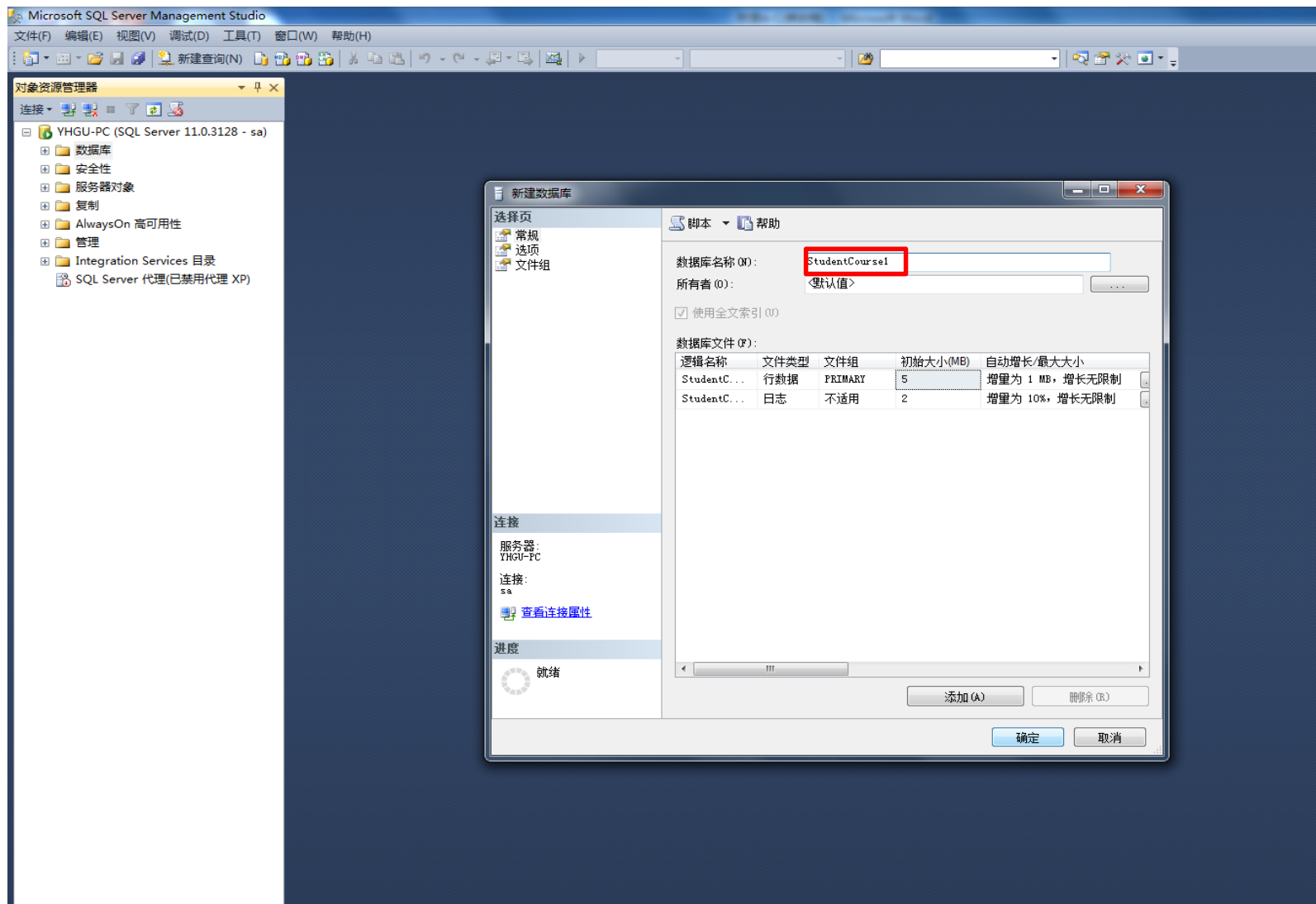
```
    ...
```

```
)
```

二、实验内容

(一) 教材练习1—“实验9 数据库完整性” (P295~297)

1. 创建数据库StudentCourse1;



2. 分别按表A9.1、表A9.2和表A9.3所示的结构和约束条件写出创建3个表的CREATE语句。

表A9.1 学生情况表（表名Student）

列 名	数 据 类 型	是否允许为空值	默 认 值	说 明
学号	char(6)	否	无	主码，表StuCourse的外码
姓名	char(12)	否	无	无学生同名
专业名	varchar(20)	是	无	
性别	char(2)	否	无	'男' '女'
出生时间	smalldatetime	是	无	
总学分	int	是	无	必须为正值
备注	text	是	无	

表A9.2 课程表（表名Course）

列 名	数 据 类 型	是否允许为空值	默 认 值	说 明
课程号	char(4)	否	无	主码，表StuCourse的外码
课程名	varchar(40)	否	无	
开课学期	int	是	无	只能为1~8
学时	int	是	无	必须为正值
学分	int	是	无	必须为正值

表A9.3 学生选课表（表名StuCourse）

列 名	数 据 类 型	是否允许为空值	默 认 值	说 明
学号	char(6)	否	无	主码
课程号	char(4)	否	无	主码
成绩	real	是	无	必须为正值

在查询分析器中输入以下创建三个表的T-SQL语句并执行：

USE StudentCourse1

GO

CREATE TABLE Student

(学号 CHAR(6) PRIMARY KEY,

姓名 CHAR (12) NOT NULL CONSTRAINT name_unique UNIQUE,

专业名 VARCHAR(20),

性别 CHAR (2) NOT NULL CONSTRAINT gen_Check CHECK (性别='男' OR 性别='女'),

出生时间 SMALLDATETIME,

总学分 INT CONSTRAINT tot_Check CHECK (总学分>=0),

备注 TEXT

)

GO

CREATE TABLE Course

(课程号 CHAR(4) PRIMARY KEY,

课程名 VARCHAR(40) NOT NULL,

开课学期 INT CONSTRAINT Sem_Check CHECK (开课学期 BETWEEN 1 AND 8),

学时 INT CONSTRAINT Tm_Check CHECK (学时>=0),

学分 INT CONSTRAINT Score_Check CHECK (学分>=0)

)

GO

CREATE TABLE StuCourse

(学号 CHAR(6) NOT NULL,

课程号 CHAR(4) NOT NULL,

成绩 INT CONSTRAINT Grade_Check CHECK (成绩>=0),

PRIMARY KEY (学号, 课程号),

FOREIGN KEY(学号) REFERENCES Student(学号),

FOREIGN KEY(课程号) REFERENCES Course(课程号)

)

GO

```
USE StudentCourse1
```

```
GO
```

```
CREATE TABLE Student
```

```
(学号      CHAR(6) PRIMARY KEY,  
 姓名      CHAR (12) NOT NULL CONSTRAINT name_unique UNIQUE,  
 专业名    VARCHAR(20),  
 性别      CHAR (2) NOT NULL CONSTRAINT gen_Check CHECK (性别='男' OR 性别='女'),  
 出生时间  SMALLDATETIME,  
 总学分    INT CONSTRAINT tot_Check CHECK (总学分>=0),  
 备注      TEXT  
)
```

```
GO
```

```
CREATE TABLE Course
```

```
( 课程号    CHAR(4)  PRIMARY KEY,  
 课程名    VARCHAR(40) NOT NULL,  
 开课学期  INT CONSTRAINT Sem_Check CHECK (开课学期 BETWEEN 1 AND 8),  
 学时      INT CONSTRAINT Tm_Check CHECK ( 学时>=0),  
 学分      INT CONSTRAINT Score_Check CHECK ( 学分>=0)
```

```
)
```

```
GO
```

```
CREATE TABLE StuCourse
```

```
( 学号      CHAR(6)  NOT NULL,  
 课程号    CHAR(4)  NOT NULL,  
 成绩      INT CONSTRAINT Grade_Check CHECK ( 成绩>=0),  
 PRIMARY KEY (学号, 课程号),  
 FOREIGN KEY(学号) REFERENCES Student(学号),  
 FOREIGN KEY(课程号) REFERENCES Course(课程号)
```

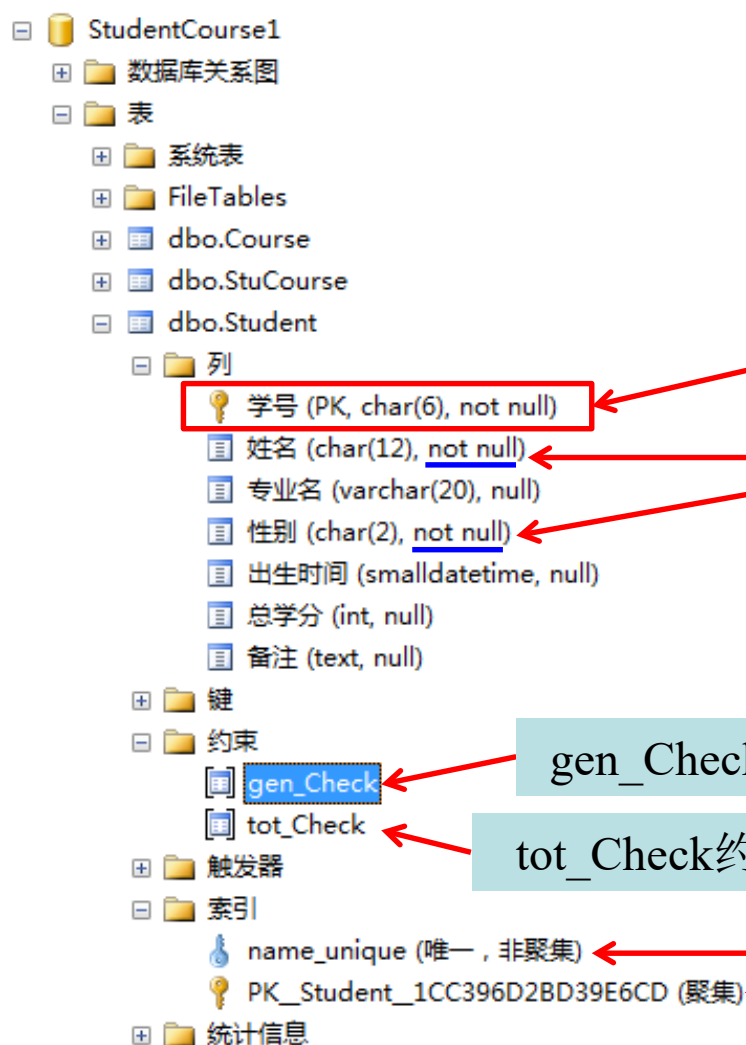
```
)
```

```
GO
```

创建表Student、Course、StuCourse

3. 在对象资源管理器中查看约束

单击要查看约束的表，打开该表的“列”、“约束”节点和“索引”等节点，将显示该表上的每个约束和索引：



PRIMARY KEY约束

NOT NULL约束

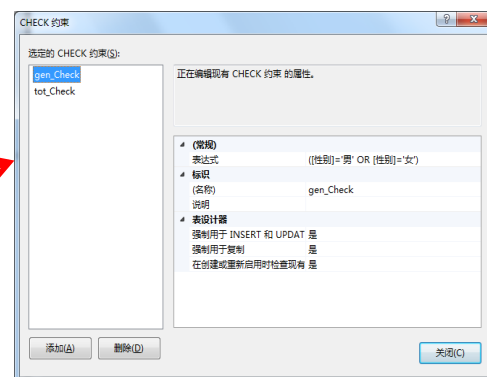
单击“修改”快捷菜单
可查看和修改约束

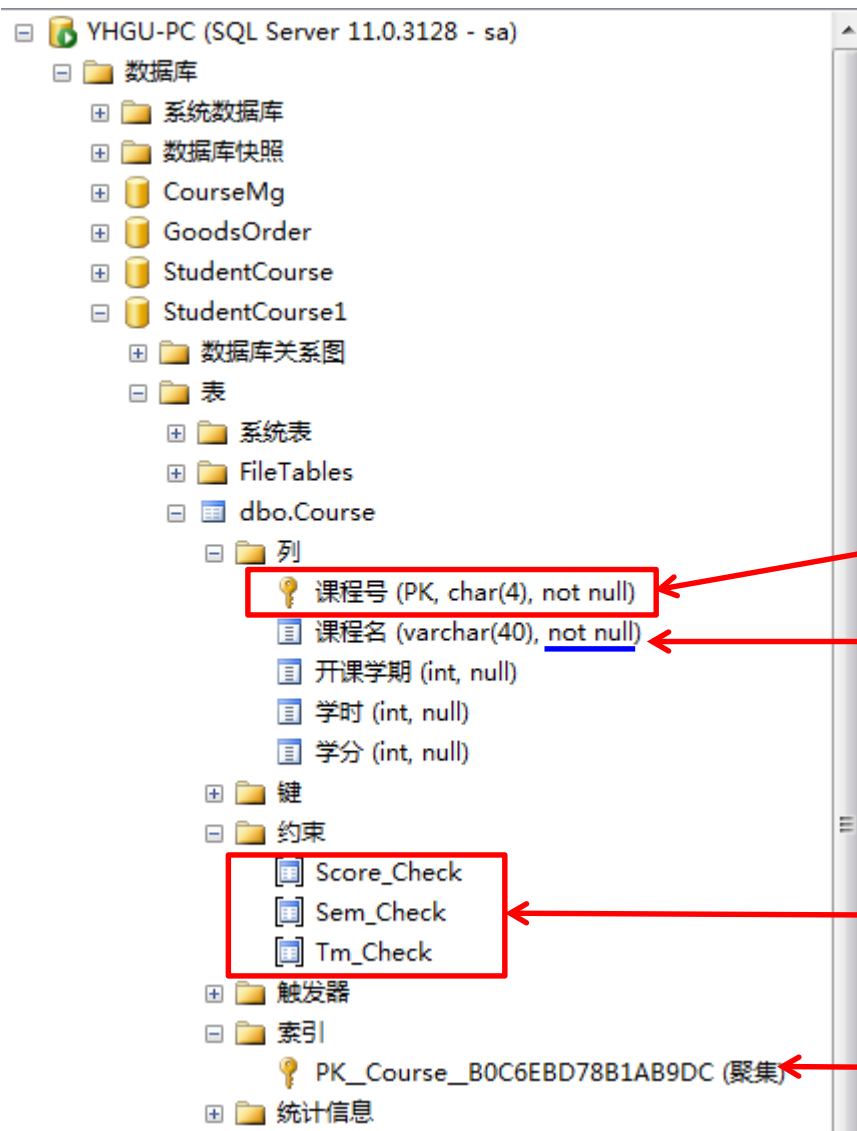
gen_Check约束

tot_Check约束

UNIQUE约束上自动建非聚集索引

PRIMARY KEY上自动建聚集索引





PRIMARY KEY约束

NOT NULL约束

CHECK约束

PRIMARY KEY上自动建聚集索引

YHGU-PC (SQL Server 11.0.3128 - sa)

数据库

系统数据库

数据库快照

CourseMg

GoodsOrder

StudentCourse

StudentCourse1

数据库关系图

表

系统表

FileTables

dbo.Course

dbo.StuCourse

列

学号 (PK, FK, char(6), not null)

课程号 (PK, FK, char(4), not null)

成绩 (int, null)

键

PK_StuCourse_77CFF86FCD98E562

FK_StuCourse_课程号_1BFD2C07

FK_StuCourse_学号_1B0907CE

约束

Grade_Check

触发器

索引

PK_StuCourse_77CFF86FCD98E562 (聚集)

统计信息

dbo.Student

PRIMARY KEY约束
FOREIGN KEY约束

查看主键、外键

CHECK约束

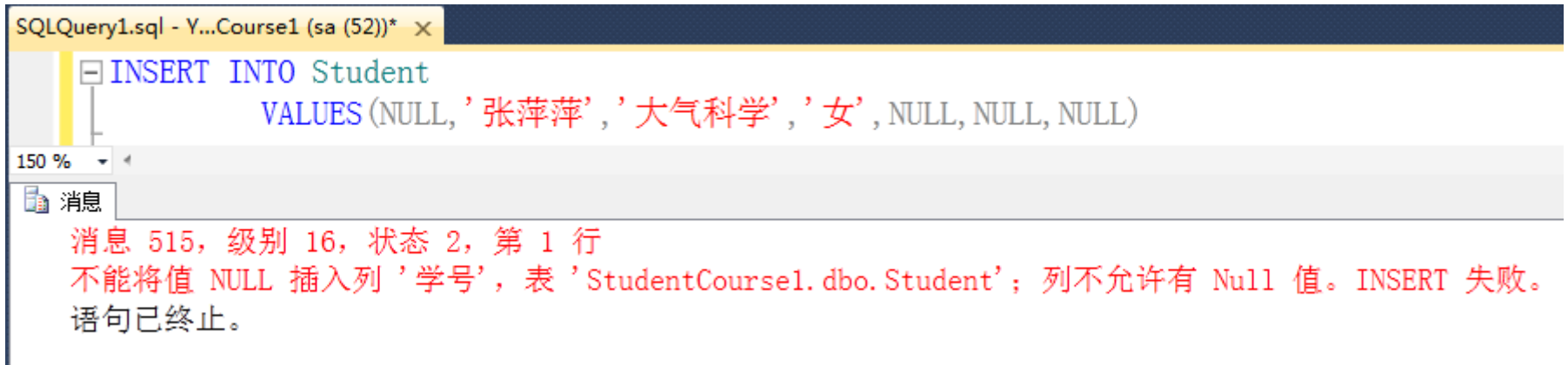
PRIMARY KEY上自动建聚集索引

测试完整性约束:

(1) 主码约束测试

在查询分析器中输入以下插入数据的T-SQL语句并执行:

```
INSERT INTO Student  
VALUES(NULL,'张萍萍','大气科学','女',NULL,NULL,NULL)
```



“学号”字段值违反了主码约束

(2) 外码约束测试

先向Student表和Course表添加以下数据：

Student表

学号	姓名	专业名	性别	出生时间	总学分	备注
070101	丁一平	计算机科学与技术	男	1989-05-01 00...	80	三好学生
070102	王红	计算机科学与技术	女	1988-12-20 00...	80	NULL

Course表

课程号	课程名	开课学期	学时	学分
1001	高等数学1	1	80	5
1002	高等数学2	2	80	5

在查询分析器中输入以下插入数据的T-SQL语句并执行：

INSERT INTO StuCourse VALUES('070105','1003',85)

“学号”、“课程号” 字段值违反了外码约束

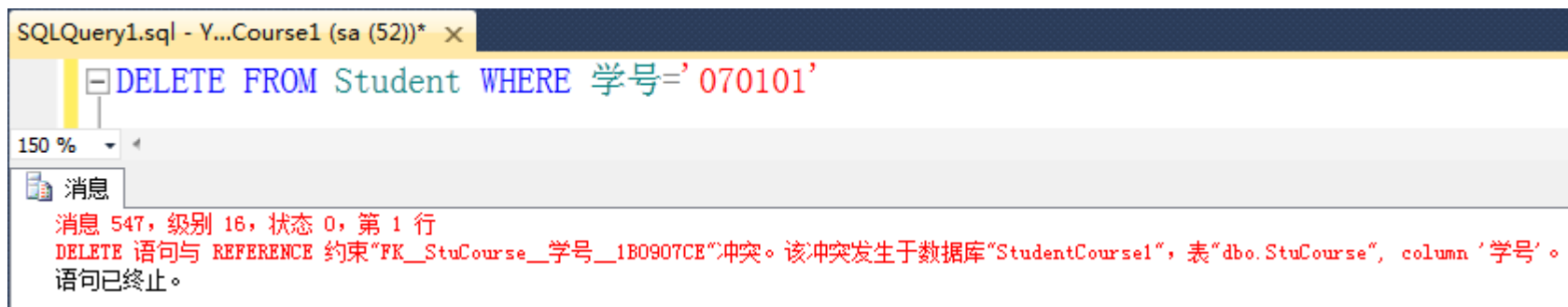
向StuCourse表添加以下数据：

StuCourse表

学号	课程号	成绩
070101	1001	85
070101	1002	90
070102	1001	98

在查询分析器中输入以下删除数据的T-SQL语句并执行：

DELETE FROM Student WHERE 学号='070101'



“学号”字段值违反了外码约束

(3) UNIQUE约束测试

在查询分析器中输入以下插入数据的T-SQL语句并执行：

INSERT INTO Student

VALUES('070501','丁一平','大气科学','女',NULL,NULL,NULL)

```
SQLQuery1.sql - Y...Course1 (sa (52))* X
INSERT INTO Student
VALUES(' 070501', ' 丁一平', ' 大气科学', ' 女', NULL, NULL, NULL)

150 %
消息
消息 2627, 级别 14, 状态 1, 第 1 行
违反了 UNIQUE KEY 约束 “name_unique”。不能在对象 “dbo.Student” 中插入重复键。重复键值为 (丁一平) 。
语句已终止。
```

“姓名”字段值违反了UNIQUE约束

在查询分析器中输入以下修改数据的T-SQL语句并执行：

UPDATE Student SET 姓名='丁一平' WHERE 学号='070102'

```
SQLQuery1.sql - Y...Course1 (sa (52))* X
UPDATE Student SET 姓名=' 丁一平' WHERE 学号=' 070102'

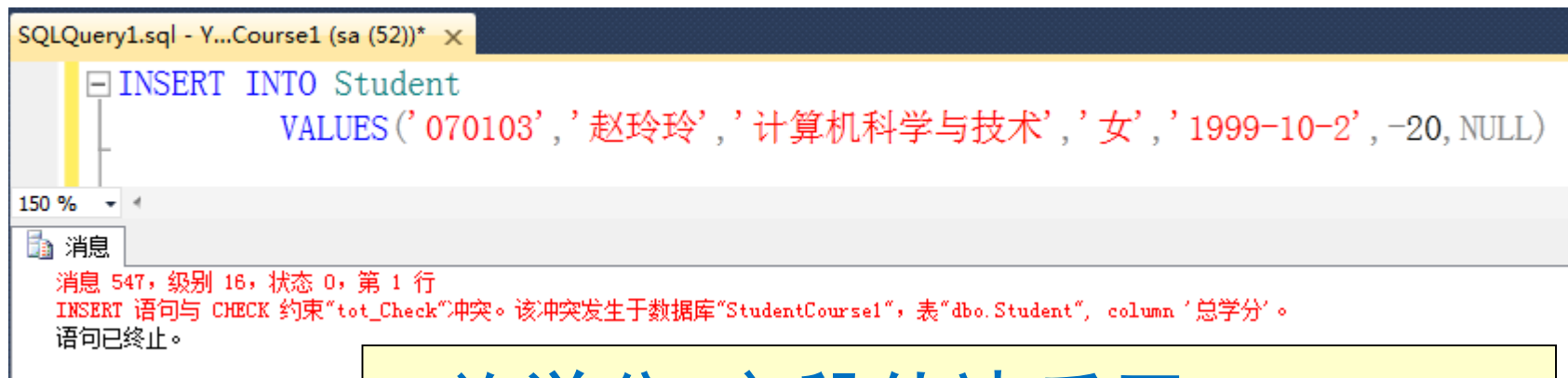
150 %
消息
消息 2627, 级别 14, 状态 1, 第 1 行
违反了 UNIQUE KEY 约束 “name_unique”。不能在对象 “dbo.Student” 中插入重复键。重复键值为 (丁一平) 。
语句已终止。
```

(4) check约束测试

在查询分析器中输入以下插入数据的T-SQL语句并执行：

```
INSERT INTO Student
```

```
VALUES('070103','赵玲玲','计算机科学与技术','女','1999-10-2',-20,NULL)
```



“总学分”字段值违反了tot_Check

【说明】以下约束测试，请自行设计测试用例：

Student表上的gen_Check约束

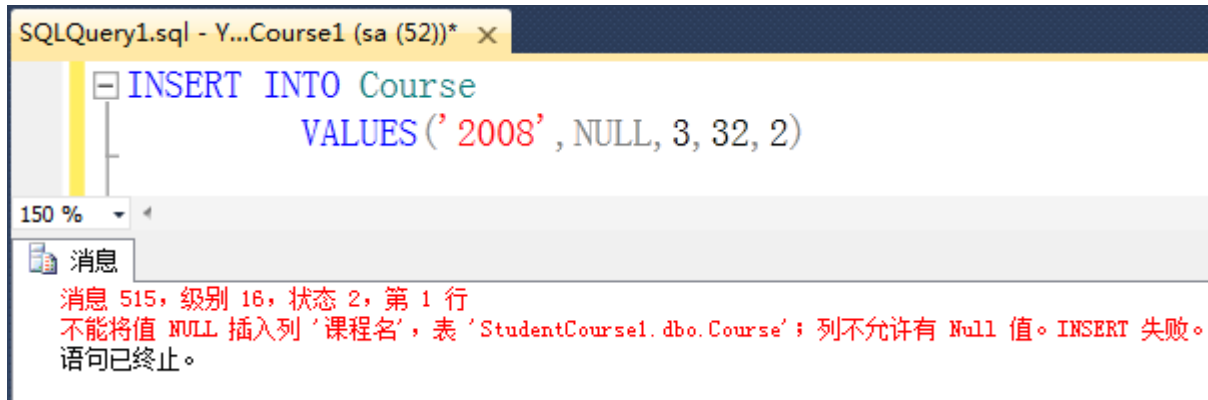
Course表上Sem_Check、Tm_Check、Score_Check约束

StuCourse表上Grade_Check约束

(5) NOT NULL约束测试

在查询分析器中输入以下插入数据的T-SQL语句并执行：

```
INSERT INTO Course  
VALUES('2008',NULL,3,32,2)
```

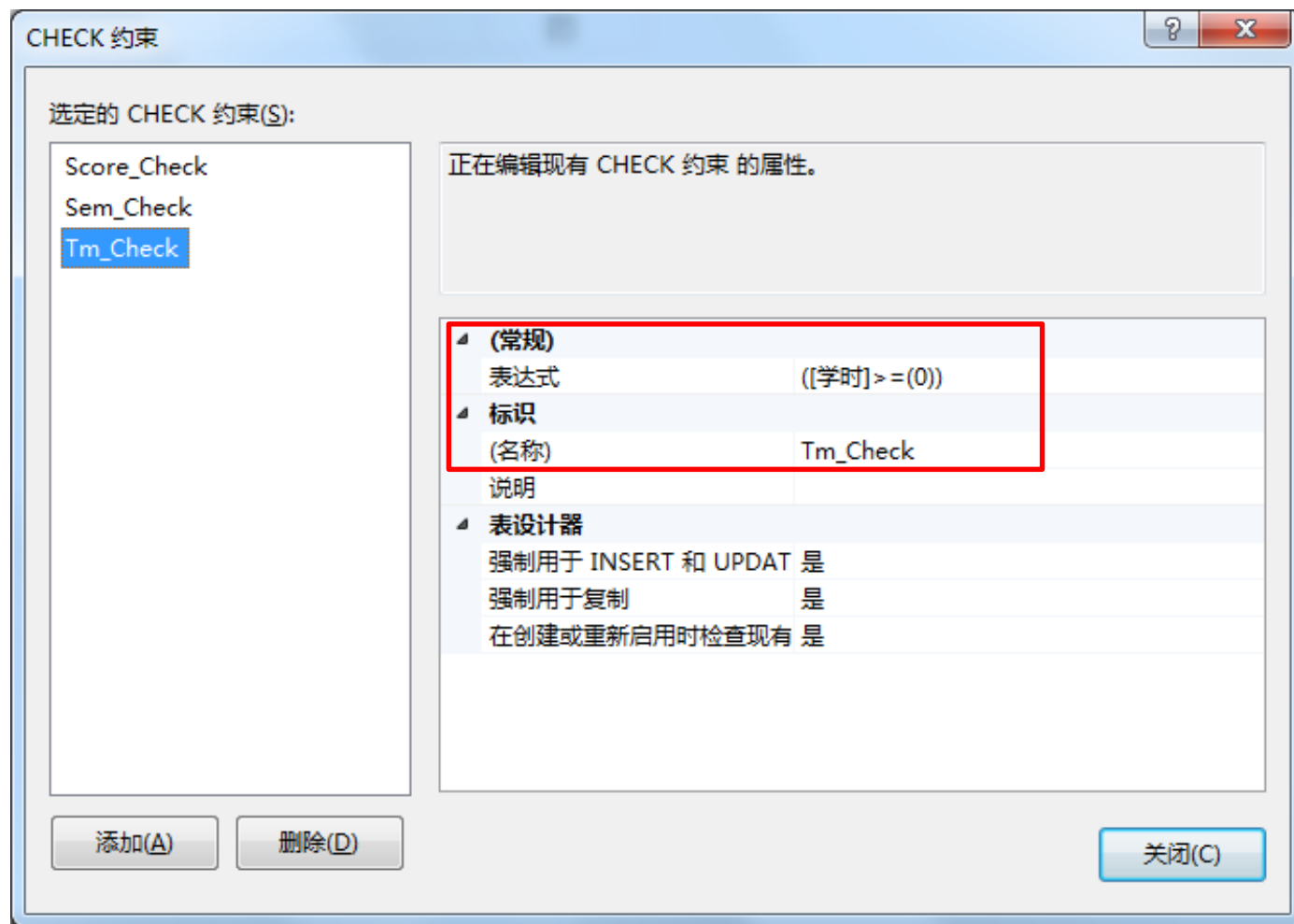


“课程名”字段值违反了NOT NULL约束

4. 修改约束定义

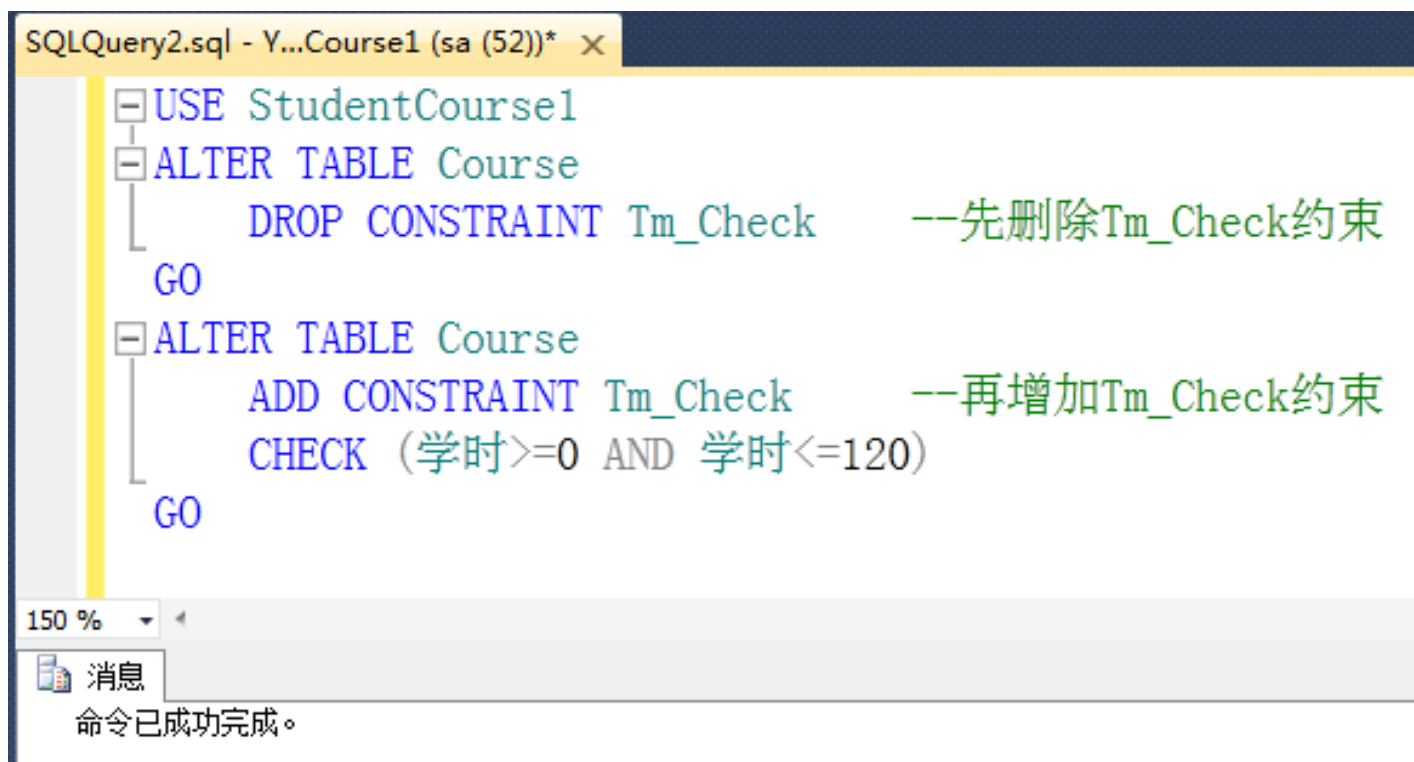
要求：将Course表的Tm_Check约束修改为大于等于0，且小于等于120

(1) 先查看Tm_Check约束内容：

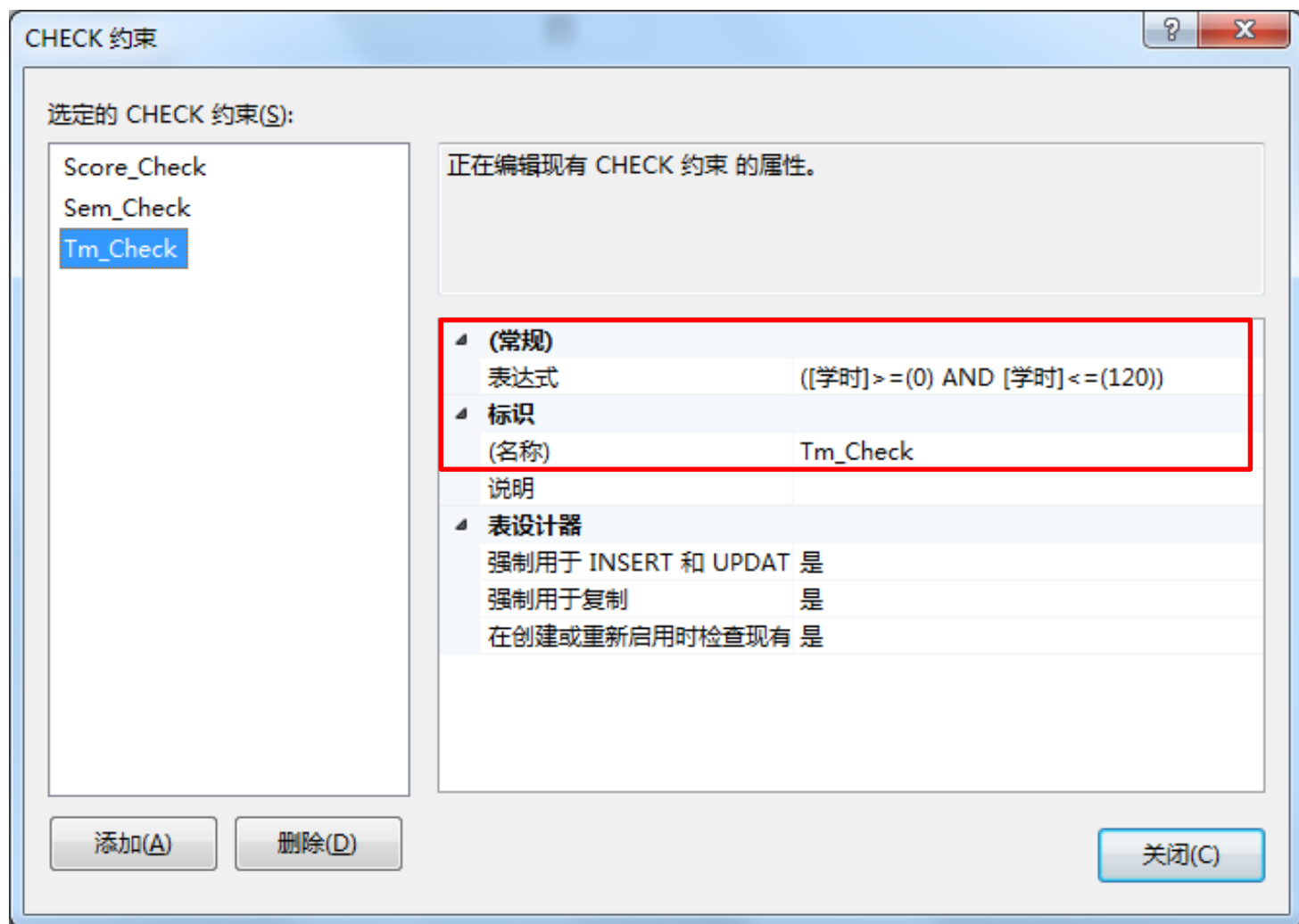


(2) 在查询分析器中输入以下T-SQL语句并执行：

```
USE StudentCourse1
ALTER TABLE Course
    DROP CONSTRAINT Tm_Check  --先删除Tm_Check约束
GO
ALTER TABLE Course
    ADD CONSTRAINT Tm_Check  --再增加Tm_Check约束
    CHECK (学时>=0 AND 学时<=120)
GO
```



(3) 查看修改后的约束内容:



5. 【思考题1】为StudentCourse1的Course表中学分字段增加“默认值为2”的约束。

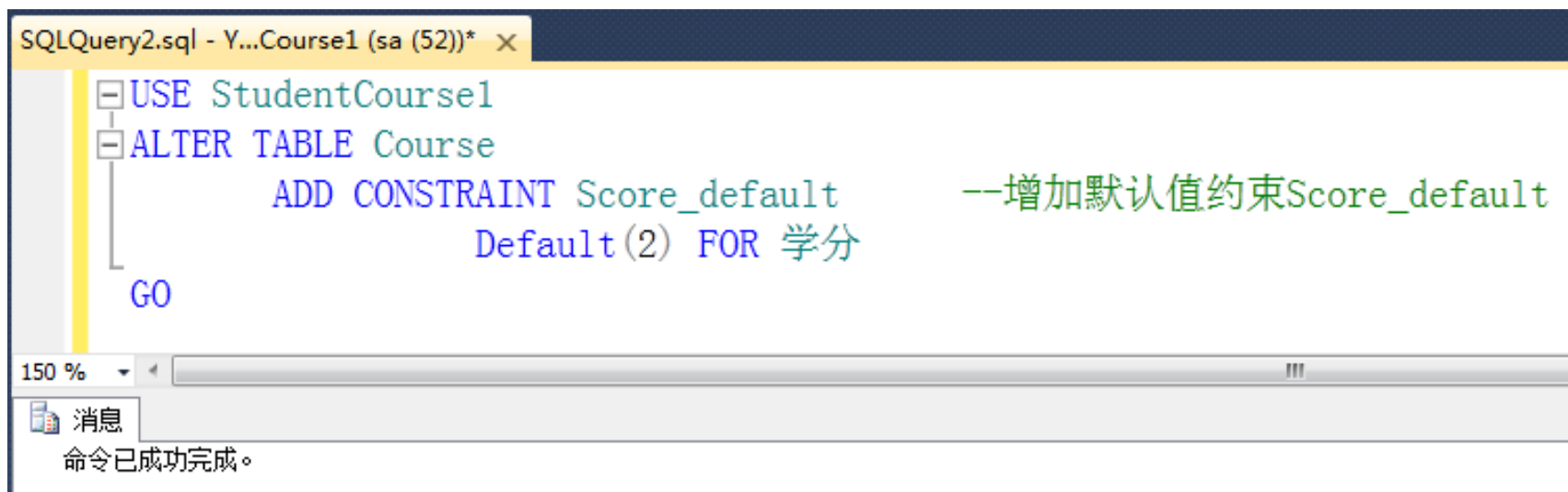
(1) 在查询分析器中输入以下T-SQL语句并执行：

```
USE StudentCourse1
```

```
ALTER TABLE Course
```

```
    ADD CONSTRAINT Score_default      --增加默认值约束Score_default  
        Default(2) FOR 学分
```

```
GO
```



(2) 查看增加的约束内容:

YHGU-PC (SQL Server 11.0.3128 - sa)

数据库

系统数据库

数据库快照

CourseMg

GoodsOrder

StudentCourse

StudentCourse1

数据库关系图

表

系统表

FileTables

dbo.Course

列

键

约束

Score_Check

Sem_Check

Tm Check

Score_default

触发器

索引

统计信息

dbo.StuCourse

dbo.Student

YHGU-PC.Student...se1 - dbo.Course x SQLQuery2.sql - Y...Course1 (sa (52))*

列名	数据类型	允许 Null 值
课程号	char(4)	<input type="checkbox"/>
课程名	varchar(40)	<input type="checkbox"/>
开课学期	int	<input checked="" type="checkbox"/>
学时	int	<input checked="" type="checkbox"/>
学分	int	<input checked="" type="checkbox"/>

在表设计器中查看字段的默认值

列属性

(常规)

(名称)

默认值或绑定

数据类型

允许 Null 值

表设计器

RowGuid

标识规范

不用于复制

大小

计算列规范

简洁数据类型

具有非 SQL Server 订阅服务器

排序规则

全文规范

是 DTS 发布的

学分

((2))

int

是

否

否

否

4

int

否

<数据库默认设置>

否

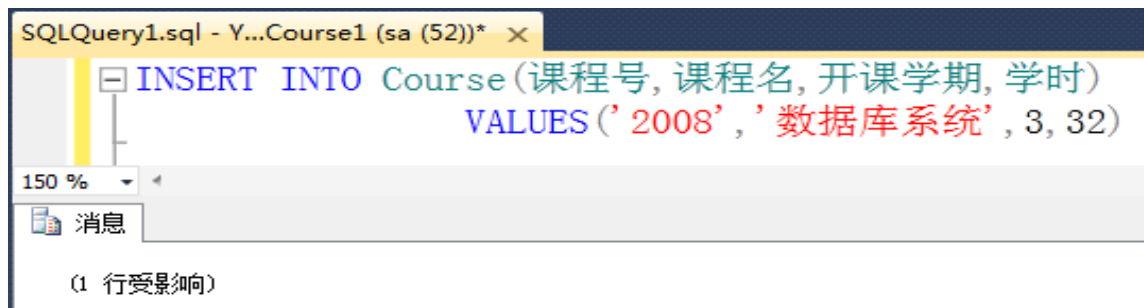
否

增加的约束

(3) DEFAULT约束测试

在查询分析器中输入以下插入数据的T-SQL语句并执行：

```
INSERT INTO Course(课程号,课程名,开课学期,学时)
VALUES('2008','数据库系统',3,32)
```



在查询分析器中输入查询语句并执行：

```
SELECT * FROM Course
```

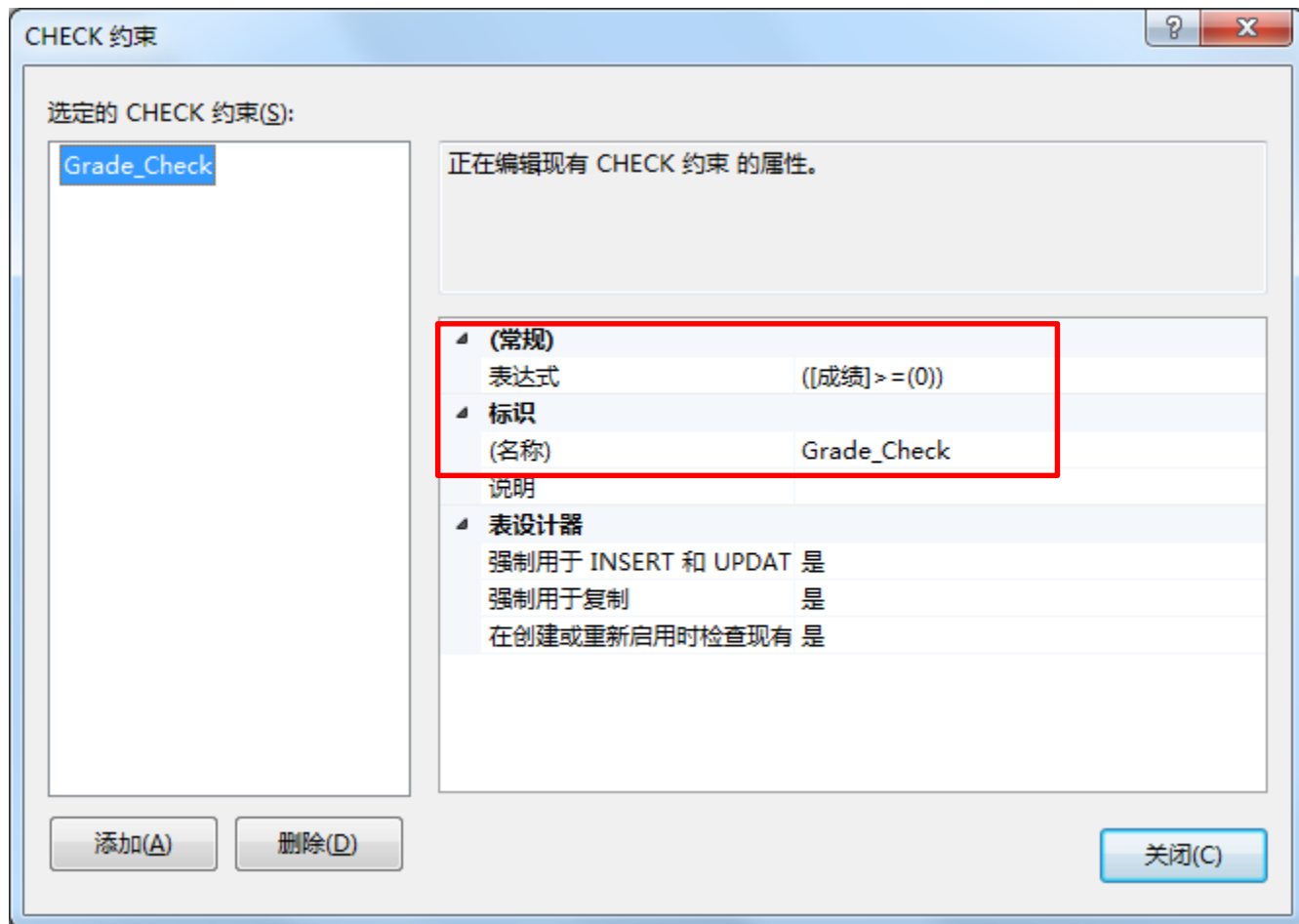
The screenshot shows the same SQL query window with the query `SELECT * FROM Course`. The results are displayed in a table with 6 columns: 课程号, 课程名, 开课学期, 学时, and 学分. The table contains 3 rows of data. The third row, representing the newly inserted record, has its '学分' (Credits) value of 2 highlighted with a red box. A red arrow points from the text '取默认值' (Take default value) to this box.

	课程号	课程名	开课学期	学时	学分
1	1001	高等数学1	1	80	5
2	1002	高学数学2	2	80	5
3	2008	数据库系统	3	32	2

取默认值

6. **【思考题2】** 将StudentCourse1的StuCourse表中Grade_Check约束修改为取值范围为0~100。

(1) 先查看Grade_Check约束内容:



(2) 在查询分析器中输入以下T-SQL语句并执行:

```
USE StudentCourse1
```

```
--检查是否已存在同名的约束，若有，则删除。
```

```
IF EXISTS (SELECT name FROM sysobjects
```

```
    WHERE name = 'Grade_Check' AND type = 'C')
```

```
    ALTER TABLE StuCourse
```

```
        DROP CONSTRAINT Grade_Check    --先删除Grade_Check约束
```

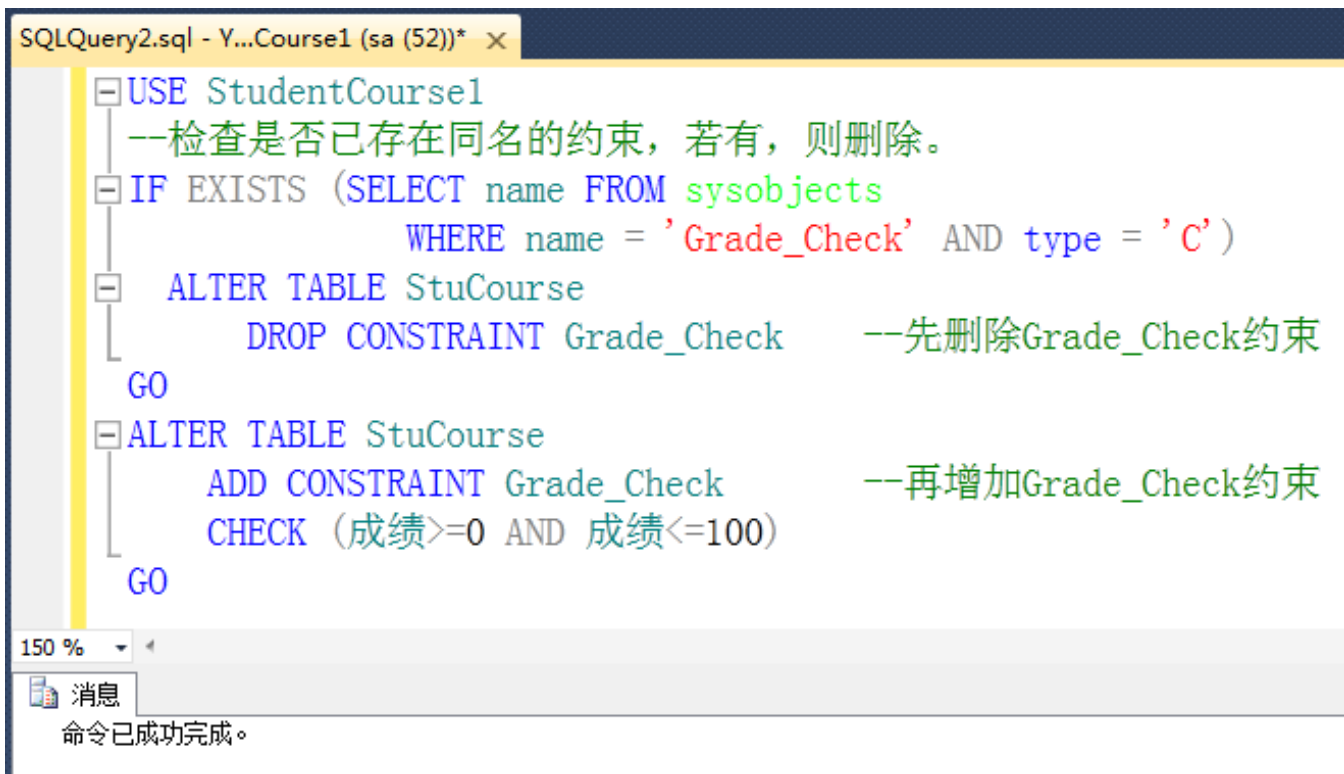
```
GO
```

```
ALTER TABLE StuCourse
```

```
    ADD CONSTRAINT Grade_Check    --再增加Grade_Check约束
```

```
    CHECK (成绩>=0 AND 成绩<=100)
```

```
GO
```



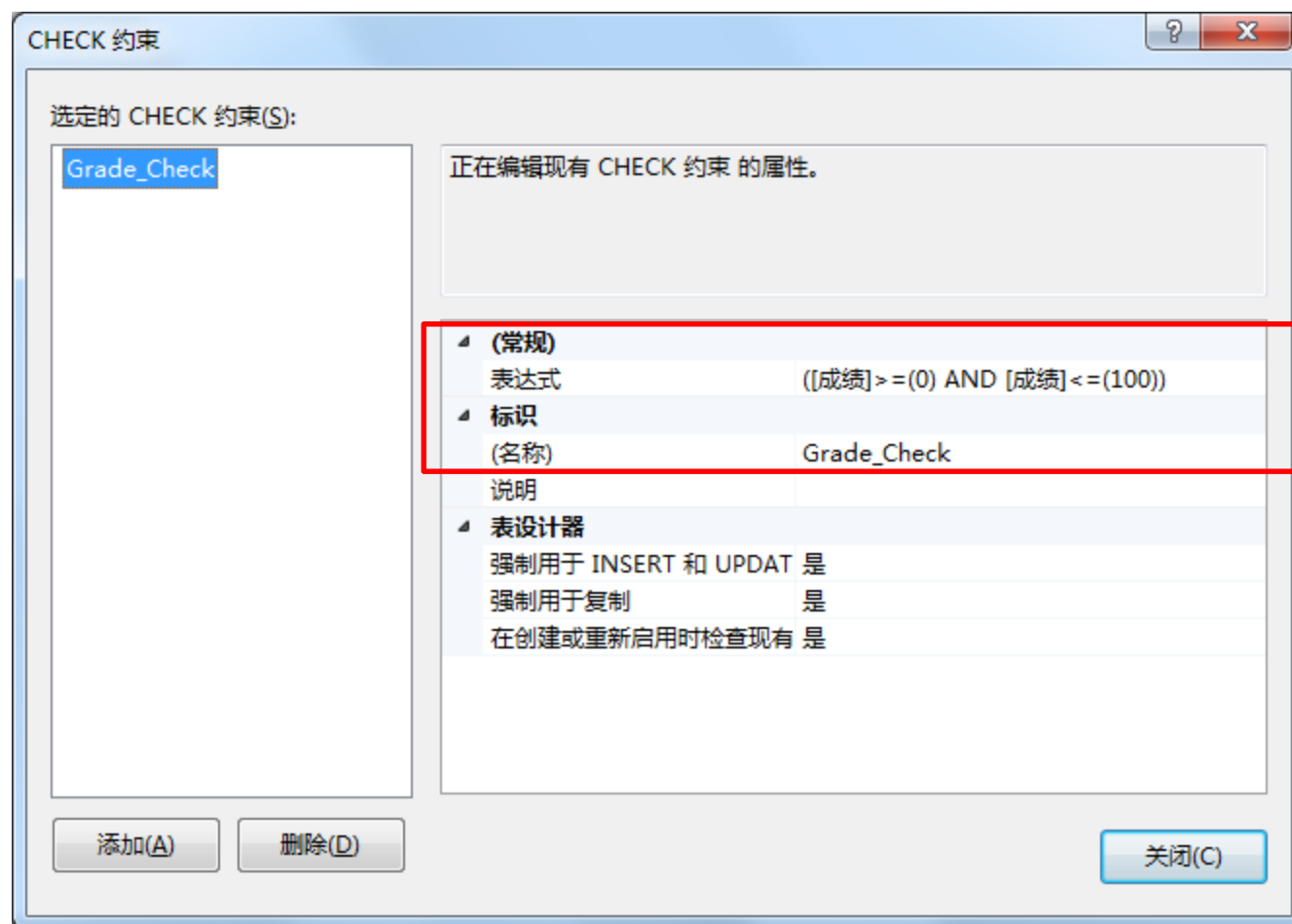
```
SQLQuery2.sql - Y...Course1 (sa (52))* x
USE StudentCourse1
--检查是否已存在同名的约束，若有，则删除。
IF EXISTS (SELECT name FROM sysobjects
            WHERE name = 'Grade_Check' AND type = 'C')
    ALTER TABLE StuCourse
        DROP CONSTRAINT Grade_Check    --先删除Grade_Check约束
GO
ALTER TABLE StuCourse
    ADD CONSTRAINT Grade_Check    --再增加Grade_Check约束
    CHECK (成绩>=0 AND 成绩<=100)
GO
```

150 %

消息

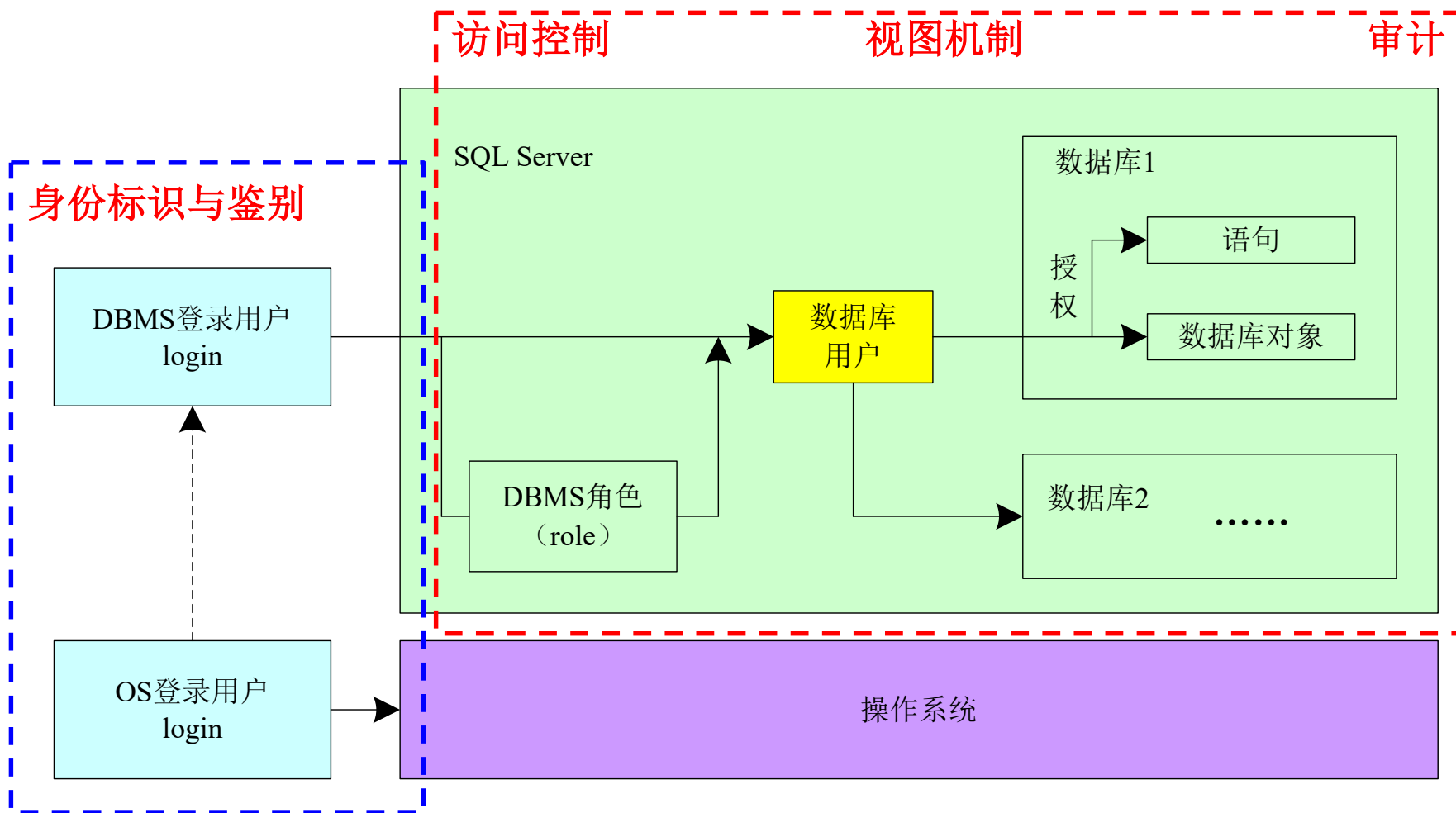
命令已成功完成。

(3) 查看修改后的约束内容:



第二部分 数据库安全性

SQL Server的安全机制



身份标识与鉴别

服务器身份验证

☐ Windows 身份验证模式 (W)

☒ SQL Server 和 Windows 身份验证模式 (S)

两种身份验证模式：

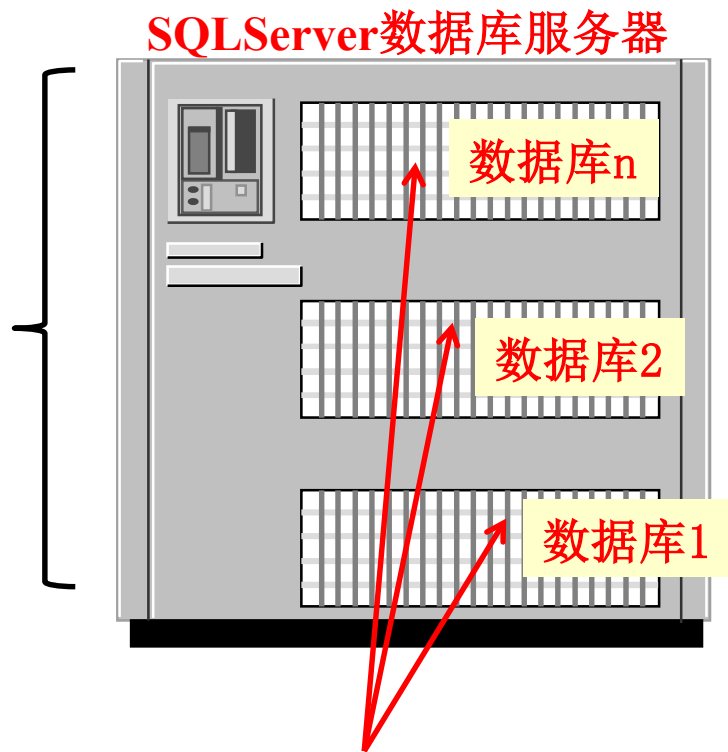
- ✓ **Windows身份验证模式：**使用Windows操作系统的安全机制进行身份验证。只要用户能够通过Windows用户账号验证，即可连接到SQL Server，SQL Server不再进行验证。
- ✓ **SQL Server和Windows身份验证模式：**在这种模式下，SQL Server要进行身份验证。用户必须提供登录名和口令，这些信息存储在SQL Server的系统表中。

登录与用户

- **登录**是连接到SQL Server数据库服务器的账号信息，包括**登录名**、**口令**等。
- 登录是**数据库服务器级**的安全策略。
- SQL Server**默认**的登录账号：

sa(最高权限)

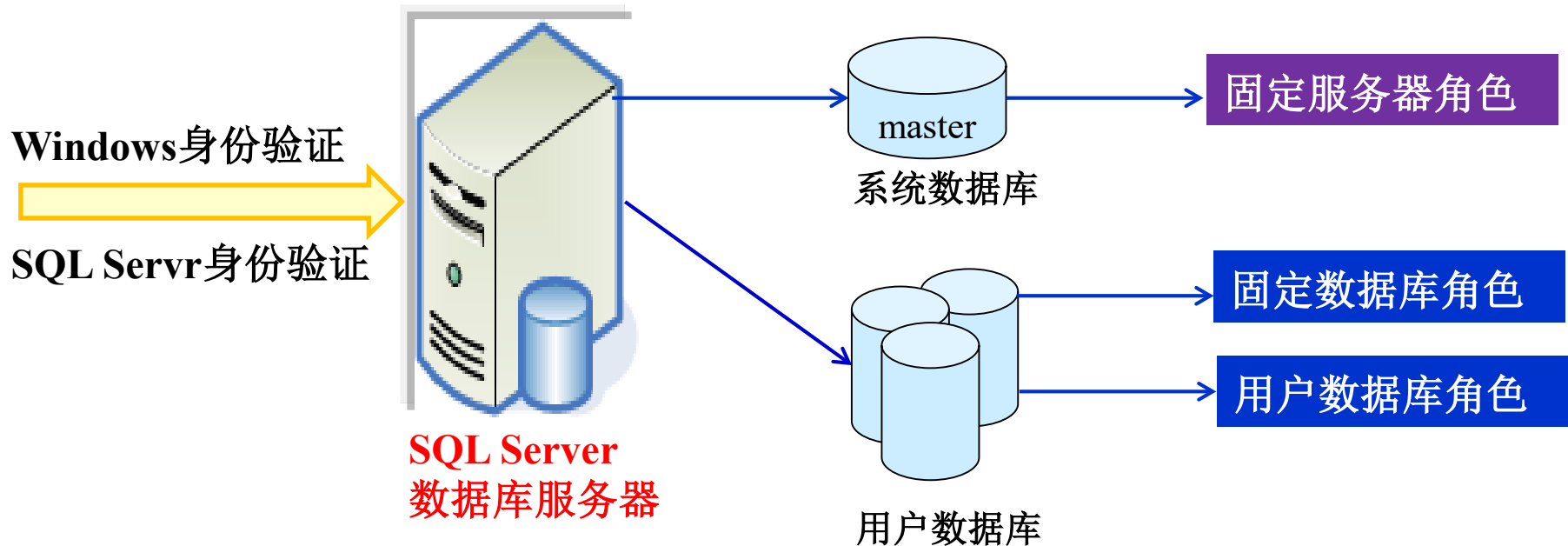
- ✓ 登录到服务器时，SQL Server 验证登录；
- ✓ 当访问数据库时，SQL Server 将验证用户。



用户是**数据库级**的安全策略，是为特定数据库定义的。

权限管理

- 权限管理是SQL Server的访问控制实现机制；
- 实现了自主访问控制(**DAC**)和基于角色的访问控制(**RBAC**).



SQL Server中的两类权限

✓ 服务器权限

- ✧ 允许DBA执行管理任务
- ✧ 一般只有DBA拥有服务器权限
- ✧ 是一种隐含权限

✓ 数据库权限

- ✧ 是数据库对象级的安全策略
- ✧ 包括对象权限和语句权限两类
- ✧ 对象权限：对表、视图等的SELECT、INSERT、UPDATE、DELETE，以及存储过程的执行
- ✧ 语句权限：创建数据库、表、视图、存储过程等的权限
- ✧ 可用GRANT、REVOKE进行授权管理

创建登录

登录名 - 新建

选择页

脚本 ? 帮助

常规

服务器角色

用户映射

安全对象

状态

登录名(N): stuUsrLogin 搜索(S)...

☐ Windows 身份验证(W)

☒ SQL Server 身份验证(S)

密码(P): ●●●

确认密码(C): ●●●

☐ 指定旧密码(I)

旧密码(O):

☐ 强制实施密码策略(E)

☐ 强制密码过期(X)

☐ 用户在下次登录时必须更改密码(U)

☐ 映射到证书(R)

☐ 映射到非对称密钥(T)

☐ 映射到凭据(M)

添加(A)

映射的凭据

凭据 提供程序

删除(Y)

默认数据库(D): master

默认语言(G): <默认>

确定 取消

连接

服务器:
LAPTOP-JCA0Q0KN\SQLSRV19

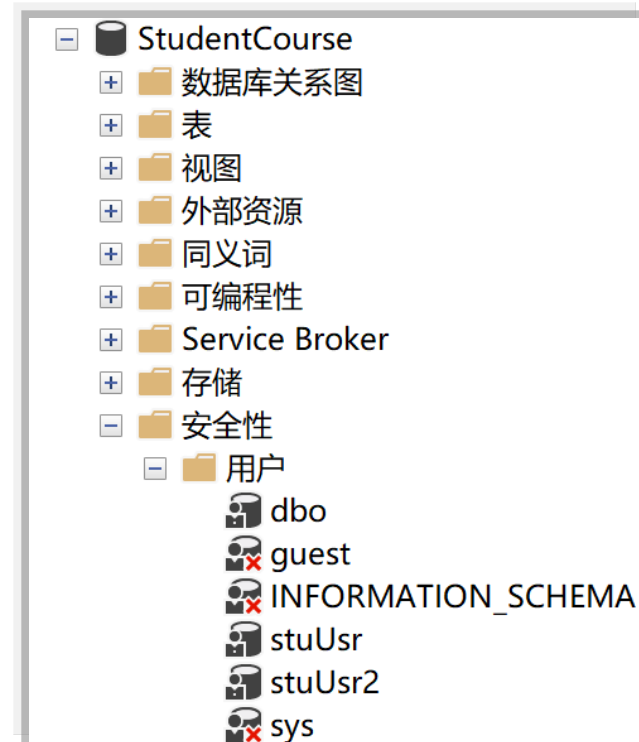
连接:
sa

查看连接属性

进度

就绪

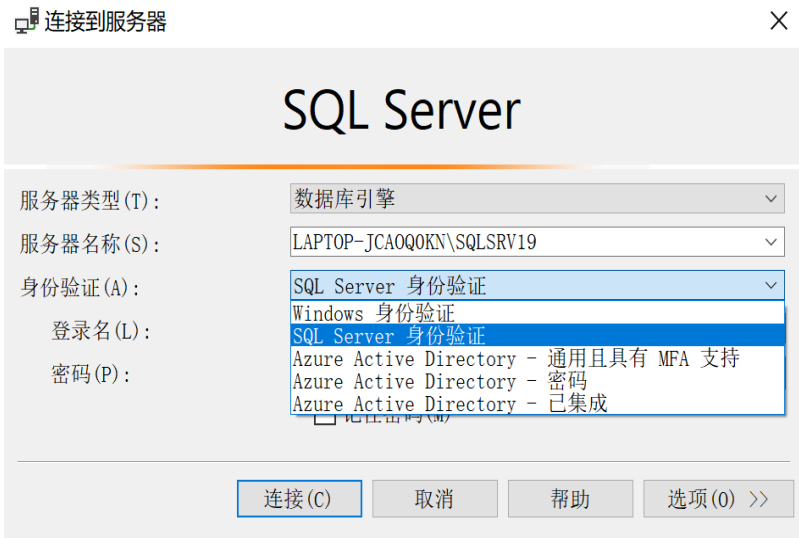
```
USE StudentCourse
CREATE USER stuUsr FOR LOGIN stuUsrLogin
CREATE USER stuUsr2 FOR LOGIN stuUsrLogin2
GRANT SELECT ON Student TO stuUsr, stuUsr2
GO
```



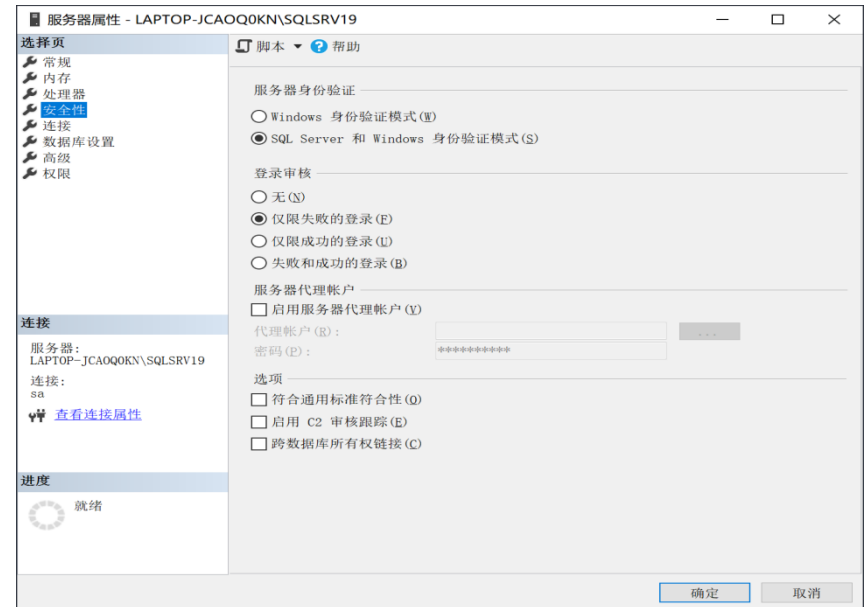
创建登录，通常还需要映射到数据库用户，方可访问数据库对象

（二）教材练习2—“实验10 数据库安全性”（P295~299）

1. 将SQL Server服务器的身份验证模式设置为“SQL Server身份验证模式”；



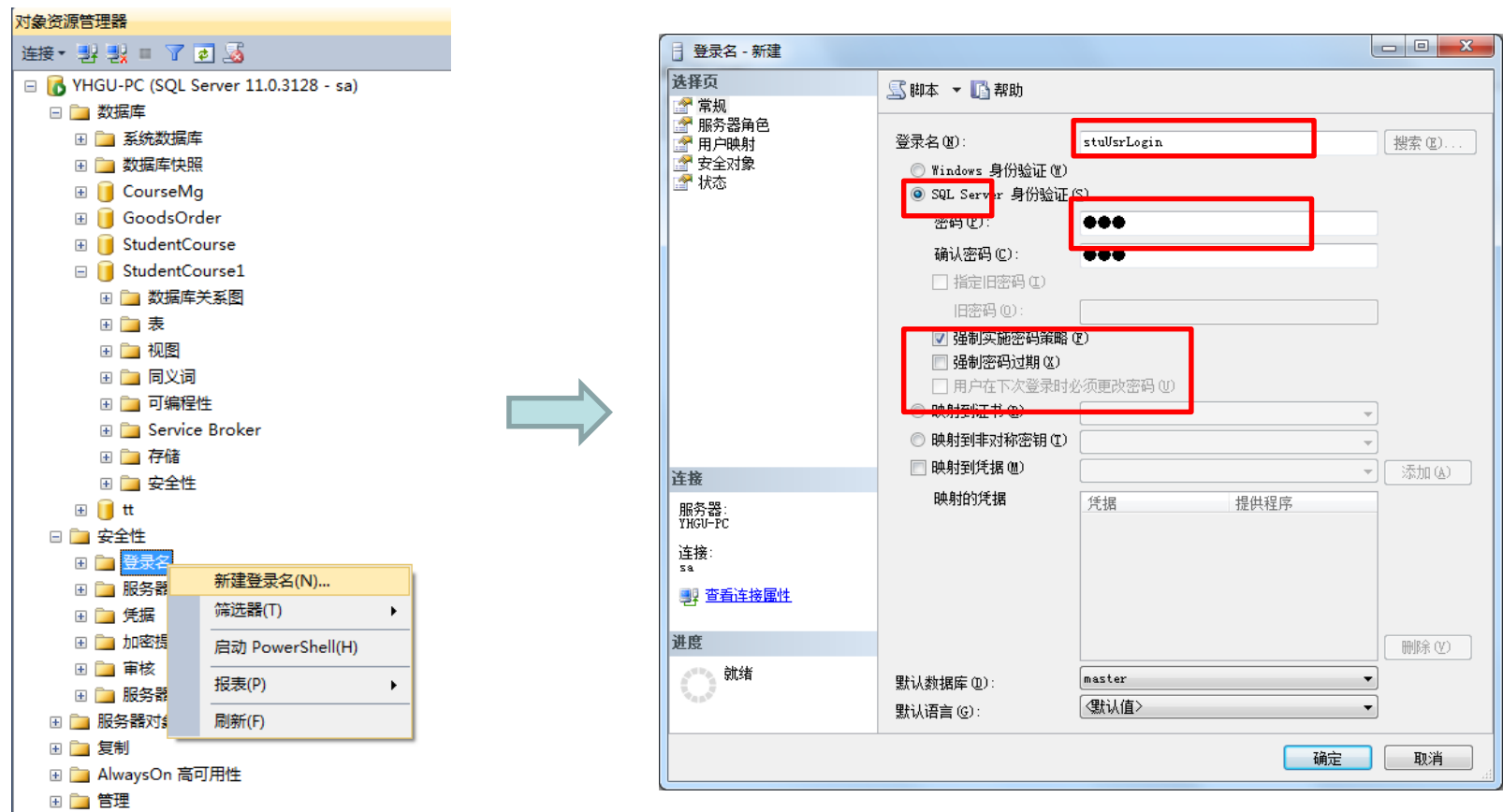
设置方法1：在启动窗口中选择



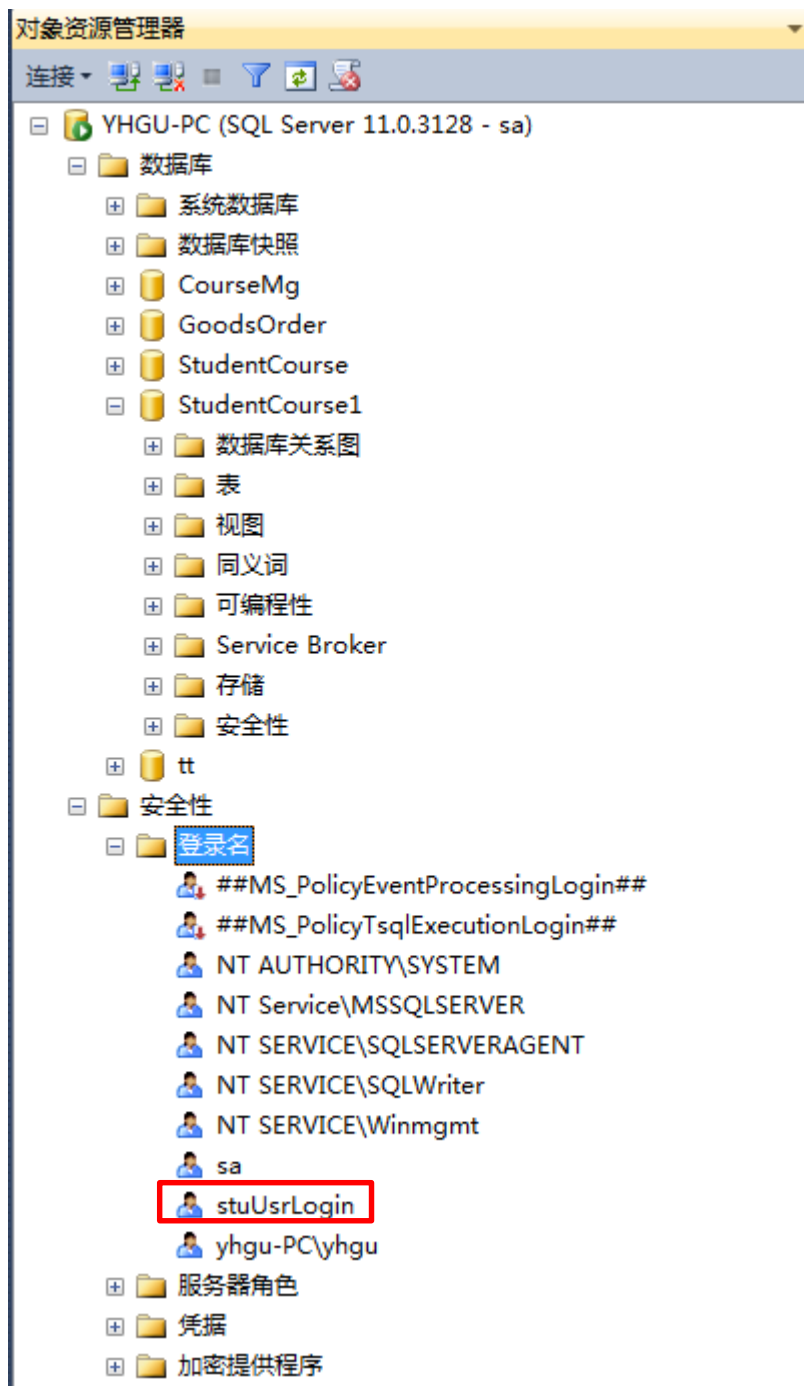
设置方法2：在对象资源管理器中数据库实例的右键快捷菜单项“属性”中单击“安全性”进行选择

2. 创建两个登录名：stuUsrLogin、stuUsrLogin2；

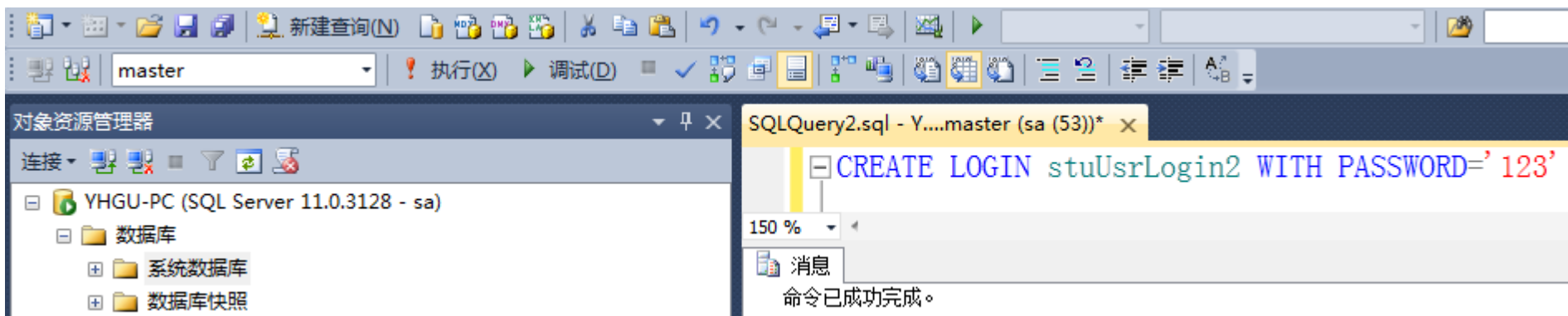
创建方法1：以sa登录，在对象资源管理器中单击“安全性”节点前的“+”，展开安全节点。在“登录名”上单击鼠标右键，在弹出的快捷菜单上选择“新建登录名”命令。选择身份验证模式为“SQL Server身份验证模式”，输入登录名、密码、确认密码等，单击“确定”按钮，即可创建登录。



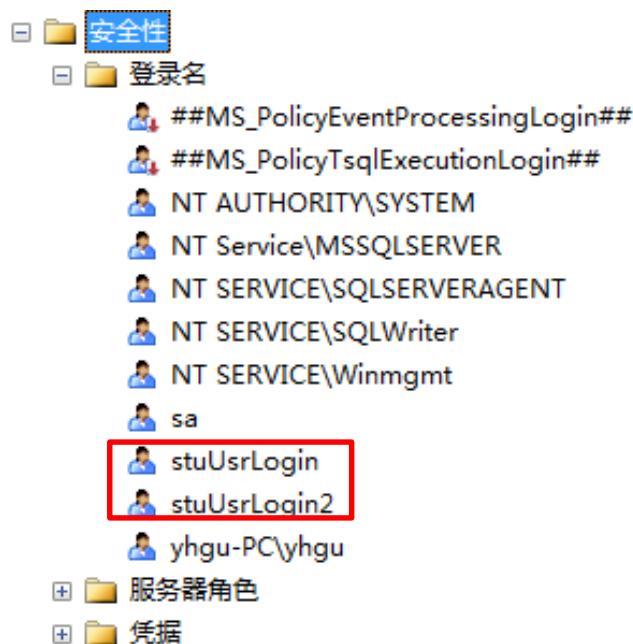
展开“安全性”节点即可查看到新建的登录名。



创建方法2: 以sa登录，在查询分析器中输入如下命令，即可创建登录名stuUsrLogin2。
CREATE LOGIN stuUsrLogin2 WITH PASSWORD='123'



展开“安全性”节点即可查看到新建的登录名。



3. 以sa登录，分别为登录名stuUsrLogin、stuUsrLogin2创建映射两个数据库用户stuUsr、stuUsr2，使这两个登录名可访问StudentCourse数据库，并授予他们可以查询Student表的权限。语句如下：

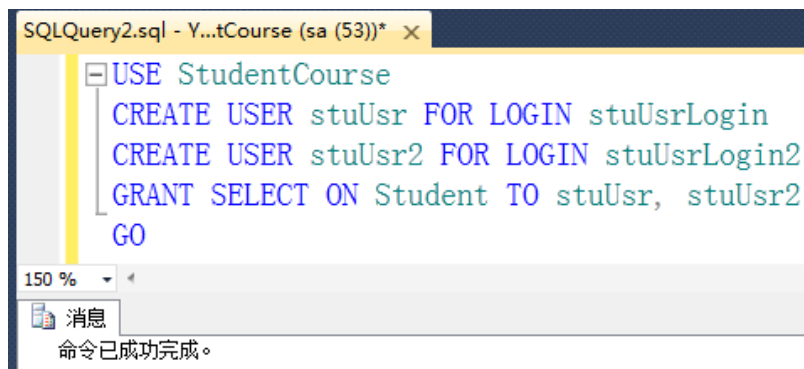
```
USE StudentCourse
```

```
CREATE USER stuUsr FOR LOGIN stuUsrLogin
```

```
CREATE USER stuUsr2 FOR LOGIN stuUsrLogin2
```

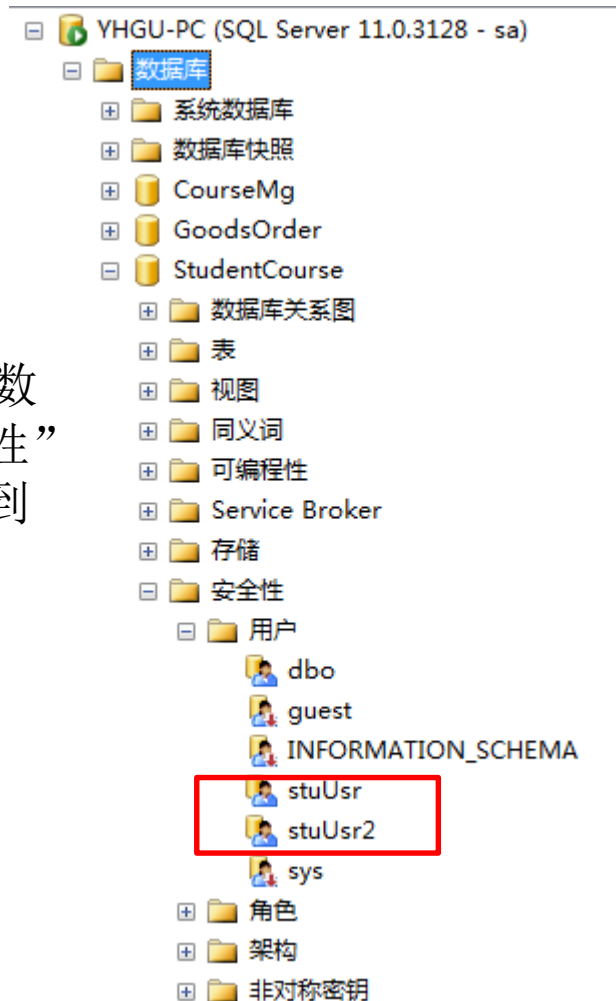
```
GRANT SELECT ON Student TO stuUsr, stuUsr2
```

```
GO
```



The screenshot shows a SQL query window titled 'SQLQuery2.sql - Y...tCourse (sa (53))'. The query text is:
USE StudentCourse
CREATE USER stuUsr FOR LOGIN stuUsrLogin
CREATE USER stuUsr2 FOR LOGIN stuUsrLogin2
GRANT SELECT ON Student TO stuUsr, stuUsr2
GO
Below the query window, a message box indicates '命令已成功完成。' (Command completed successfully).

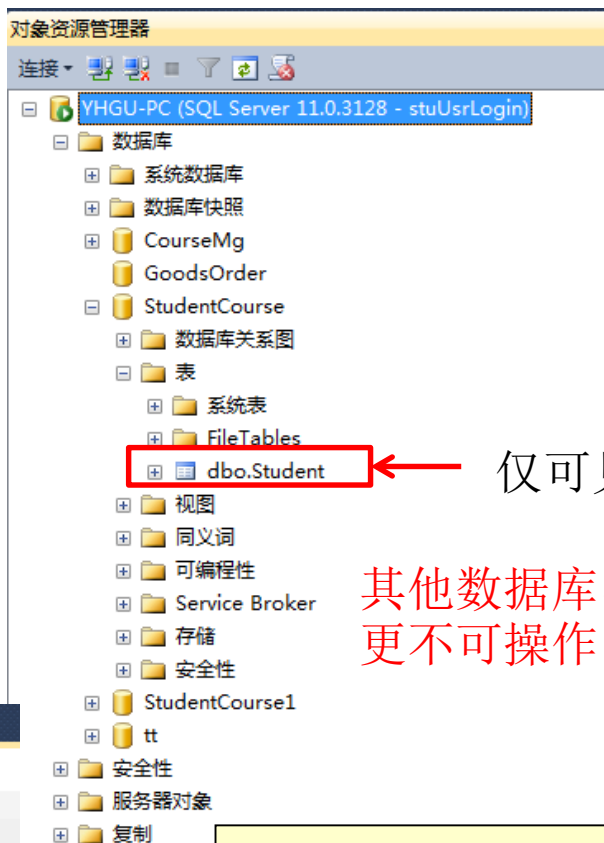
展开StuCourse数据库的“安全性”节点即可查看到新建的用户。



4. 验证第3步的授权。

重新以stuUsrLogin登录SQL Server，查看数据库、表信息。

类似可验证stuUsr2用户的权限。



仅可见Student表

其他数据库、表均不可见，
更不可操作

仅可查询Student表

SQLQuery1.sql - Y...stuUsrLogin (52) x

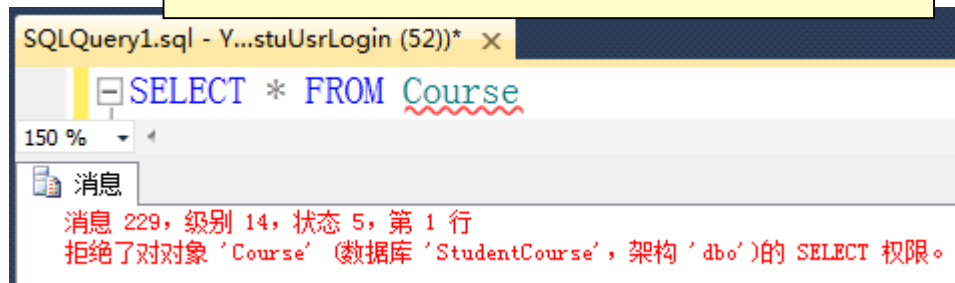
```
SELECT * FROM Student
```

150 %

结果 消息

	学号	姓名	专业名	性别	出生时间	总学分	备注
1	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好学生
2	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL
3	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目
4	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL
5	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获奖学金
6	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL
7	070207	朱平平	电子信息工程	女	1990-01-10 00:00:00	74	NULL
8	070208	李进	电子信息工程	男	1989-09-12 00:00:00	74	NULL
9	070501	张林	计算机科学与技术	男	1988-10-10 00:00:00	80	成绩优秀

不可查询其他表

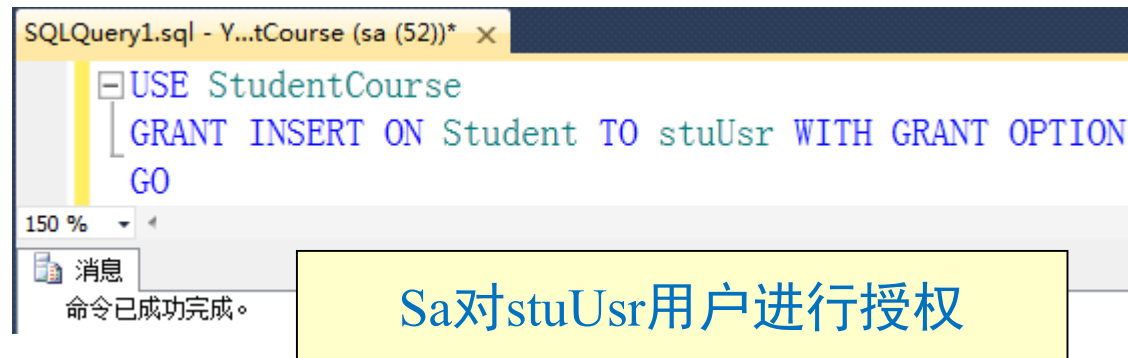


5. 重新以sa登录，将对Student表的INSERT权限授予stuUsr，并允许他将此权限再授予其他用户。语句如下：

USE StudentCourse

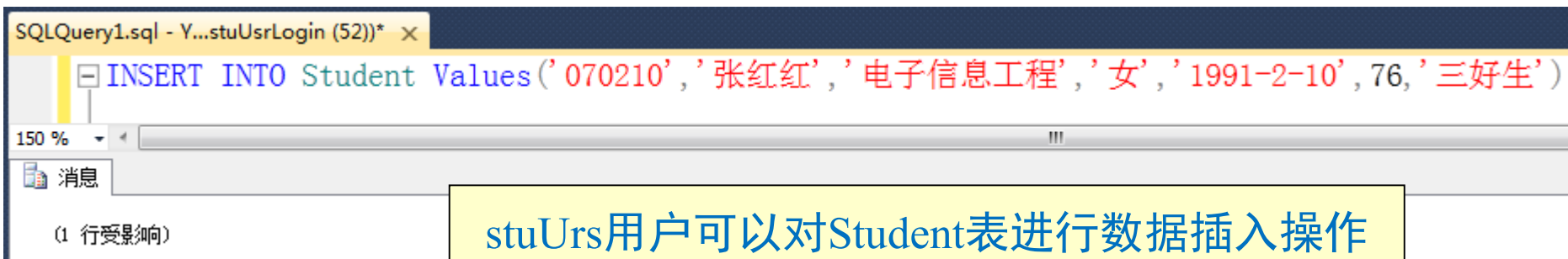
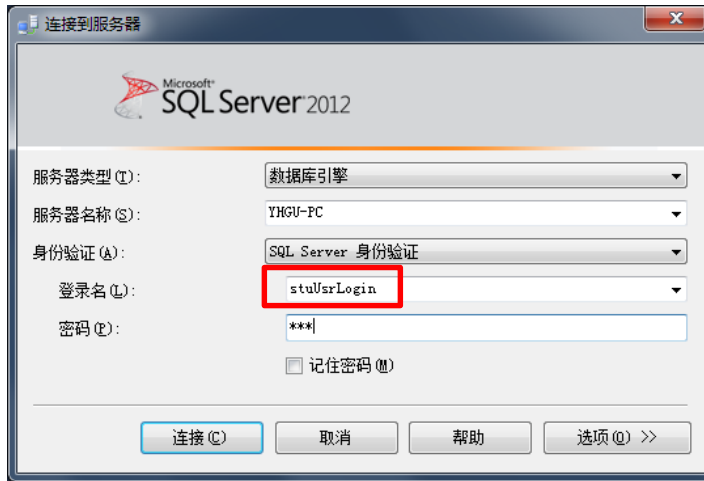
GRANT INSERT ON Student TO stuUsr WITH GRANT OPTION

GO



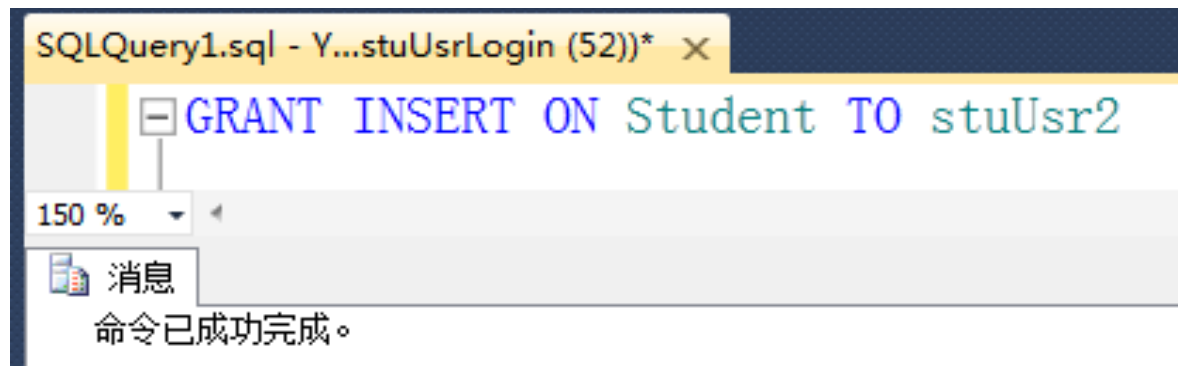
6. 重新以stuUsrLogin登录，向Student表的插入一条记录，能够正常插入记录，表明权限授予正确。语句如下：

```
INSERT INTO Student Values('070210','张红红','电子信息工程','女','1991-2-10',76,'三好生')
```



7. 将Student表的插入权限授予stuUsr2，语句如下：

GRANT INSERT ON Student TO stuUsr2



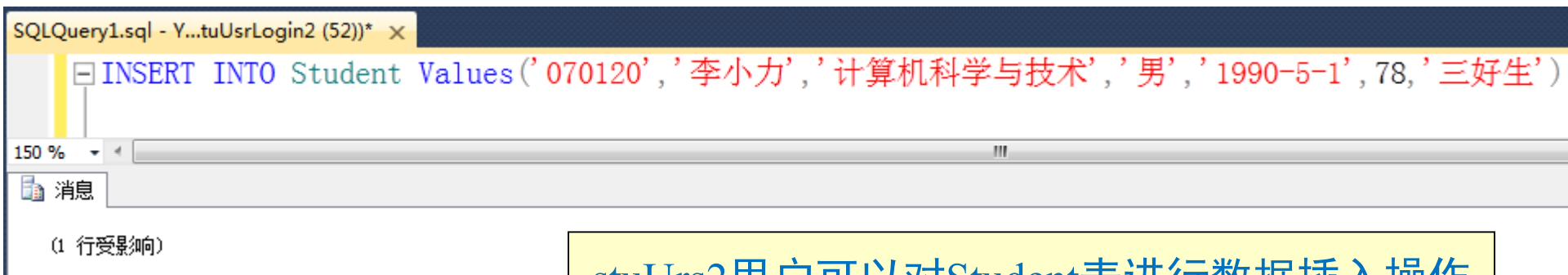
stuUsr用户将对Student表进行数据插入操作的权限再授予stuUsr2

8. 验证对stuUrs2的授权。重新以stuUrsLogin2登录:



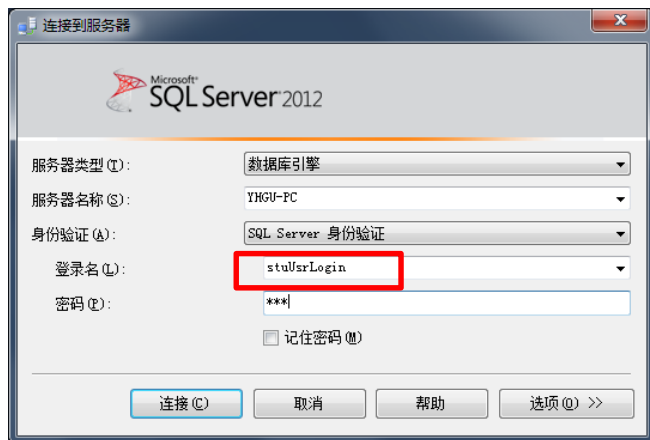
向Student表的插入一条记录，能够正常插入记录，表明权限授予正确。
语句如下：

```
INSERT INTO Student Values('070120','李小力','计算机科学与技术','男','1990-5-1',78,'三好生')
```



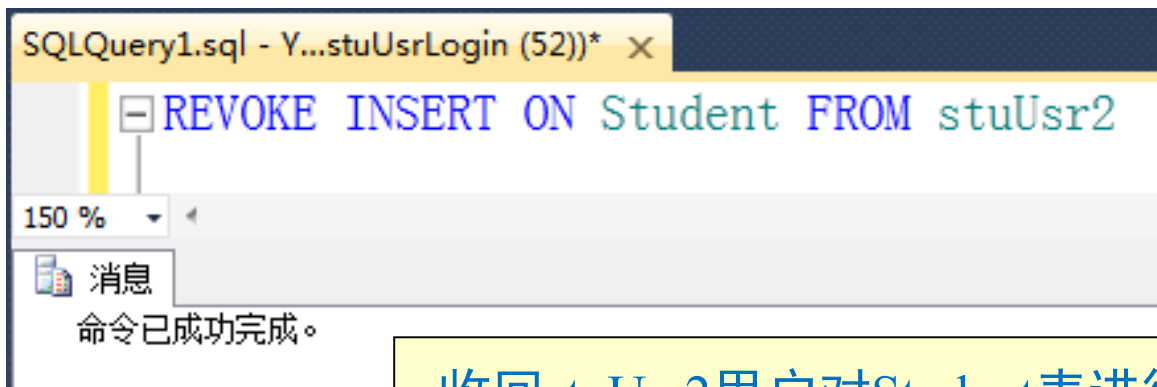
stuUrs2用户可以对Student表进行数据插入操作

9. 把授予stuUrs2用户对Student表插入的权限收回，重新以stuUrsLogin登录：



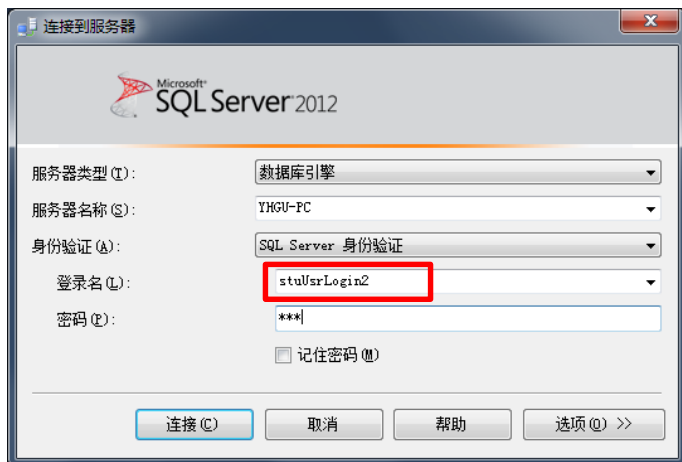
在查询分析器中输入以下语句将收回授权：

REVOKE INSERT ON Student FROM stuUrs2



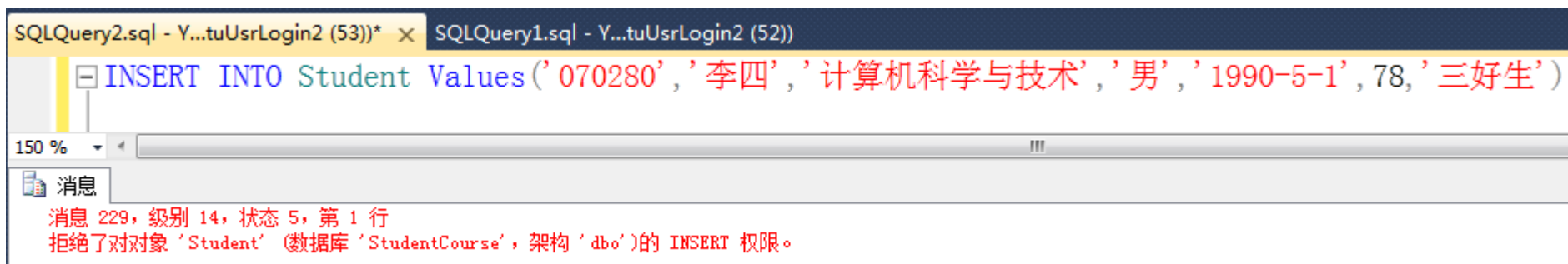
收回stuUrs2用户对Student表进行数据插入操作权限

10. 验证对stuUrs2用户权限收回操作是否成功。重新以stuUrsLogin2登录:



向Student表的插入一条记录，不能插入记录，表明权限回收正确。语句如下：

```
INSERT INTO Student Values('070280','李四','计算机科学与技术','男','1990-5-1',78,'三好生')
```



成功收回stuUrs2用户对Student表进行数据插入操作权限

三、实验报告

- 本次课实验报告：请参照本PPT的实验步骤和内容撰写并提交实验报告，按实验报告格式模板（可在线上课程“10.3”节或资料模块下载），形成WORD文件，提交电子稿。
- 提交方式：在老师发布的名为“数据库课程实验9—数据库完整性和安全性”的作业中提交，注意以WORD附件提交，不要提交拍照的图片。
- 提交截止时间：（具体见作业截止时间）。

实验报告要求

一、按照学校实验报告的格式要求书写；

二、报告内容包括：

1、实验目的

2、实验内容

3、实验过程与结果

4、实验总结

（附：实验报告模板）