

# 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。

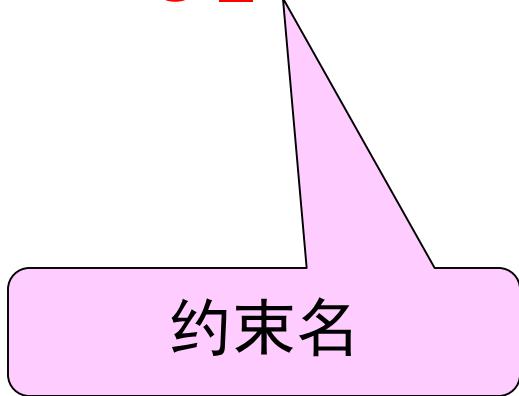
**【例】** 以下的表定义中限定Age字段只能取整数值，并且范围只能为0~100。

```
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,

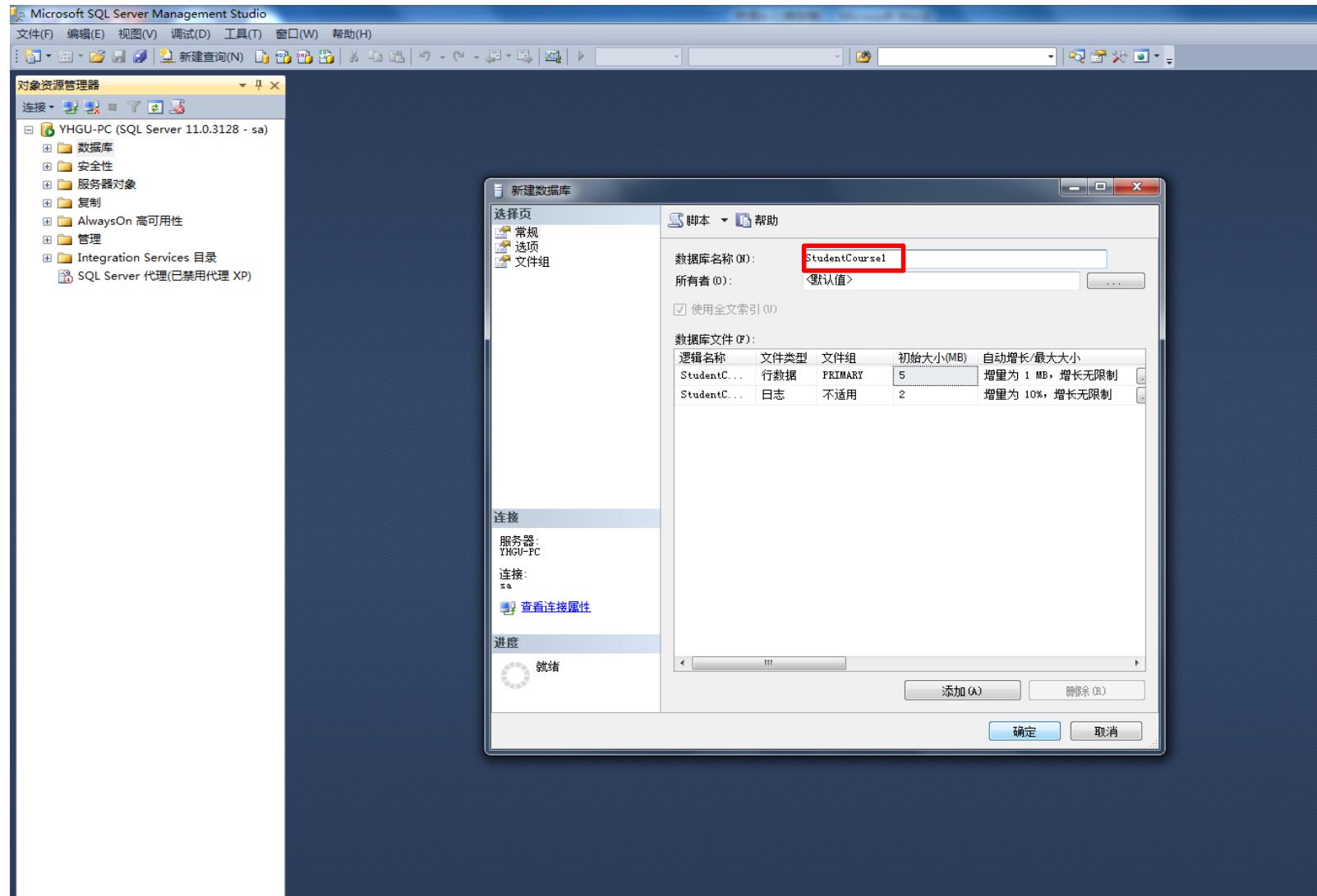
...

)

## 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：

The screenshot shows the Object Explorer pane of SQL Server Management Studio. The tree view displays the following structure for the 'StudentCourse1' database:

- StudentCourse1
- + 数据库关系图
- + 表
  - + 系统表
  - + FileTables
  - + dbo.Course
  - + dbo.StuCourse
  - + dbo.Student
  - + 列
    - 学号 (PK, char(6), not null) (highlighted with a red box)
    - 姓名 (char(12), not null)
    - 专业名 (varchar(20), null)
    - 性别 (char(2), not null)
    - 出生时间 (smalldatetime, null)
    - 总学分 (int, null)
    - 备注 (text, null)
  - + 键
  - + 约束
    - gen\_Check (highlighted with a red box)
    - tot\_Check
  - + 触发器
  - + 索引
    - name\_unique (唯一, 非聚集)
    - PK\_Student\_1CC396D2BD39E6CD (聚集) (highlighted with a red box)
  - + 统计信息

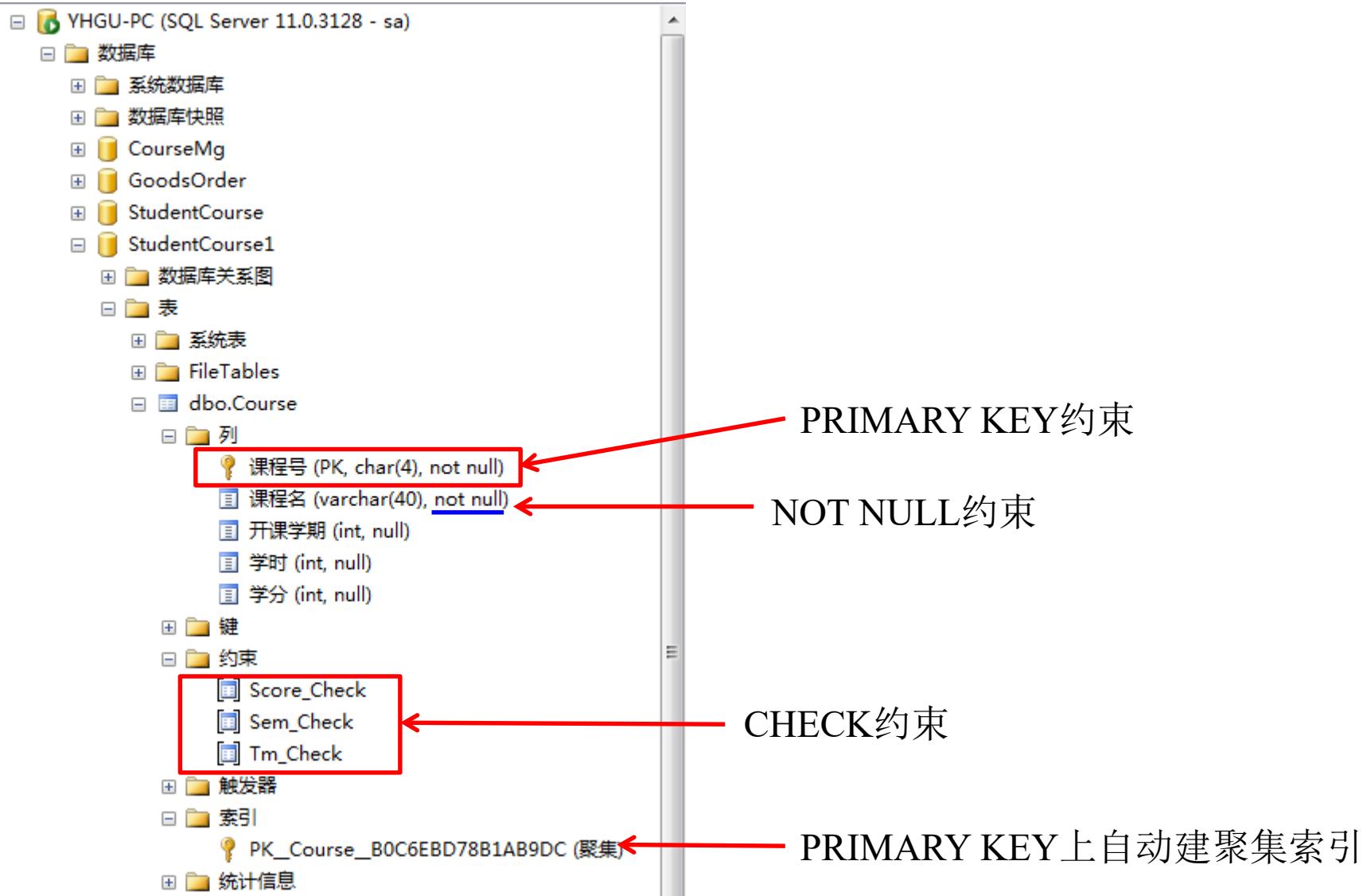
Annotations with arrows point to specific items:

- A red arrow points from the text "PRIMARY KEY约束" to the highlighted primary key constraint "学号 (PK, char(6), not null)".
- A red arrow points from the text "NOT NULL约束" to the three columns with NOT NULL constraints: "姓名 (char(12), not null)", "性别 (char(2), not null)", and "出生时间 (smalldatetime, null)".
- A red arrow points from the text "gen\_Check约束" to the "gen\_Check" constraint node.
- A red arrow points from the text "tot\_Check约束" to the "tot\_Check" constraint node.
- A red arrow points from the text "UNIQUE约束上自动建非聚集索引" to the "name\_unique" index node.
- A red arrow points from the text "PRIMARY KEY上自动建聚集索引" to the "PK\_Student\_1CC396D2BD39E6CD" index node.

A yellow callout bubble contains the text: "单击“修改”快捷菜单  
可查看和修改约束" (Right-click the "Modify" context menu to view and modify constraints).

A screenshot of the "CHECK 约束" (CHECK Constraint) dialog box is shown on the right side of the interface. It lists two constraints: "gen\_Check" and "tot\_Check". The "gen\_Check" constraint is selected, showing its properties:

(常规)	表达式	$(\text{性别} = \text{男} \text{ OR } \text{性别} = \text{女})$
标识	名称	gen_Check
说明		
表设计器	强制用于 INSERT 和 UPDATE 指令	是
	强制用于复制	是
	在创建或重新启用时检查现有值	是



The screenshot shows the SQL Server Object Explorer with the following tree structure:

- 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

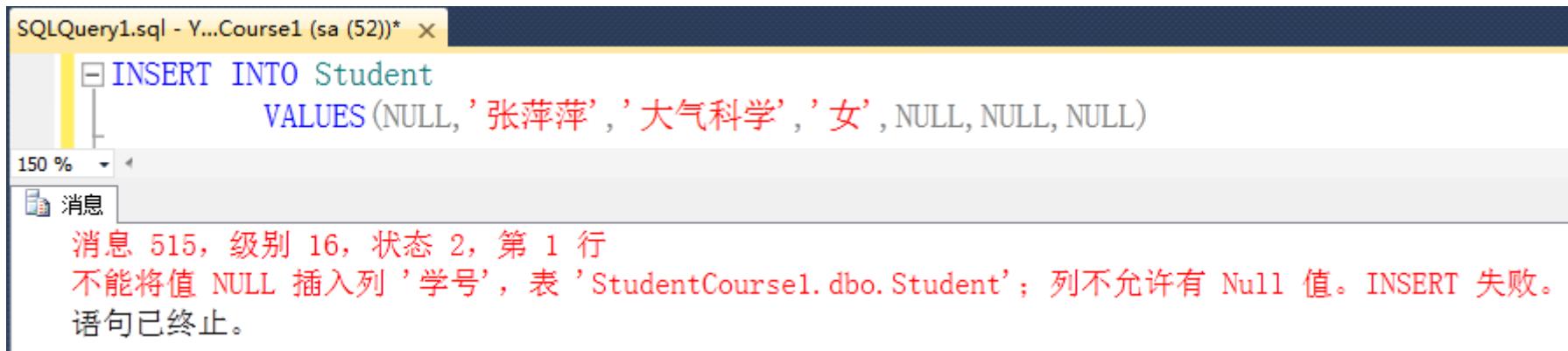
PRIMARY KEY上自动建聚集索引

# 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)
```



The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

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

The status bar indicates "150 %". Below the query window is a message window titled "消息" (Messages). It displays the following error message:

消息 515, 级别 16, 状态 2, 第 1 行  
不能将值 NULL 插入列 '学号', 表 'StudentCourse1.dbo.Student'; 列不允许有 Null 值。INSERT 失败。  
语句已终止。

“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)**

The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

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

The status bar indicates "150 %". Below the query window is a "消息" (Messages) window with the following error message:

消息 547, 级别 16, 状态 0, 第 1 行  
INSERT 语句与 FOREIGN KEY 约束"FK\_StuCourse\_学号\_1B0907CE"冲突。该冲突发生于数据库"StudentCourse1", 表"dbo.Student", column '学号'。  
语句已终止。

“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'**

The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

```
DELETE FROM Student WHERE 学号='070101'
```

The results pane, labeled "消息" (Messages), displays the following error message:

消息 547, 级别 16, 状态 0, 第 1 行  
DELETE 语句与 REFERENCE 约束"FK\_StuCourse\_学号\_1B0907CE"冲突。该冲突发生于数据库"StudentCourse1", 表"dbo.StuCourse", column '学号'。  
语句已终止。

“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)**

The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

```
INSERT INTO Student  
VALUES('070501','丁一平','大气科学','女',NULL,NULL,NULL)
```

The "消息" (Messages) tab displays the following error message:

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'**

The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

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

The "消息" (Messages) tab displays the following error message:

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)
```

The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

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

The results pane shows an error message:

消息 547, 级别 16, 状态 0, 第 1 行  
INSERT 语句与 CHECK 约束“tot\_Check”冲突。该冲突发生于数据库“StudentCourse1”，表“dbo.Student”，column ‘总学分’。  
语句已终止。

A yellow callout box highlights the error message: “‘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)
```

The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

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

The results pane shows an error message:

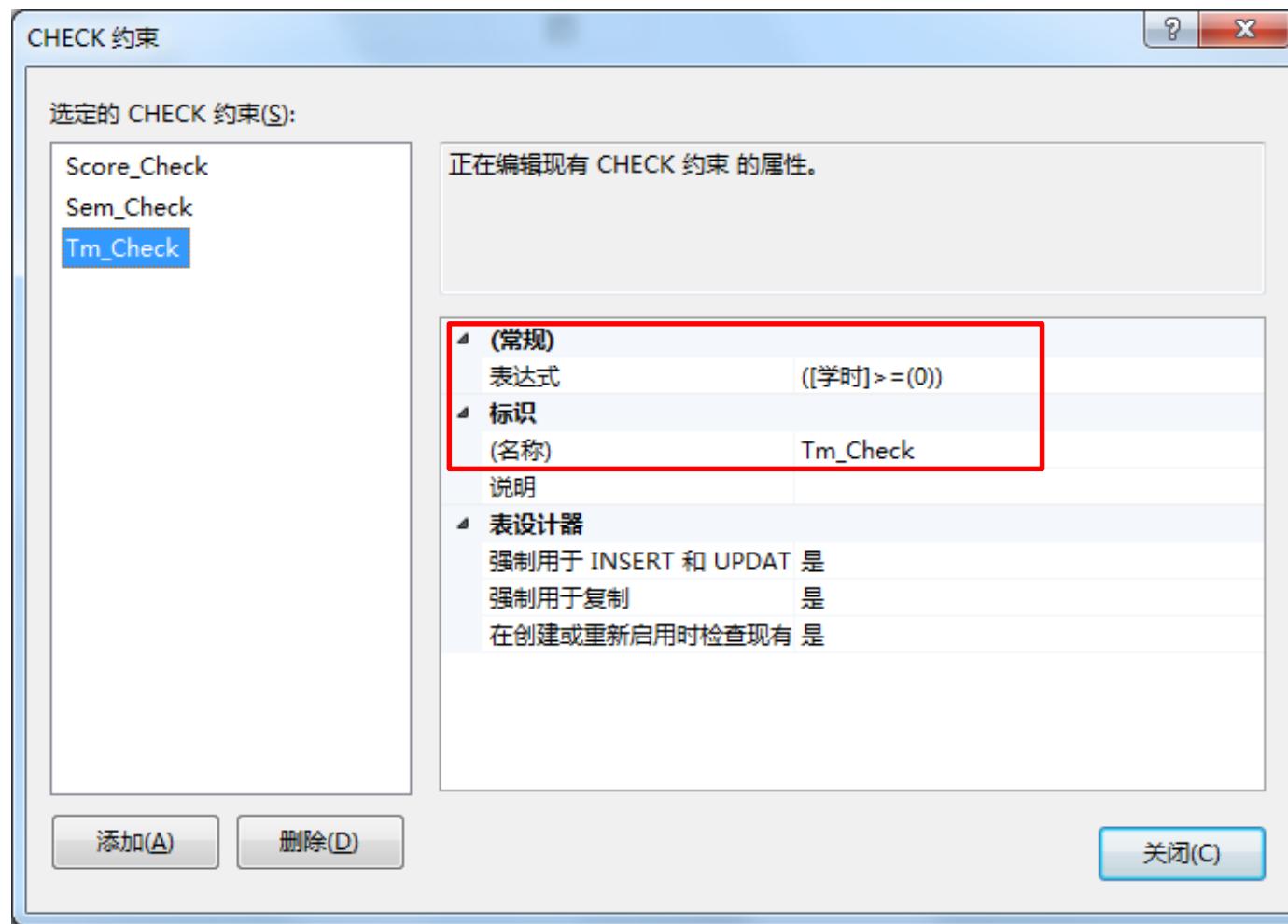
消息 515, 级别 16, 状态 2, 第 1 行  
不能将值 NULL 插入列 '课程名', 表 'StudentCourse1.dbo.Course'；列不允许有 Null 值。INSERT 失败。  
语句已终止。

“课程名”字段值违反了NOT NULL约束

## 4. 修改约束定义

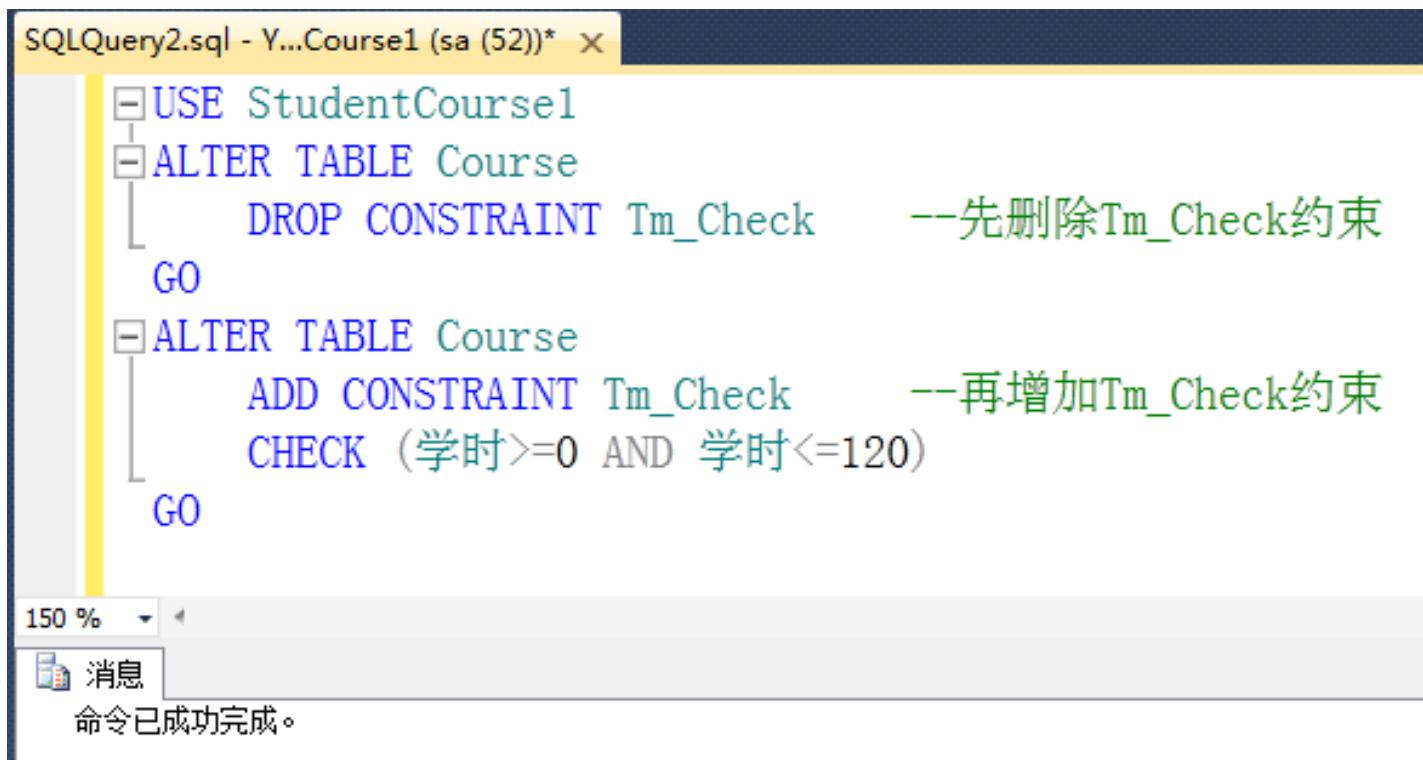
要求：将Course表的Tm\_Check约束修改为大于等于0，且小于等于120

(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
```



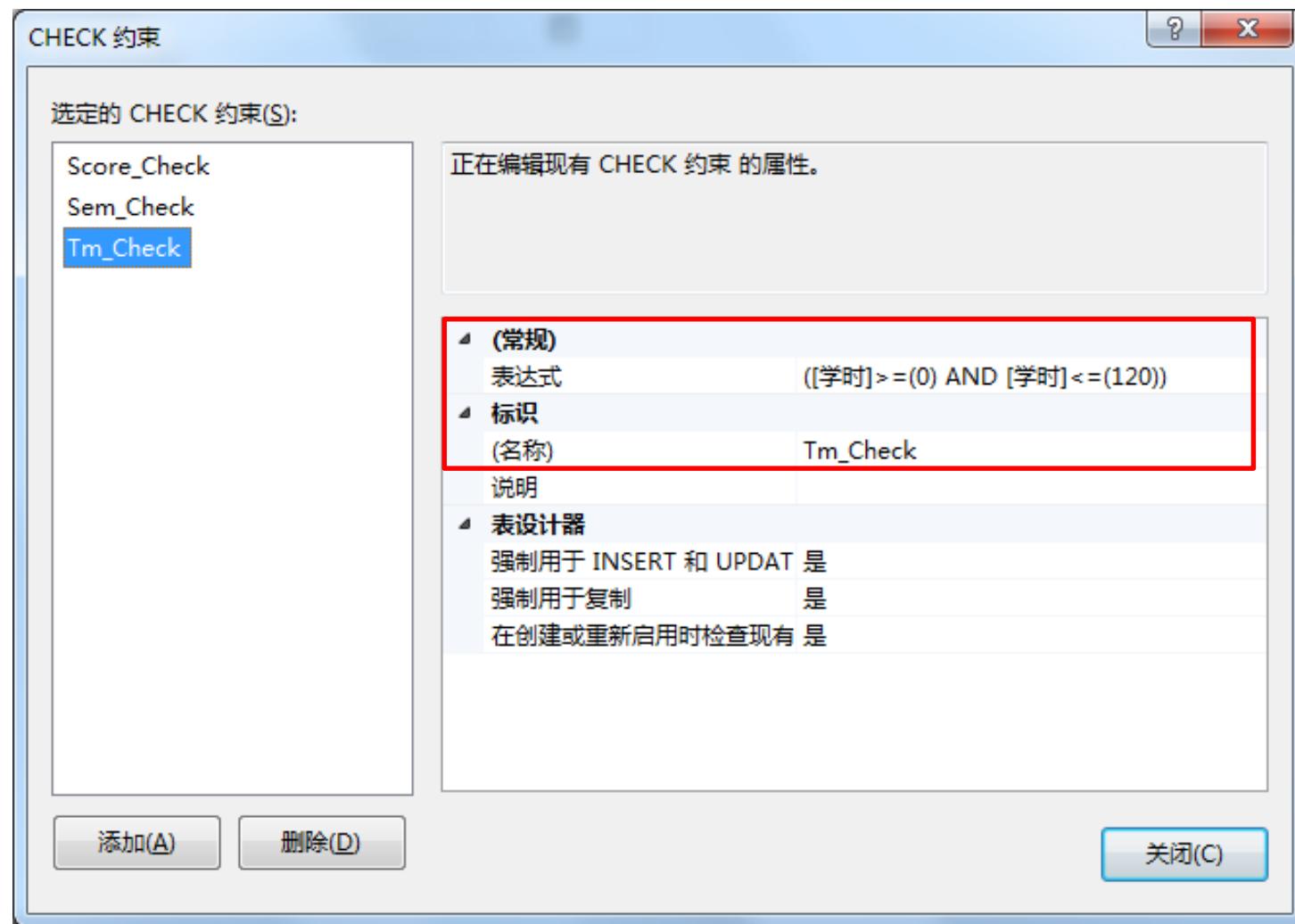
```
SQLQuery2.sql - Y...Course1 (sa (52))*
USE StudentCourse1
ALTER TABLE Course
    DROP CONSTRAINT Tm_Check    --先删除Tm_Check约束
GO
ALTER TABLE Course
    ADD CONSTRAINT Tm_Check    --再增加Tm_Check约束
    CHECK (学时>=0 AND 学时<=120)
GO
```

150 %

消息

命令已成功完成。

### (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
```

The screenshot shows a SQL query window titled "SQLQuery2.sql - Y...Course1 (sa (52))". The query is:

```
USE StudentCourse1
ALTER TABLE Course
    ADD CONSTRAINT Score_default      --增加默认值约束Score_default
        Default(2) FOR 学分
GO
```

The code is highlighted with syntax coloring. The message bar at the bottom says "命令已成功完成。" (Command completed successfully).

## (2) 查看增加的约束内容:

The screenshot shows the Object Explorer on the left and the Table Designer on the right.

**Object Explorer:**

- 展开数据库 > StudentCourse
- 展开表 > dbo.Course
- 展开约束 > Score\_default (高亮显示)
- 触发器 (带有指向 Score\_default 的红色箭头)

**Table Designer (dbo.Course):**

列名 | 数据类型 | 允许 Null 值

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: 学分) > 默认值或绑定: ((2))

其他约束:

- Score\_Check
- Sem\_Check
- Tm\_Check

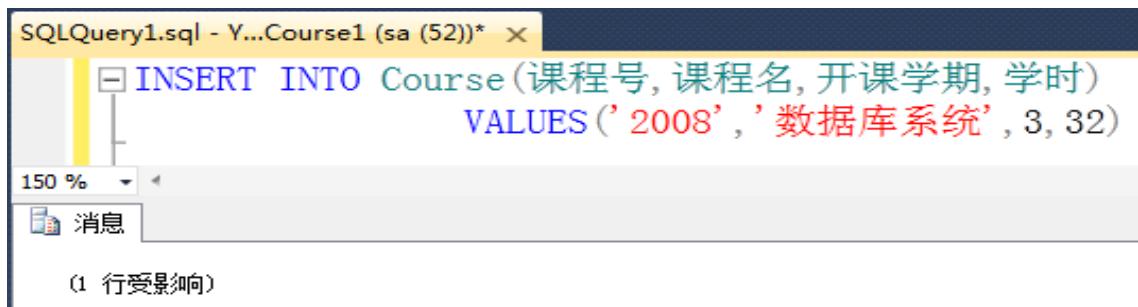
增加的约束

### (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)**



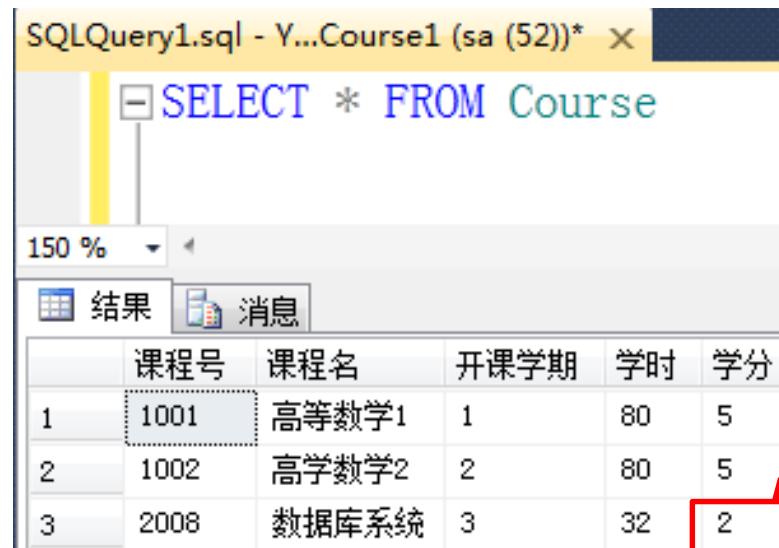
The screenshot shows the SQL Query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

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

The status bar at the bottom indicates "(1 行受影响)".

在查询分析器中输入查询语句并执行：

**SELECT \* FROM Course**



The screenshot shows the SQL Query window titled "SQLQuery1.sql - Y...Course1 (sa (52))". The query is:

```
SELECT * FROM Course
```

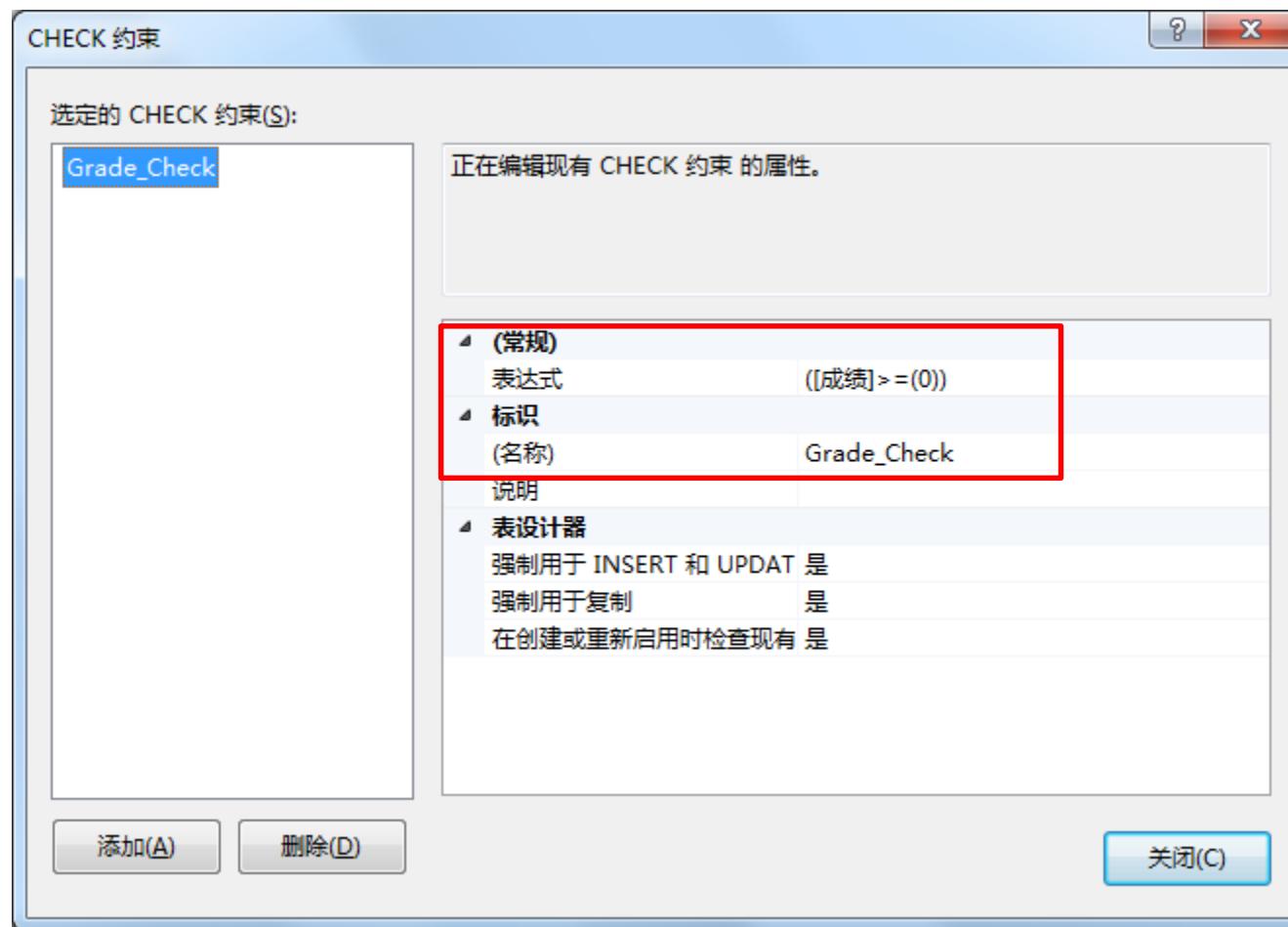
The results are displayed in a table:

	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

A red arrow points to the last row's "学分" column value "2", with the text "取默认值" (Get default value) next to it.

## 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:

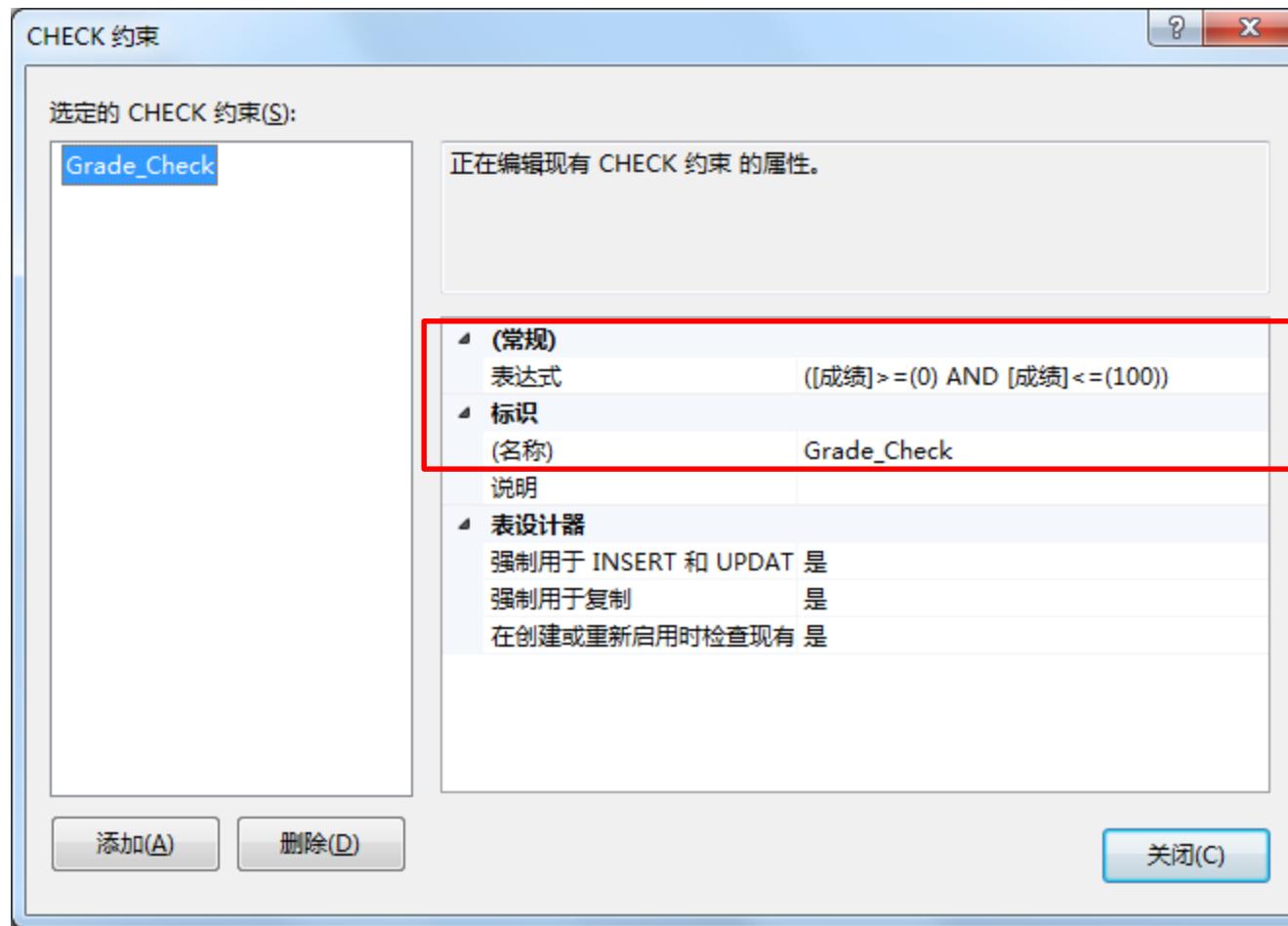
```
USE StudentCourse1
--检查是否存在同名的约束，若有，则删除。
IF EXISTS (SELECT name FROM sysobjects
            WHERE name = 'Grade_Check' AND type = 'C')
    ALTER TABLE StuCourse
        DROP CONSTRAINT Grade_Check --先删除Grade_Check约束
GO
ALTER TABLE StuCourse
    ADD CONSTRAINT Grade_Check      --再增加Grade_Check约束
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

The screenshot shows the SQL Query Editor window titled "SQLQuery2.sql - Y...Course1 (sa (52))". The code is displayed in a tree-like structure with expandable nodes. The code is identical to the one provided in the text above, including the comments and the GO statements.

```
USE StudentCourse1
--检查是否存在同名的约束，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
```

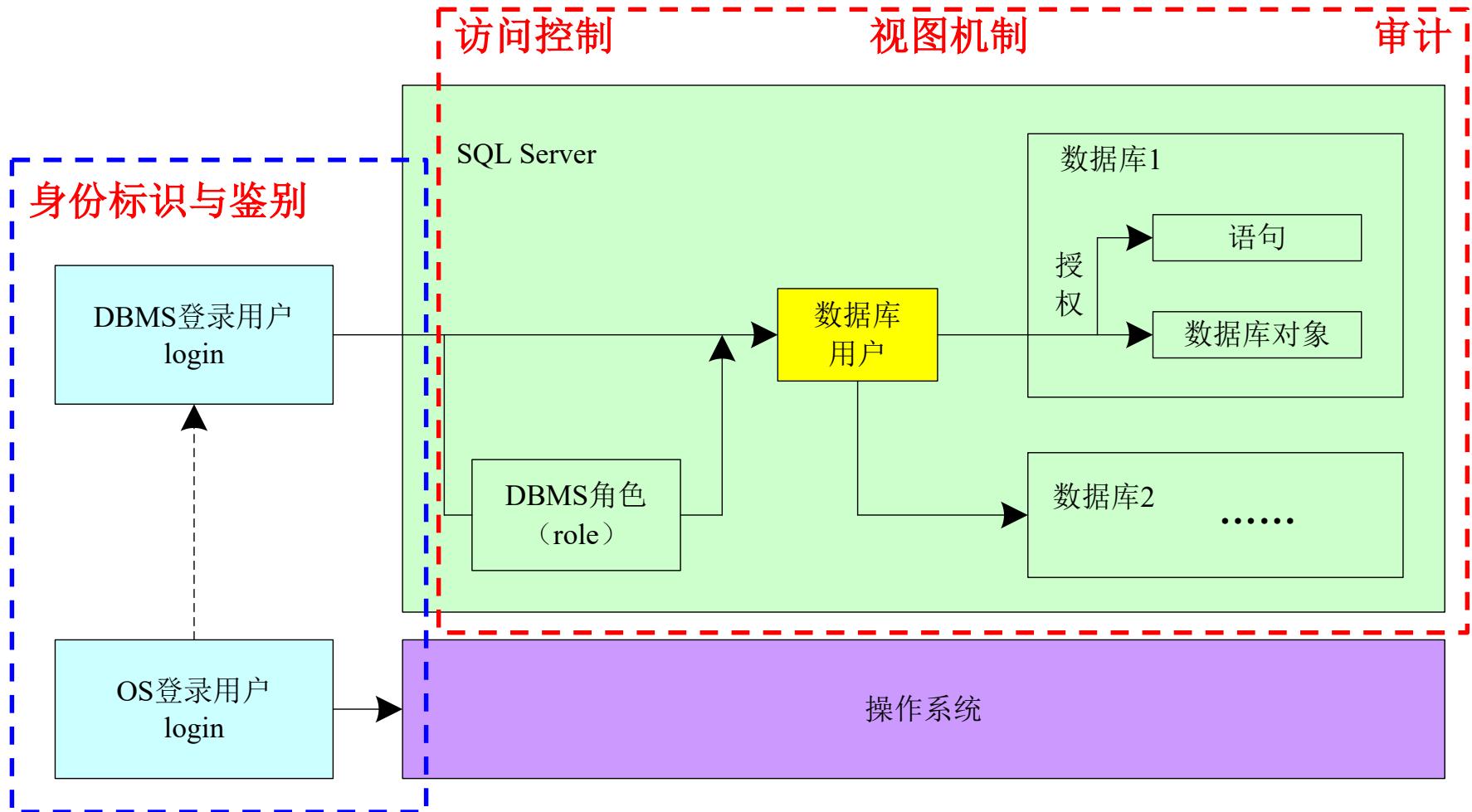
The status bar at the bottom of the editor shows "150 %". In the bottom-left corner, there is a message icon with the text "消息" (Messages). Below it, a message box displays the text "命令已成功完成。" (The command has been successfully completed.).

### (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：

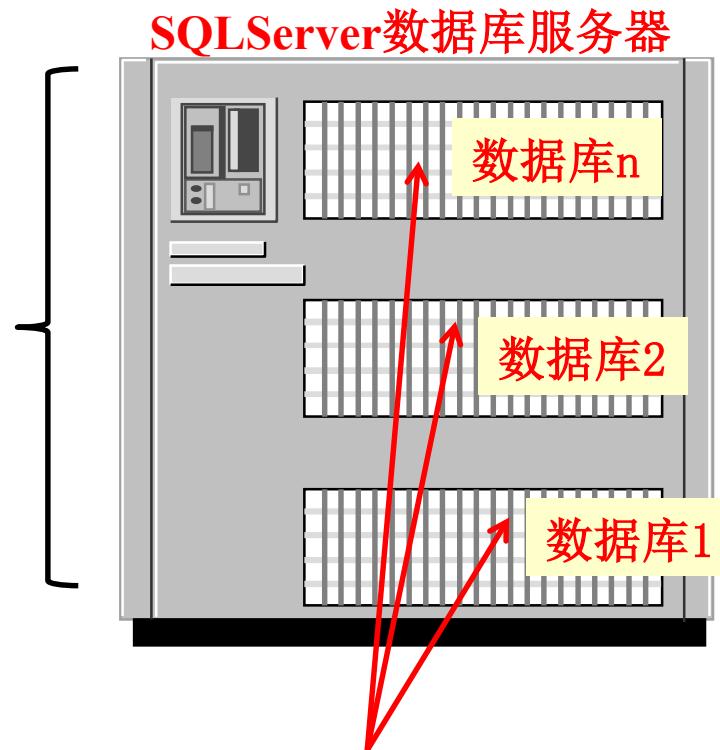
- ✓ **Windows身份验证模式：** 使用Windows操作系统的安全机制进行身份验证。只要用户能够通过Windows用户账号验证，即可连接到SQL Server，SQL Server不再进行验证。
- ✓ **SQL Server和Windows身份验证模式：** 在这种模式下，SQL Server要进行身份验证。用户必须提供登录名和口令，这些信息存储在SQL Server的系统表中。

# 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)**

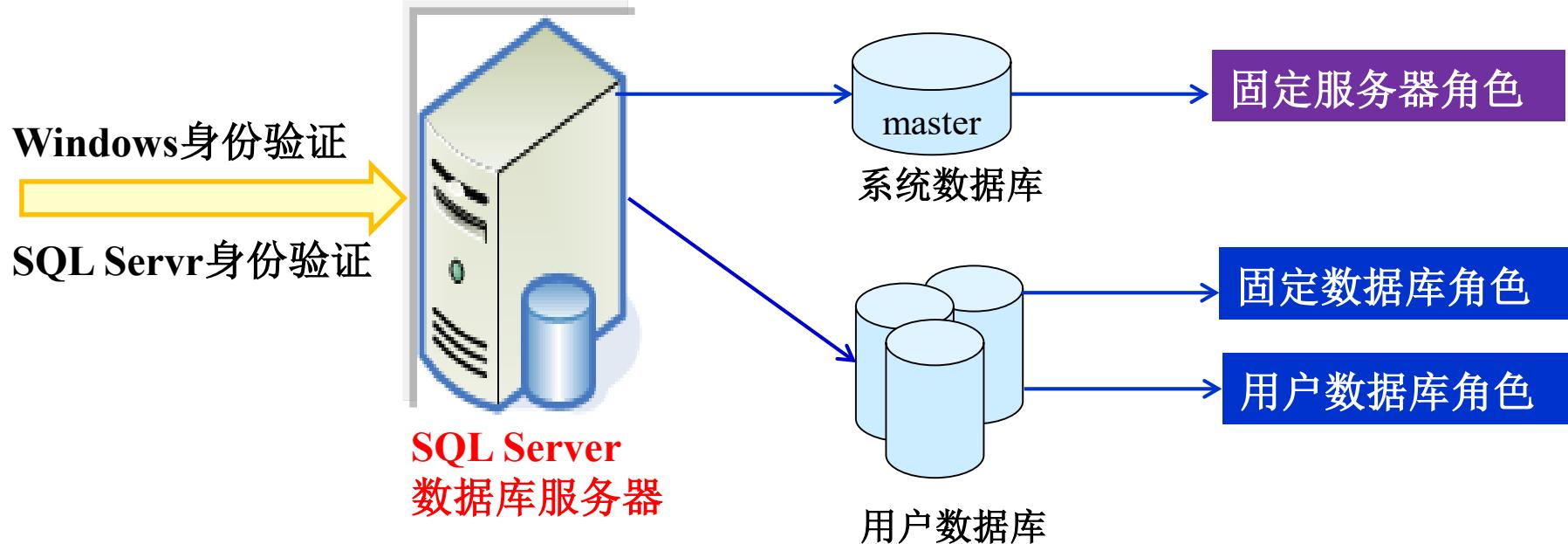
- ✓ 登录到服务器时，SQL Server验证登录；
- ✓ 当访问数据库时，SQL Server将验证用户。



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

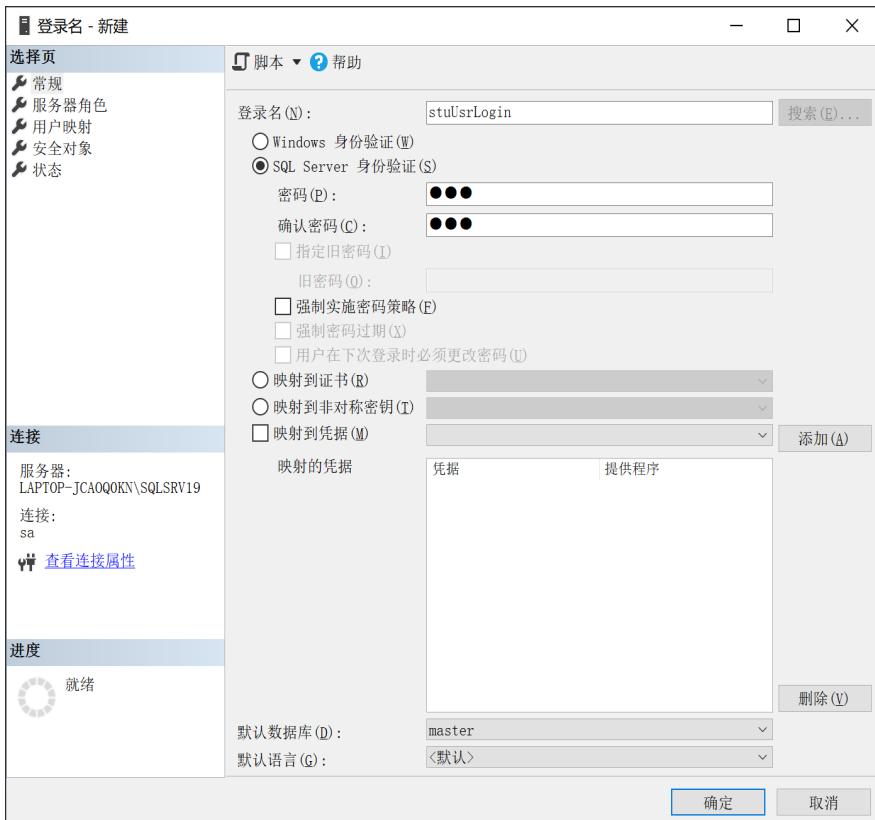
## ✓ 服务器权限

- ✧ 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

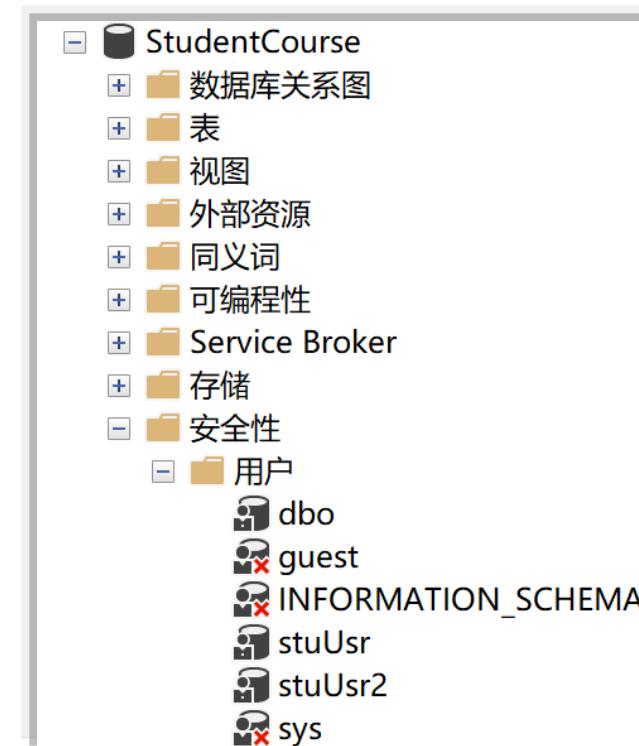
# 创建登录



USE StudentCourse

CREATE USER stuUsr FOR LOGIN stuUsrLogin  
CREATE USER stuUsr2 FOR LOGIN stuUsrLogin2

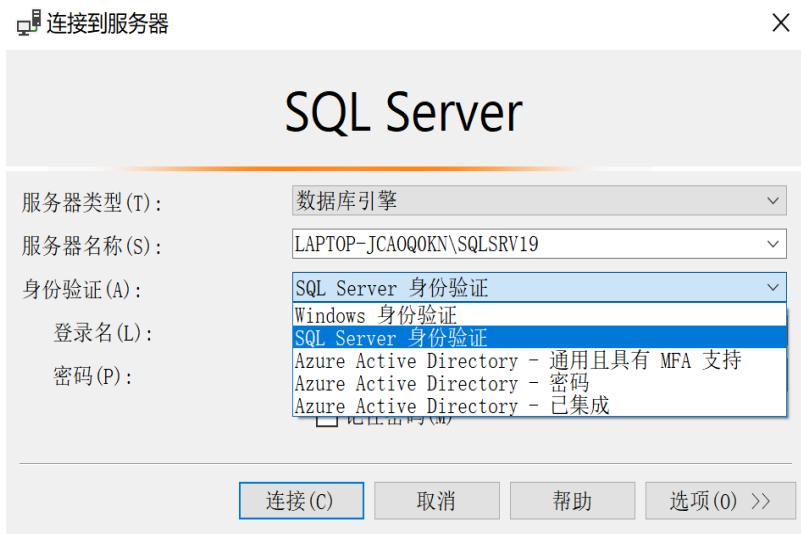
**GRANT SELECT ON Student TO stuUsr, stuUsr2**  
GO



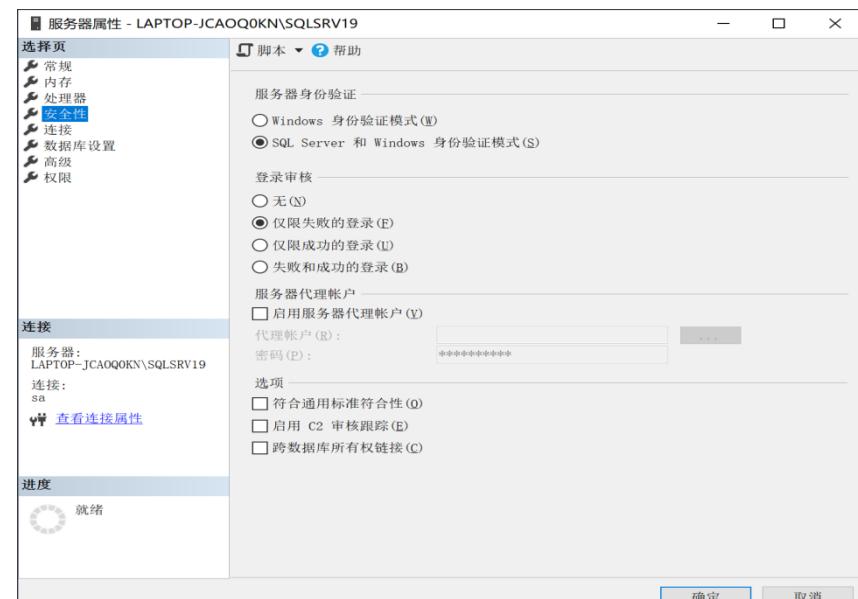
创建登录，通常还需要映射到数据库用户，方可访问数据库对象

## (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”；



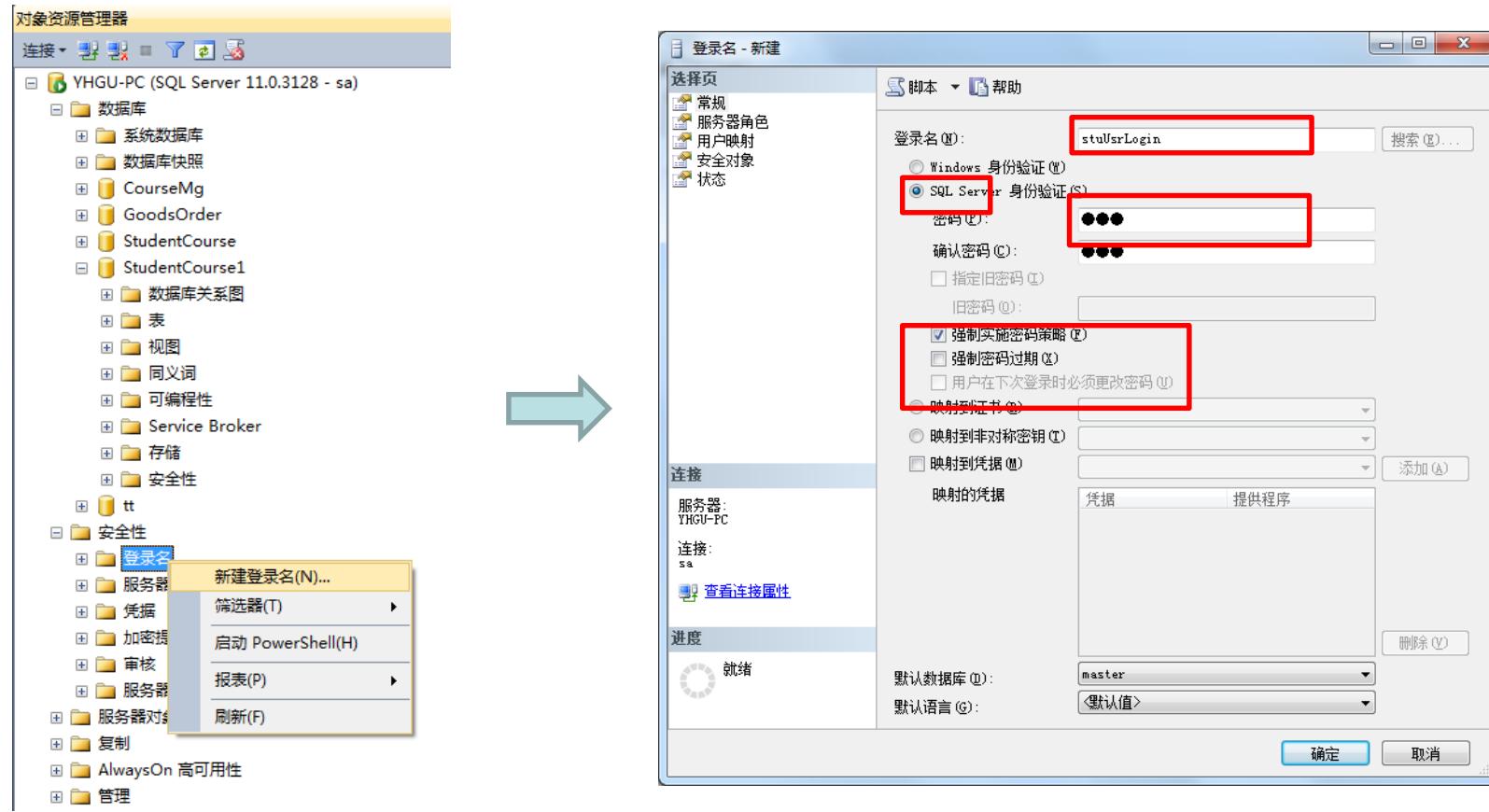
设置方法1：在启动窗口中选择



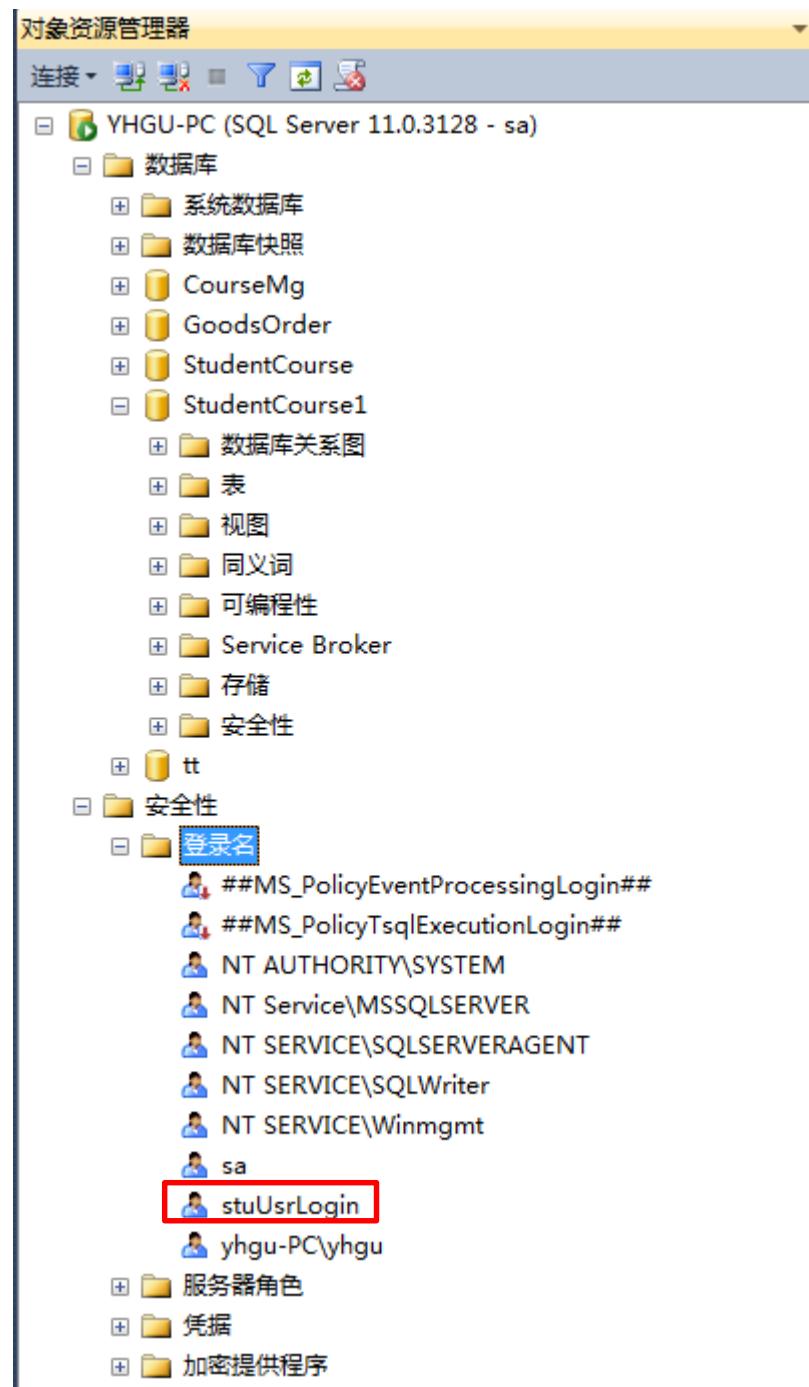
设置方法2：在对象资源管理器中数据库实例的右键快捷菜单项“属性”中单击“安全性”进行选择

## 2. 创建两个登录名：stuUsrLogin、stuUsrLogin2；

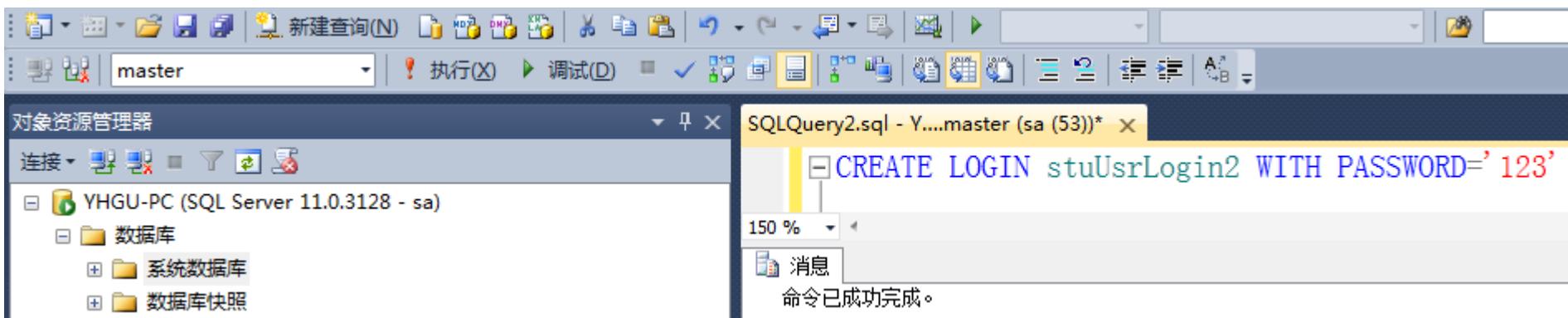
**创建方法1：**以sa登录，在对象资源管理器中单击“安全性”节点前的“+”，展开安全节点。在“登录名”上单击鼠标右键，在弹出的快捷菜单上选择“新建登录名”命令。选择身份验证模式为“SQL Server身份验证模式”，输入登录名、密码、确认密码等，单击“确定”按钮，即可创建登录。



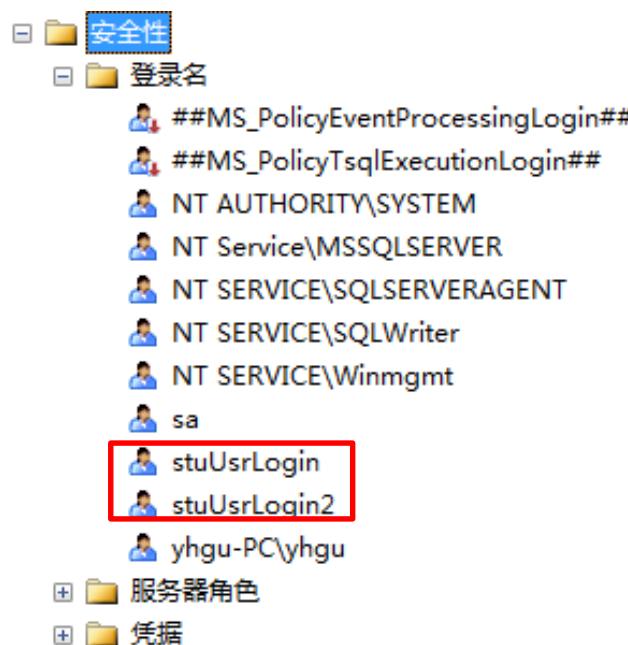
展开“安全性”节点即可查看到新建的登录名。



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
```

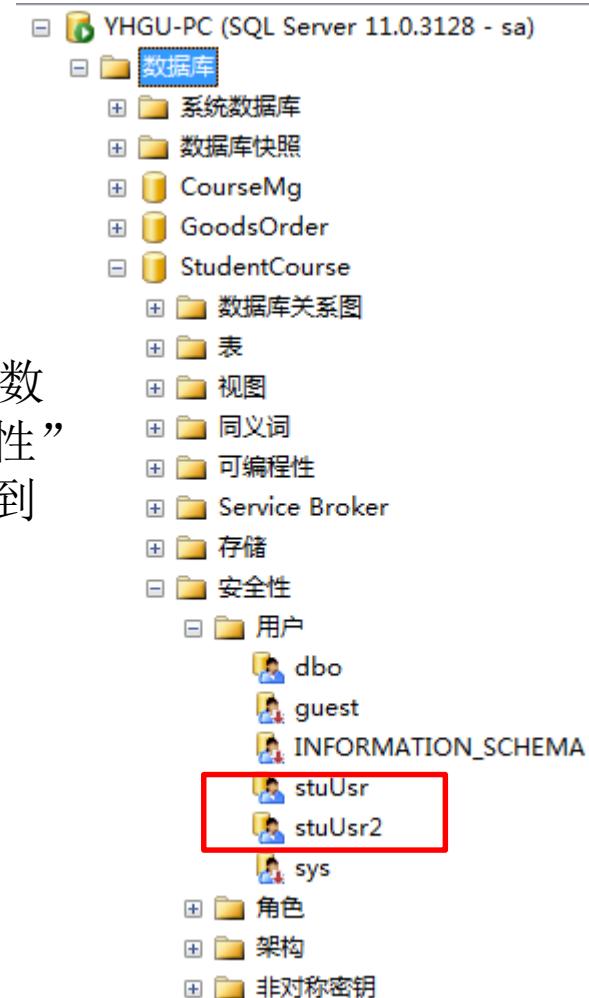
SQLQuery2.sql - Y...tCourse (sa (53))\*

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

150 %

消息 命令已成功完成。

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。



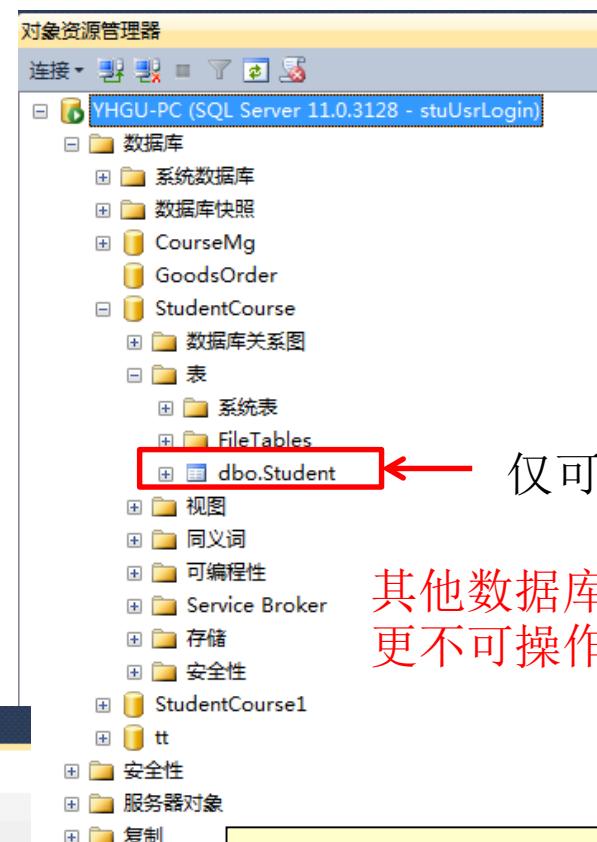
仅可查询Student表

SQLQuery1.sql - Y...stuUsrLogin (52) \*

```
SELECT * FROM Student
```

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



仅可见Student表  
其他数据库、表均不可见，  
更不可操作

不可查询其他表

SQLQuery1.sql - Y...stuUsrLogin (52) \*

```
SELECT * FROM Course
```

150 % 消息

消息 229, 级别 14, 状态 5, 第 1 行  
拒绝了对对象 'Course' (数据库 'StudentCourse', 架构 'dbo') 的 SELECT 权限。

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
```



```
SQLQuery1.sql - Y...tCourse (sa (52))*
```

```
USE StudentCourse  
GRANT INSERT ON Student TO stuUsr WITH GRANT OPTION  
GO
```

消息  
命令已成功完成。

Sa对stuUsr用户进行授权

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')
```



SQLQuery1.sql - Y...stuUsrLogin (52)\*

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

150 %

消息

(1 行受影响)

stuUsrLogin用户可以对Student表进行数据插入操作

7. 将Student表的插入权限授予stuUsr2，语句如下：

GRANT INSERT ON Student TO stuUsr2

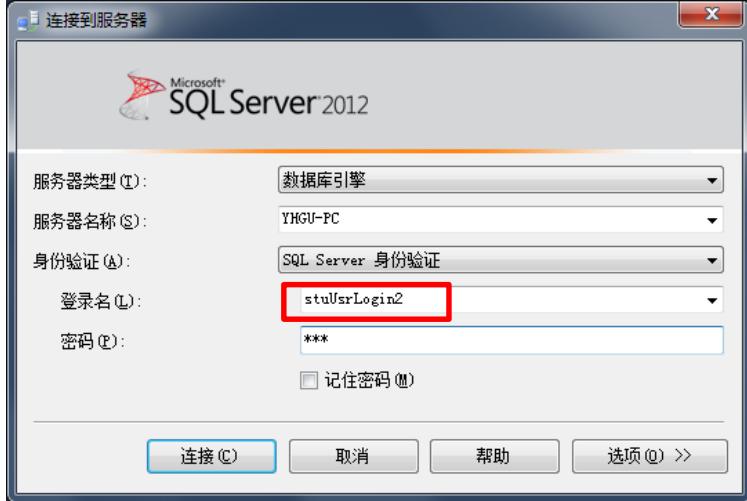
The screenshot shows a SQL query window titled "SQLQuery1.sql - Y...stuUsrLogin (52)\*". The query itself is:

```
GRANT INSERT ON Student TO stuUsr2
```

Below the query, there is a message window titled "消息" (Message) containing the text "命令已成功完成。" (The command has been successfully completed.).

stuUrs用户将对Student表进行数据插入操作的权限再授予stuUrs2

## 8. 验证对stuUsr2的授权。重新以stuUsrLogin2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')
```

SQLQuery1.sql - Y...tuUsrLogin2 (52)\*

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

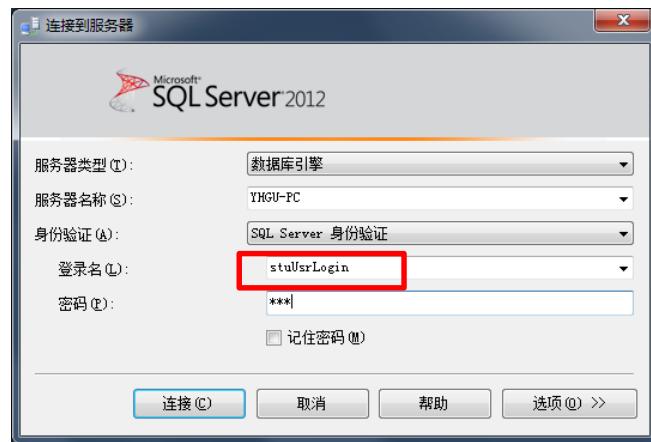
150 %

消息

(1 行受影响)

stuUsr2用户可以对Student表进行数据插入操作

## 9. 把授予stuUsr2用户对Student表插入的权限收回，重新以stuUsrLogin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 stuUsr2

SQLQuery1.sql - Y...stuUsrLogin (52)\*

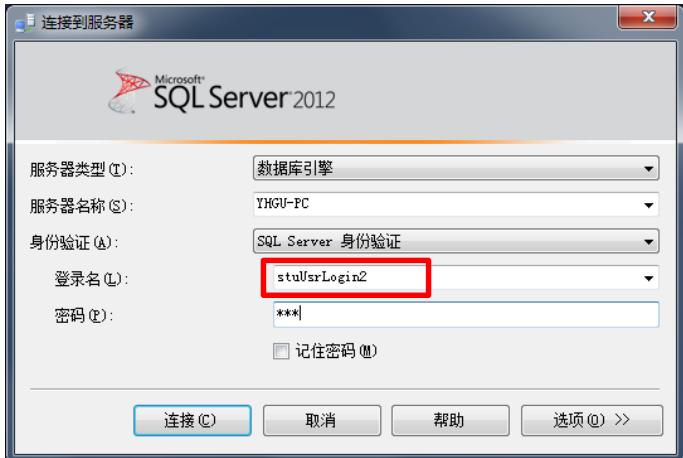
```
REVOKE INSERT ON Student FROM stuUsr2
```

150 %

消息 命令已成功完成。

收回stuUsr2用户对Student表进行数据插入操作权限

## 10. 验证对stuUsr2用户权限收回操作是否成功。重新以stuUsrLogin2登录：



向Student表的插入一条记录，不能插入记录，表明权限收回正确。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')
```

```
SQLQuery2.sql - Y...tuUsrLogin2 (53)* × SQLQuery1.sql - Y...tuUsrLogin2 (52)
[...] INSERT INTO Student Values('070280','李四','计算机科学与技术','男','1990-5-1',78,'三好生')
150 %
消息
消息 229, 级别 14, 状态 5, 第 1 行
拒绝了对对象 'Student' (数据库 'StudentCourse', 架构 'dbo') 的 INSERT 权限。
```

成功收回stuUsr2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)