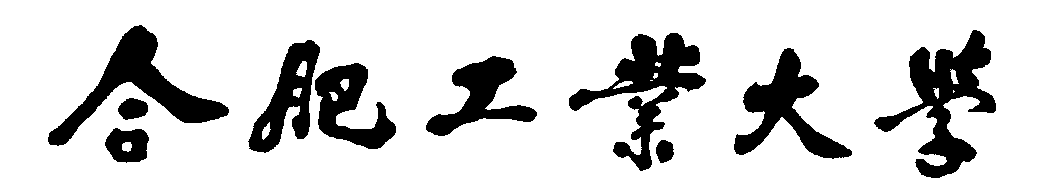
****

计算机与信息学院

实验报告

专 业 班 级： 计算机科学与技术18-3班

学生姓名及学号： 余梓俊 2018211991

课程教学班号： 0521082B--001

任 课 教 师： 沈明玉

实验指导教师： 胡敏

实 验 地 点： D502

|  |
| --- |
| 2020 - 2021 学年第一学期 |

实验1 数据库的创建和删除

1. **实验目的**

1. 了解SQL Server 2008或以上版本的数据库的逻辑结构和物理结构。

2. 掌握使用SQL 语句创建和删除数据库。

1. **实验要求**

1. 熟练利用查询分析器进行交互式SQL语句的使用方法。

2. 理解SQL Server中的共享架构dbo。

1. 注意操作结果的截图与保存，供撰写实验报告使用。
2. **实验内容**

1. 使用SQL语句创建数据库EDUC。

数据库的参数要求：

数据库名称：EDUC

数据库存储路径：将数据库文件存储在D:\jxgl\目录中。

其他参数要求：

数据文件的初始大小：10M 最大：60M 按5%的比例自动增长。

日志文件的初始大小：4M 最大：10M 增量为1M。

2. 将新创建的数据库设置为当前数据库。

**四、实验步骤**

1．创建数据库EDUC的SQL语句：

Create database 数据库名

[on

(name=数据文件的逻辑文件名,

filename=数据文件的物理文件名(含有具体存储路径)

[,Size=文件初始大小]

[,maxsize=文件可以增长到的最大大小]

[,filegrowth=文件的增长增量])]

[log on

( name=日志文件的逻辑文件名(注意不能和数据文件的逻辑名相同),

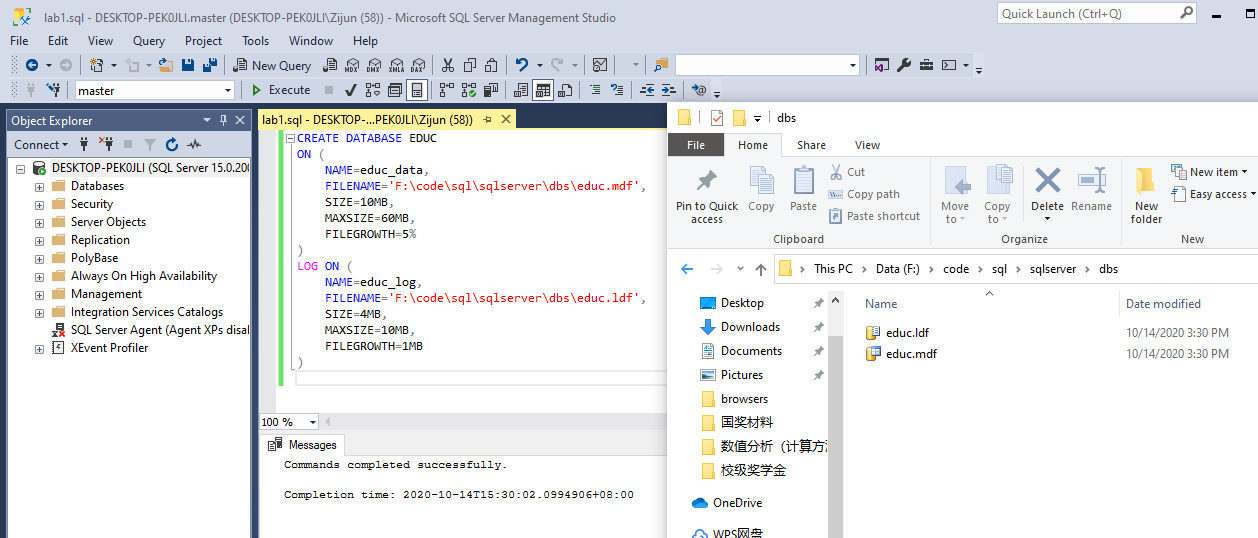
filename=日志文件的物理文件名(含有具体存储路径)

[,size=文件初始大小]

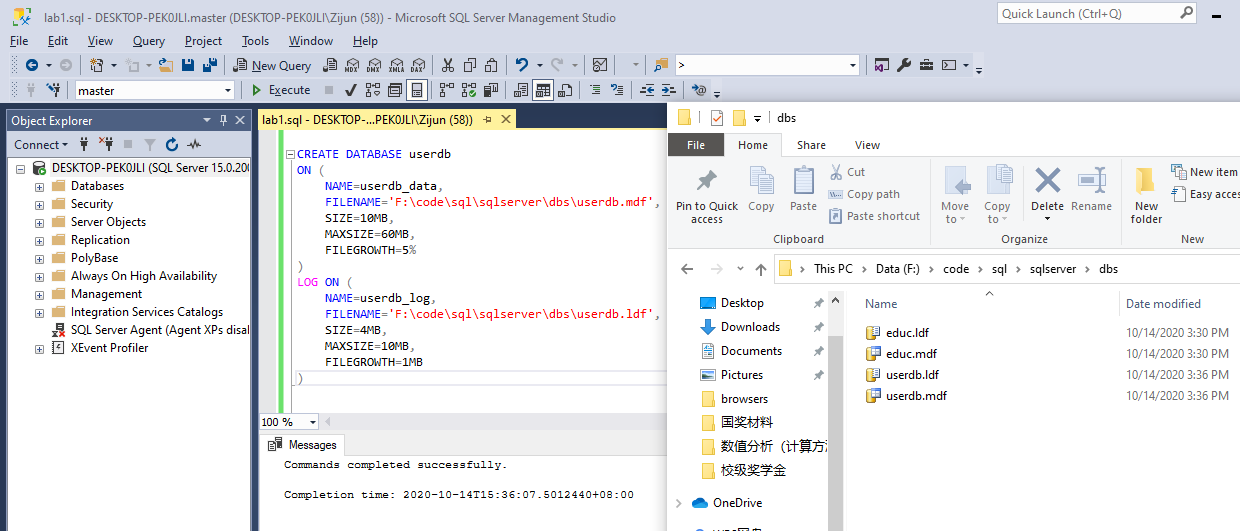
[,maxsize=文件可以增长到的最大大小]

[,filegrowth=文件的增长增量])]

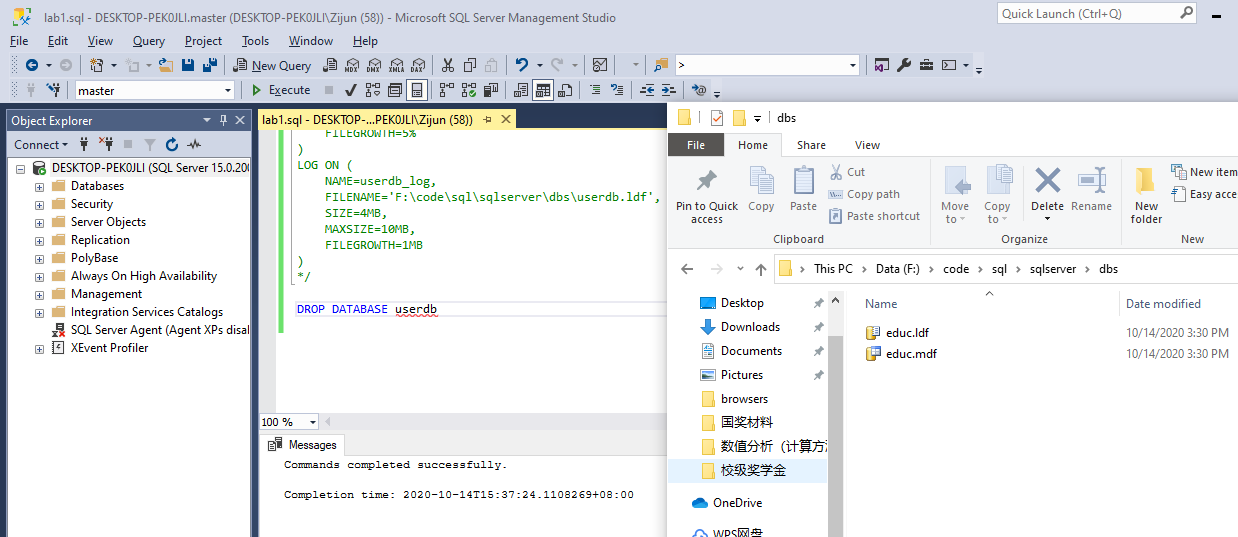
1. 根据步骤1的sql语句，写出创建数据库EDUC的sql语句，点击‘执行’按钮建立数据库EDUC。



1. 按照上述相同的要求创建数据库userdb。



1. 用SQL语句删除步骤3建立的数据库userdb。



**五、实验总结**

通过此次实验，我充分理解并掌握了利用SQL创建删除和使用数据库的步骤以及操作。

**实验二 使用SQL语句创建和删除基本表**

**一. 实验目的**

1．了解SQL Server的基本数据类型。

2．学会使用Create table语句创建基本表。

**二. 实验要求**

* 1. 在查询分析器中使用SQL语句完成基本表的创建、修改、删除。
  2. 实现基本表定义中有关完整性约束的要求。
  3. 注意操作结果的截图与保存，供撰写实验报告使用。

**三. 实验内容**

在数据库EDUC中，创建如下四个表：

Class、student、course、sc

表的具体要求如下：

1. class表的结构要求

班级号： clsNO, 字符型（定长），长度为6，主码

班级名称：clsName,字符型（变长），长度16，非空

辅导员： Director,字符型（变长），长度10

专业： Specialty,字符型（变长），长度30

1. student表的结构要求

学号：sno,字符型（定长），长度为8，主码

姓名：sname,字符型（变长），长度为10，非空

性别：ssex,字符型（定长），长度为2，性别只能为’男或女’

班级号：clsNO,字符型（定长），长度为6，外键

住址：saddr,字符型（变长），长度为20

年龄：sage，数值型（整数），长度为3，年龄要求10～30之间

身高：height，数值型（长度为4，含两位小数）

1. course表的结构要求

课程号：cno，字符型（定长），长度为4，主键

课程名：cname，字符型（变长），长度为16，非空

先修课程号：cpno，字符型（定长），长度为4，外键

学分：Credit，数值型（长度为2，含1位小数）

1. sc表的结构要求

学号：sno,字符型（定长），长度为8

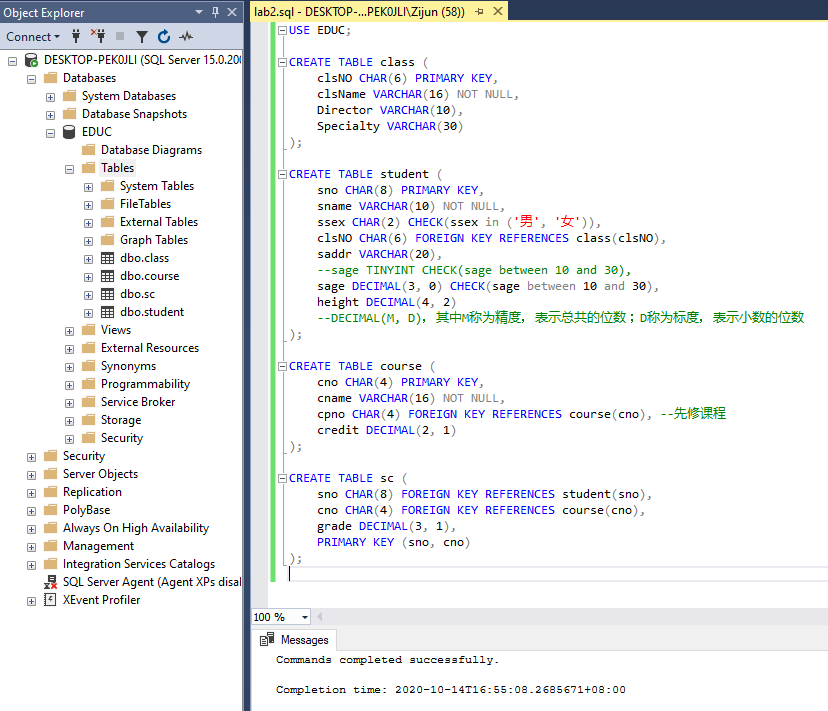
课程号：cno，字符型（定长），长度为4

成绩：grade，数值型（长度为3，含一位小数）

注意：本表主码与外码的定义。

1. **实验步骤**
2. 创建基本表

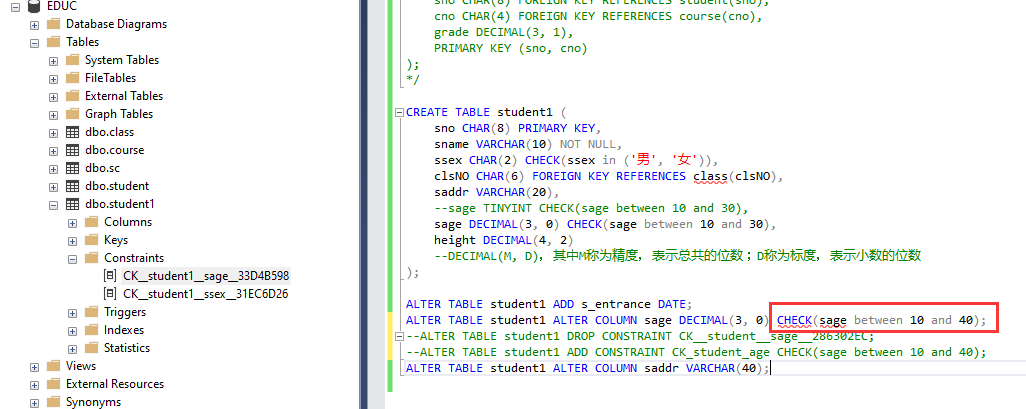
打开查询分析器，在其编辑窗口中输入基本表的创建语句，点击“执行”按钮，分别完成四个基本表的创建。

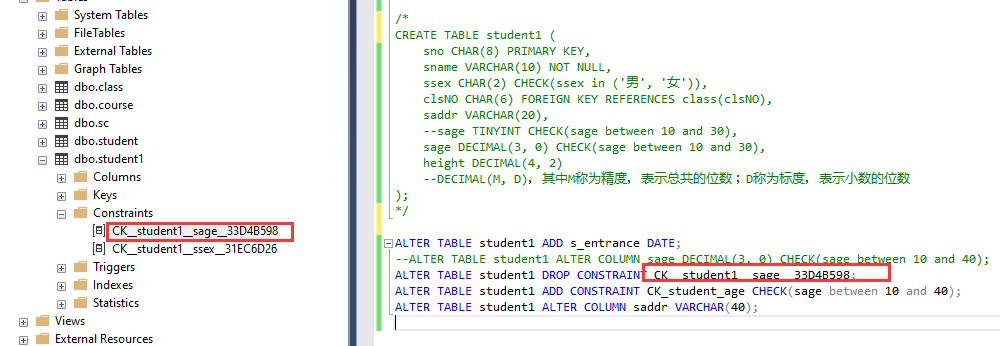


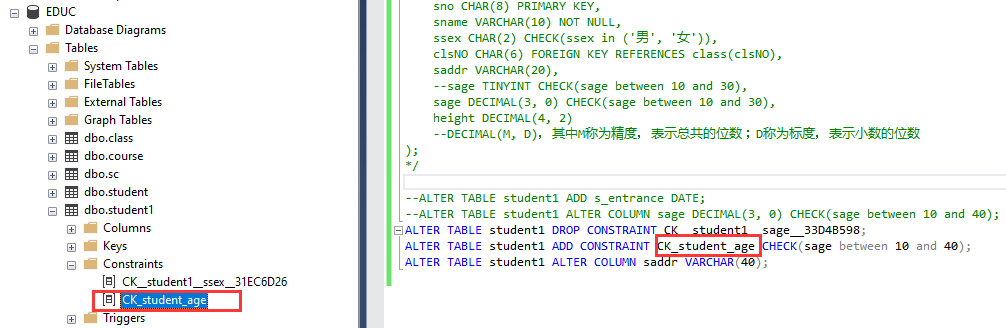
1. 修改表结构

按照student的要求创建表student1，修改基本表student1：

1. 增加“入学时间”列s\_entrance，其数据类型为日期型。
2. 将年龄属性值的限制由10~30改为10~40之间。
3. 将住址（saddr）的长度改为40。



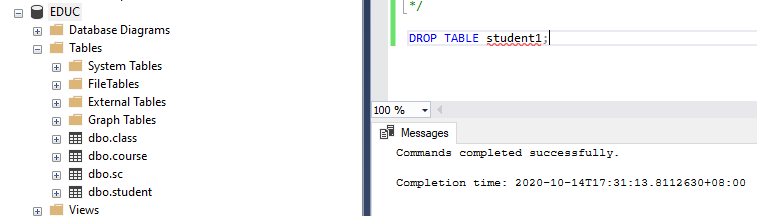




**修改限制并不能直接修改，而由于在刚开始创建的时候没有用SQL语句给属性限制命名，因此通过在管理器中查找发现了SQL Server为属性限制自动创建的名字，然后通过删除该限制并重新添加新的限制来实现目的。**

1. 删除基本表

删除基本表student1。



1. **实验总结**

通过此次实验，学会了数据表的基本操作，如创建和删除，修改限制条件等等，并了解了SQL Server的基本数据类型。在修改限制这一任务时遇到的问题也使我进一步熟悉了对属性限制的操纵。

**实验三 数据更新操作**

**一．实验目的**

1. 掌握INSERT/UPDATE/DELETE语句的使用。
2. 体会数据更新操作对完整性约束的影响。

**二．实验准备**

1. 准备基本表中的数据。

2. 了解INSERT、UPDATE及DELETE语句。

**三. 实验内容**

1. 数据插入操作

（1）向表（Class）中插入数据（表中数据仅供参考，请同学们自己准备数据！）

|  |  |  |  |
| --- | --- | --- | --- |
| **ClsNO** | **ClsName** | **Director** | **Specialty** |
| CS01 | 计算机一班 | 张宁 | 计算机应用 |
| CS02 | 计算机二班 | 王宁 | 计算机应用 |
| MT04 | 数学四班 | 陈晨 | 数学 |
| PH08 | 物理八班 | 葛格 | 物理 |
| GL01 | 地理一班 | 张四 | 应用地理 |

插入数据之后使用命令：Select \* from Class; 检查插入数据的正确性

(2) 向表（Student）中插入数据（数据自己准备，学生人数不要少于20人！）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sno** | **Sname** | **Ssex** | **ClsNO** | **Saddr** | **Sage** | **Height** |
| 20170101 | 王军 | 男 | CS01 | 下关40# | 20 | 1.76 |
| 20170102 | 李杰 | 男 | CS01 | 江边路96# | 22 | 1.72 |
| 20170306 | 王彤 | 女 | MT04 | 中央路94# | 19 | 1.65 |
| 20170107 | 吴杪 | 女 | PH08 | 莲化小区74# | 18 | 1.60 |

插入数据之后使用命令：Select \* from Student; 检查插入数据的正确性

（3）向表（Course ）中插入数据（数据自己准备，课程数不要少于10门！）

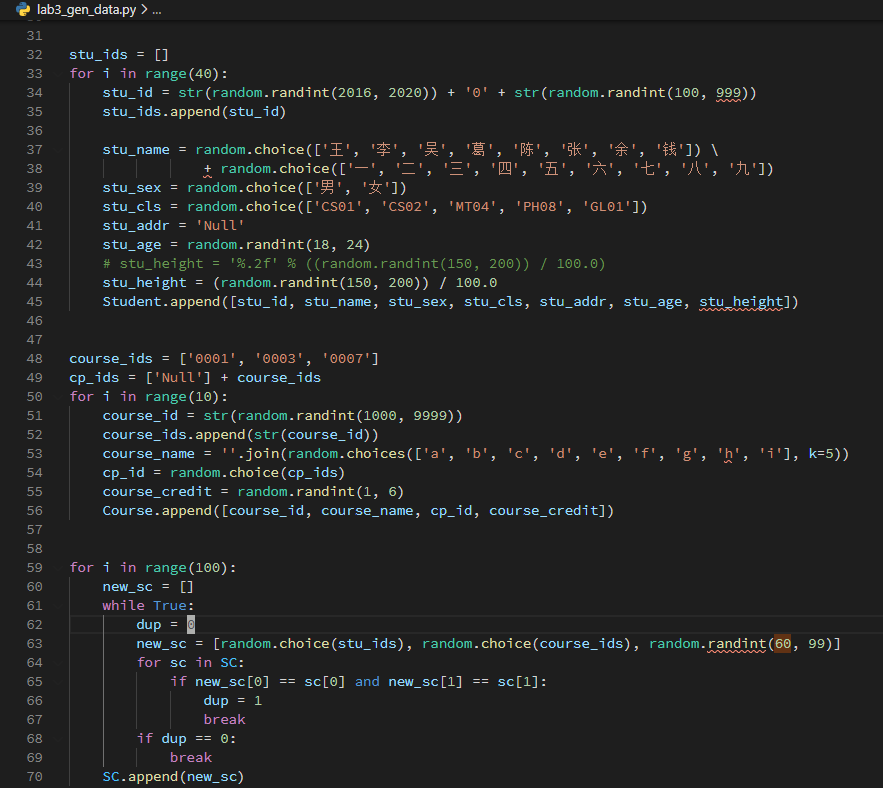
|  |  |  |  |
| --- | --- | --- | --- |
| **Cno** | **Cname** | **Cpno** | **Credit** |
| 0001 | 高等数学 | Null | 6 |
| 0003 | 计算机基础 | 0001 | 3 |
| 0007 | 物理 | 0001 | 4 |

插入数据之后使用命令：Select \* from Course; 检查插入数据的正确性

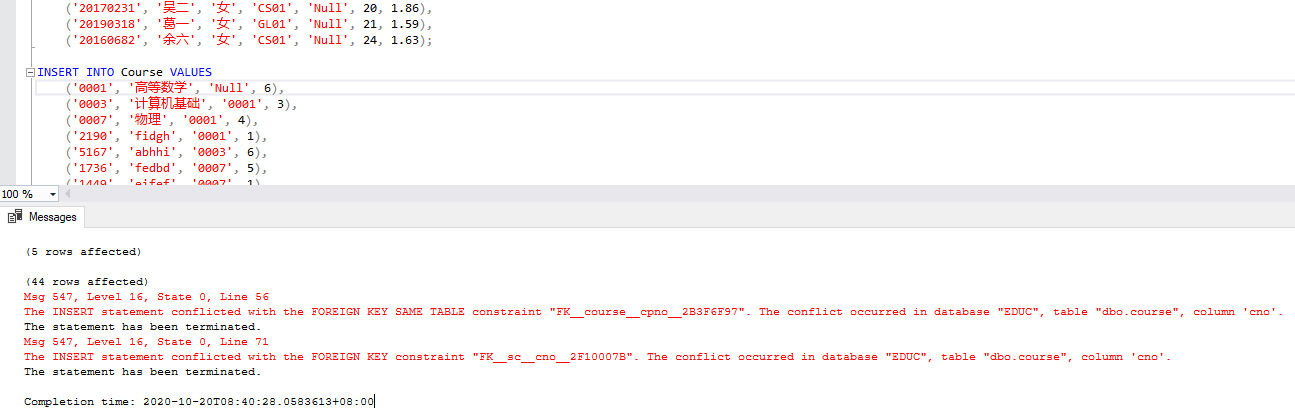
（4）向表（SC ）中插入数据（平均每位学生的选课门数不要少于2门！）

|  |  |  |
| --- | --- | --- |
| **SNO** | **CNO** | **Grade** |
| 20170101 | 0001 | 90 |
| 20170101 | 0007 | 86 |
| 20170102 | 0001 | 87 |
| 20170102 | 0003 | 76 |
| 20170306 | 0001 | 87 |
| 20170306 | 0003 | 93 |
| 20170107 | 0007 | 85 |
| 20170306 | 0007 | 90 |

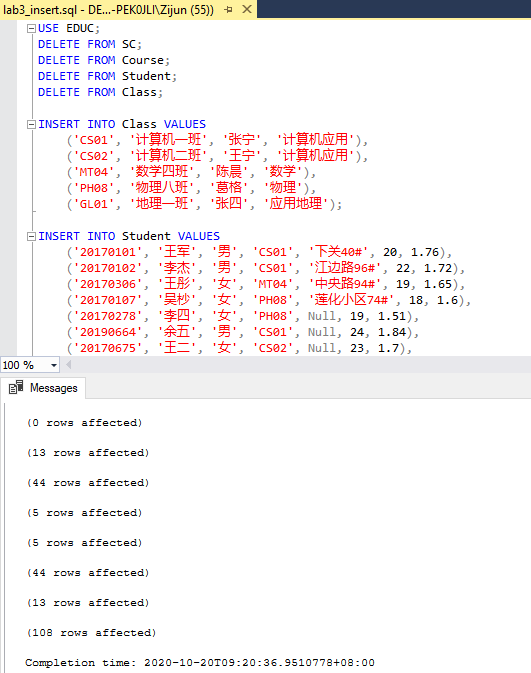
**利用Python生成一些随机数据，生成SQL语句并写入一个SQL文本文件：**



**中间遇到过一些小问题，例如空置应该是NULL，而不是 ‘NULL’ ：**

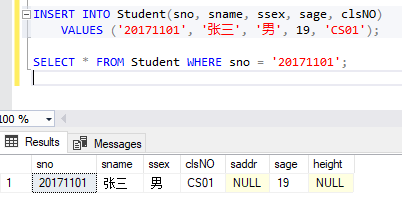


**修改后：**

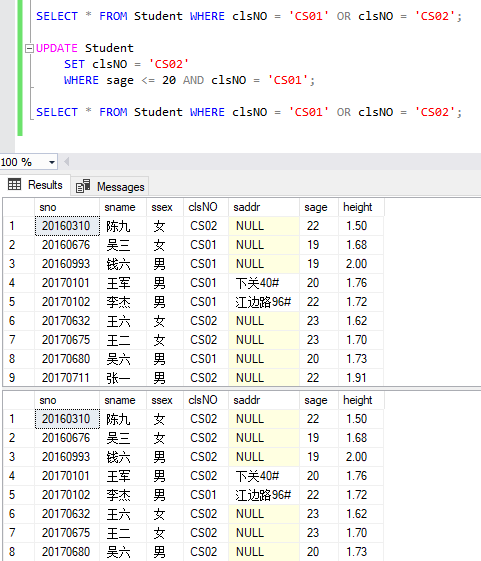


2.数据修改与删除操作

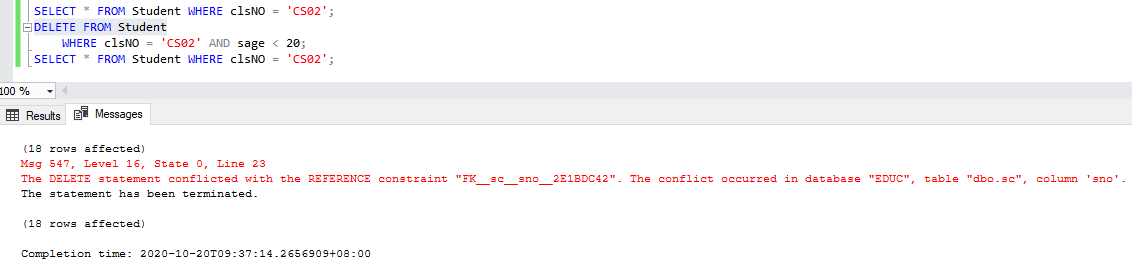
首先在student表中插入一条新记录：学号：20171101、姓名：张三、性别：男、年龄：19、班级编号：‘CS01’。



1. 对于student表，将所有班级号为‘CS01’的，并且年龄小于20岁的学生的班级号改为‘CS02’。



1. 对于student表，删掉所有年龄小于20岁，并且专业号为‘CS02’的学生的记录。



**删除时报冲突，因为SC表中引用了Student的外键，所以有选课的同学不能直接被删除。**

**五、实验总结**

通过此次实验，我更加熟悉了SQL插入数据的方法和一些要注意的点，比如上文提到的NULL值等等。对数据的更新操作也有了进一步的理解，并熟练掌握了INSERT/UPDATE/DELETE语句的使用。并且体会到了数据更新操作对完整性约束的影响。

# 实验四 数据库的分离和附加

**一. 实验目的**

掌握数据库的分离和附加的具体操作方法。

**二. 实验要求**

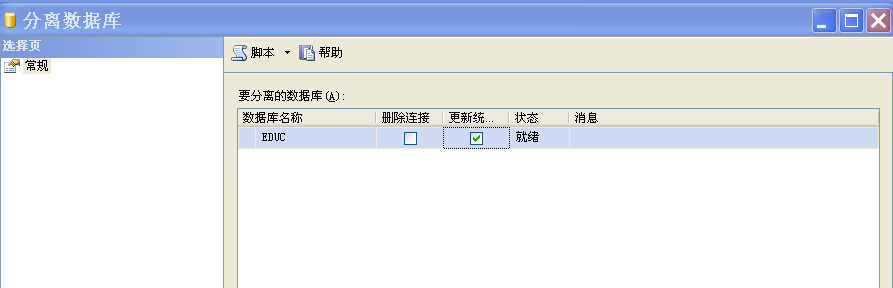
1. 将创建的数据库EDUC进行分离并拷贝到U盘。
2. 注意操作结果的截图与保存，供撰写实验报告使用。

**三. 实验内容**

1）数据库的分离

在对象资源管理器中，选择数据库节点EDUC，右击-任务-分离，进入如下图所示的界面，选择“更新统计消息”下的复选框，单击确定，完成数据库的分离，此时可以在存储数据库的文件夹中将分离后的数据库的数据文件和日志文件拷贝到U盘里。

（若存在问题，可尝试勾选删除连接！）



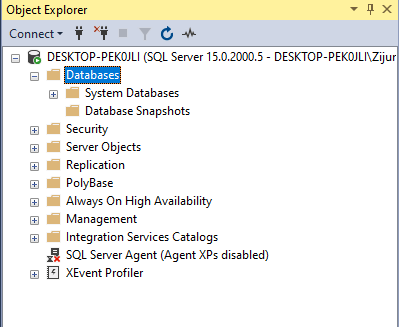
2）数据库的附加

在对象资源管理器中，选择数据库节点，右击-附加-添加，选择数据文件，单击确定，完成数据库的附加。

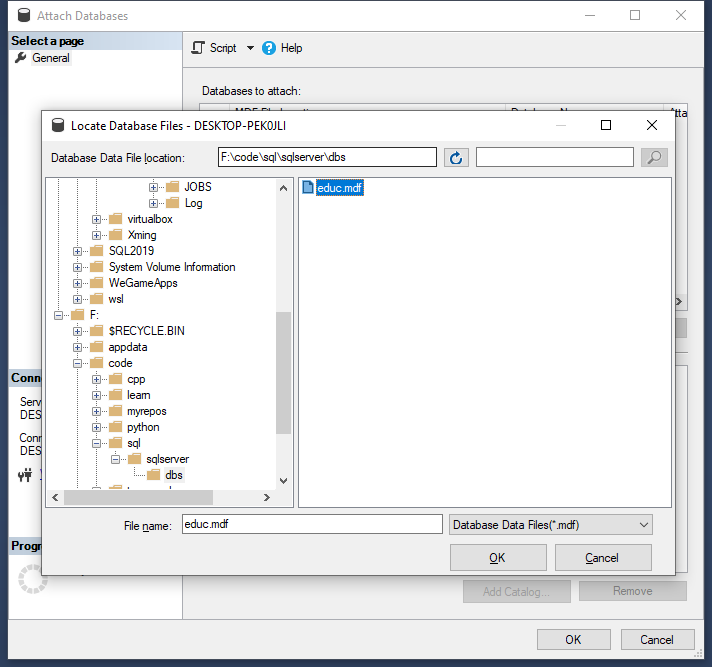
**四、实验步骤：**

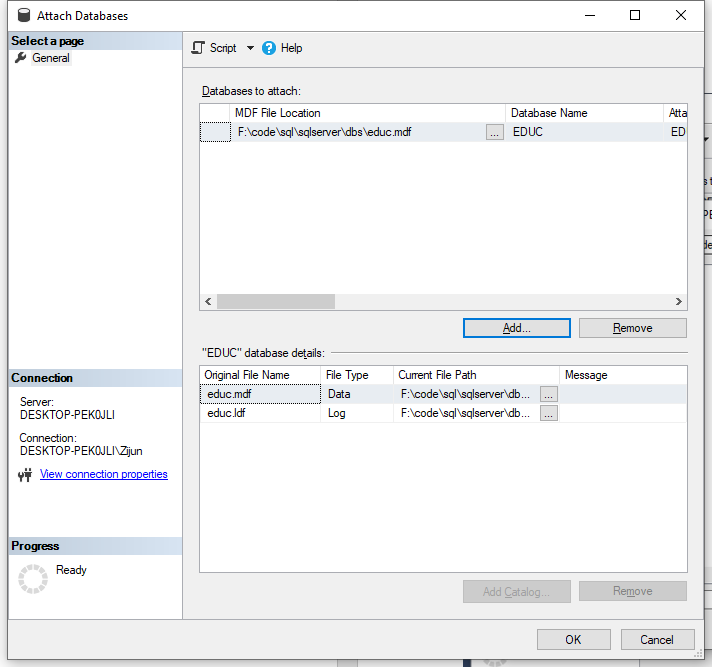
按照实验内容中的提示进行分离。

分离后Databases中已经没有EDUC：

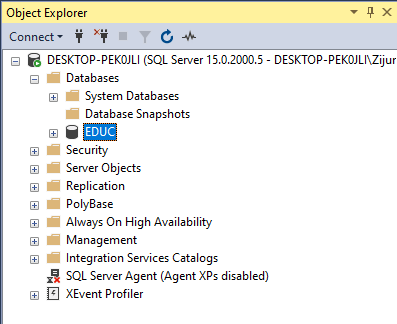


然后再进行附加：





附加成功后再次看到EDUC基本表：



# 实验五 创建和删除索引

**一. 实验目的**

1.了解索引的类型和作用。

2.学会用SQL语句对表创建和删除索引

**二. 实验要求**

1. 了解索引类型并比较各类索引的不同之处。

2. 完成索引的创建和删除。

3. 注意操作结果的截图与保存，供撰写实验报告使用。

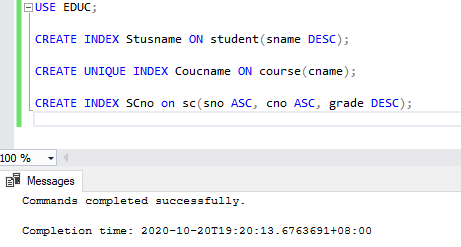
**三. 实验内容**

1. 用SQL语句分别建立以下索引

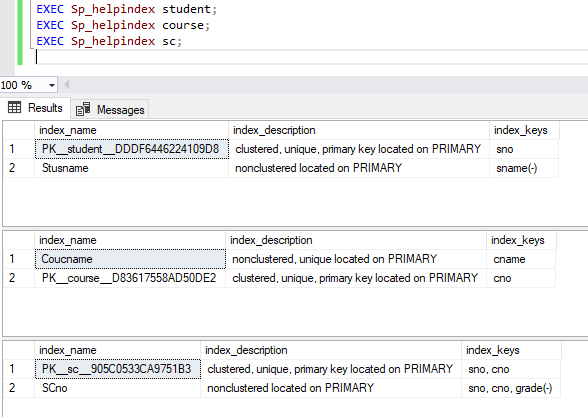
(1) 在student表的Sname列上建立普通降序索引Stusname。

(2) 在course表的Cname列上建立唯一索引Coucname。

(3) 在sc表的Sno(升序), Cno(升序)和grade (降序)三列上建立一个普通索引SCno。

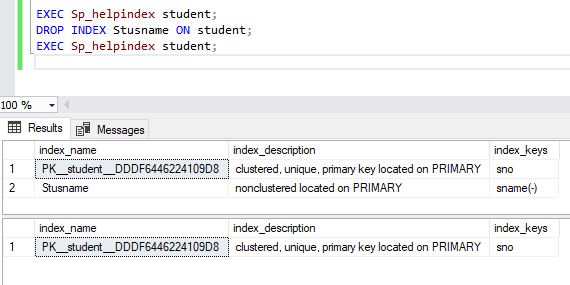


2. 检查创建索引的完成情况。



3. 用SQL语句删除索引

删除Student表的Stusname索引。



1. **实验总结**

通过此次实验，我学会了如何利用SQL语句对基本表创建和删除索引，对于不同的索引类型及其作用也有了进一步的认识。

**实验六 数据查询**

**一．实验目的**

1. 观察查询结果, 体会SELECT语句实际应用；
2. 要求学生能够在查询分析器中使用SELECT语句进行简单查询。
3. 熟练掌握简单表的数据查询、数据排序和数据连接查询的操作方法。

**二．实验要求**

1. 完成所要求的全部查询。

2. 注意操作结果的截图与保存，供撰写实验报告使用。

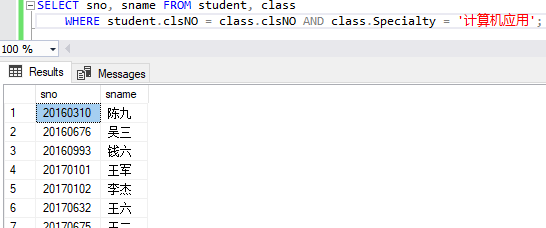
**三．实验内容**

所有的查询全部用Transact-SQL语句实现。此部分查询包括投影、选择条件表达、数据排序、使用临时表等。

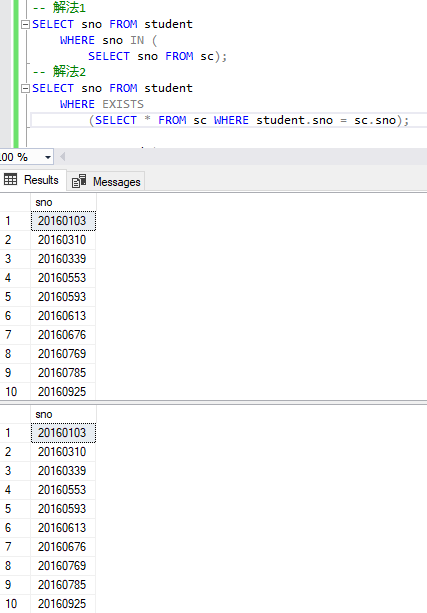
对EDUC数据库实现以下查询：

（可根据数据表中的数据，对要求中的条件作出调整！）

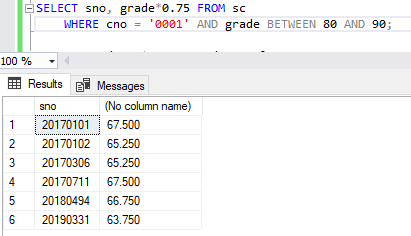
1. 查询“计算机应用”专业的学生学号和姓名；



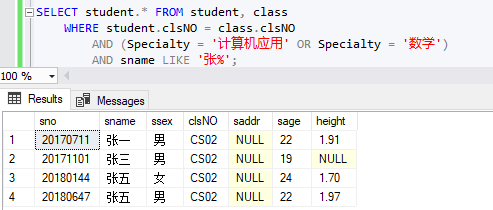
1. 查询选修了课程的学生学号；



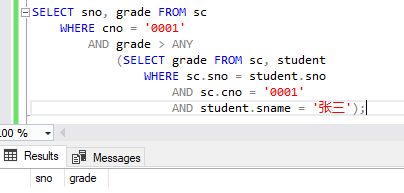
1. 查询选修课程0001 且成绩在80－90 之间的学生学号和成绩，并将成绩乘以系数0.75 输出；



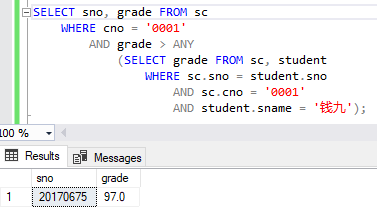
1. 查询“计算机应用”和“数学”专业的姓“张”的学生的信息。



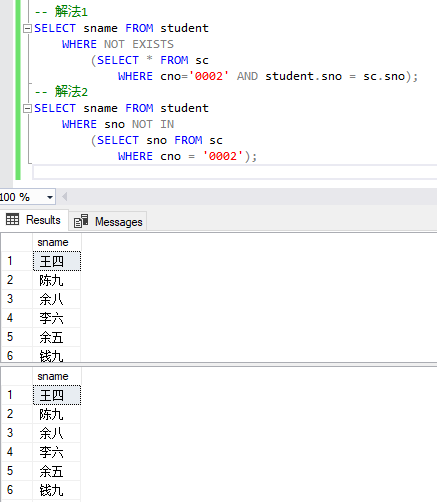
1. 查询“0001”课程的成绩高于张三的学生学号和成绩；



**发现没有张三选修了0001，找到一个成绩为94分的选修了0001的学生“钱九“**



1. 查询没有选修“0002”课程的学生姓名。



**五．实验总结**

通过此次实验，我切实掌握了SELECT语句的实际应用，熟悉了简单查询、连接查询、嵌套查询等等，同时也明白了IN、EXISTS等关键词的作用以及应用场景。同时我进一步理解了相关和不相关子查询，例如题2和题6，我均给出了EXISTS和IN两种解决方法，而其中使用EXISTS时便是相关子查询。通过此次实验，我对数据表的结构也有了更深的认识。

实验七 使用聚集函数的SELECT语句

**一、实验目的**

1. 熟练掌握数据查询中的分组、统计、计算和组合的操作方法。

2. 进一步掌握SQL Server 查询分析器的使用，加深对SQL语言的嵌套查询语句的理解。

**二、实验要求**

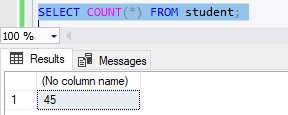
1. 在实验之前做好准备。

2. 注意操作结果的截图与保存，供撰写实验报告使用。

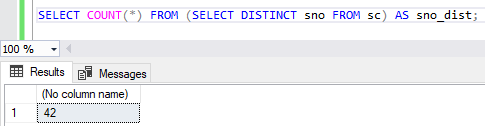
**三、实验内容**

在数据库EDUC中用SQL语句实现如下查询：

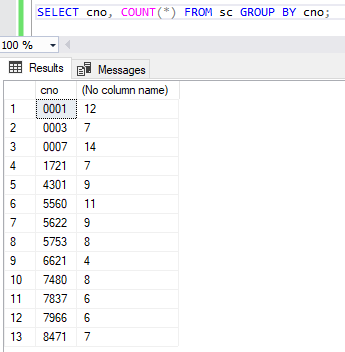
1) 查询学生的总人数。



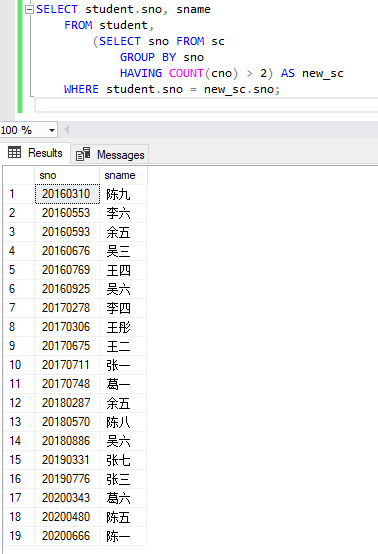
2) 查询选修了课程的学生人数。



3) 查询课程的课程号和选修该课程的人数。



4) 查询选修课程超过2 门课的学生学号和姓名。



**五. 实验总结**

通过此次实验，我掌握了数据查询中的分组、统计的操作方法并进一步加深了对SQL语言的嵌套查询和GROUP BY语句的理解。同时还注意到了对于SQL语句中用到的任何一个虚拟表，都需要用AS关键字为其赋予一个名字，即使可能不需要引用到这个名字。

实验八 视图的定义与使用

1. **实验目的**
2. 熟悉和掌握对数据表中视图的查询操作和SQL语句的使用；
3. 熟悉和掌握对数据表中视图的更新操作和SQL语句的使用，并注意视图更新与基本表更新的区别与联系；
4. 学习灵活熟练的进行视图的操作，认识视图的作用。

**二．实验要求**

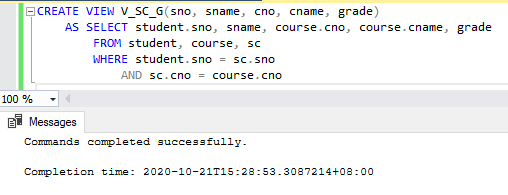
1. 在实验开始之前做好准备工作。
2. 思考视图和基本表的区别。
3. 注意操作结果的截图与保存，供撰写实验报告使用。

**三．实验步骤**

1.定义视图

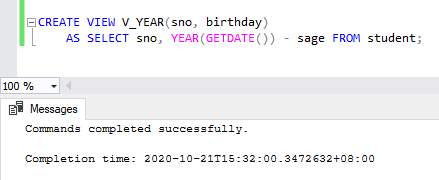
在EDUC数据库中，已Student、Course 和SC表为基础完成一下视图定义:

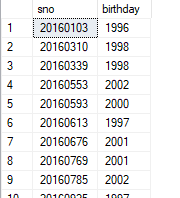
1. 定义视图V\_SC\_G：该视图包含Student、 Course 和SC表中学生的学号、姓名、课程号、课程名和成绩；



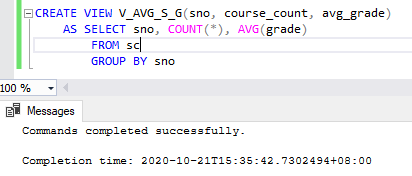


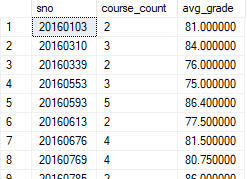
1. 定义一个反映学生出生年份的视图V\_YEAR，该视图要求使用系统函数来获取当前日期及转换；



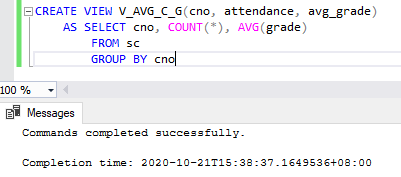


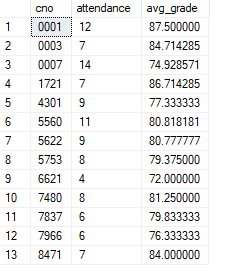
1. 定义视图V\_AVG\_S\_G：该视图将反映学生选修课程的门数及平均成绩；





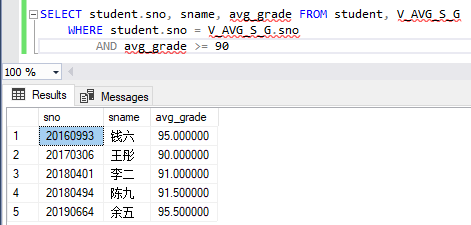
1. 定义视图V\_AVG\_C\_G：该视图将统计各门课程的选修人数及平均成绩。



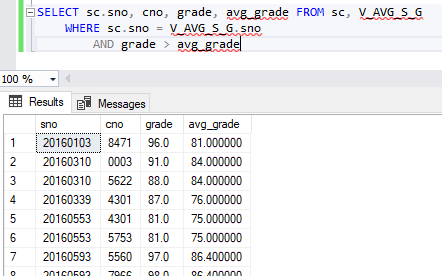


2.使用视图

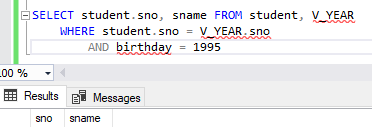
1. 查询以上所建的视图结果。
2. 查询平均成绩为90分及以上的学生学号、姓名和成绩；



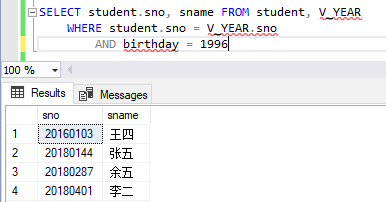
1. 查询科目成绩大于平均成绩的学生学号、课程号、成绩和平均成绩；



1. 查询1995年出生的学生学号和姓名。



**发现没有95年的学生，尝试查询96年的学生：**



**五、实验总结**

通过此次实验，我学会了视图的创建和使用，思考了视图更新和基本表更新的区别，进一步理解了视图和虚拟表的区别，也体会到了使用视图带来的方便。

# 实验九 触发器的创建与使用

1. **实验目的**

本实验的目的是使学生进一步掌握SQL Server触发器的创建及使用方法，加深SQL触发器的理解。通过对数据的更新操作体会其触发器的作用。

**二．实验要求**

1. 掌握T-SQL创建触发器语句Create trigger的使用。

2. 完成所规定的触发器定义要求。

3. 通过数据查询检查触发器的使用效果。

4. 注意操作结果的截图与保存，供撰写实验报告使用。

**三．实验内容**

1. 在班级表class中增加班级人数（c\_total）字段。

2. 为学生表(student)创建INSERT触发器t\_inst\_stu：新增一名学生时，若其班级编号非空，则将班级表（class）中相应班级的人数（c\_total）自动加1。

3. 为学生表(student)创建DELETE触发器t\_dele\_stu：删除一名学生时，若其班级编号非空，则将班级表（class）中相应班级的人数（c\_total）自动减1。

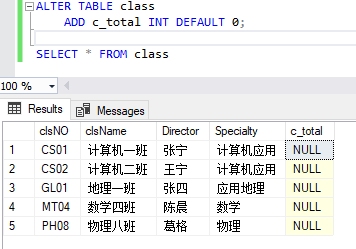
4. 为学生表(student)创建UPDATE触发器t\_update\_stu: 当某学生所在班号发生变化时（即调到另一班级后），将其原先所在班级的人数（c\_total）减1，将新调入的班级班级的人数（c\_total）加1。

5. 体会触发器的作用。

**四、实验步骤**

在查询分析器中完成实验内容。

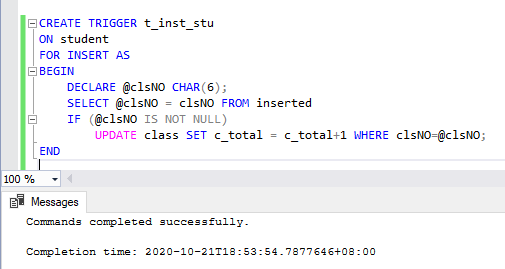
1. 使用alter table语句在班级表class中增加班级人数（c\_total）字段，默认值（DEFAULT）是0；



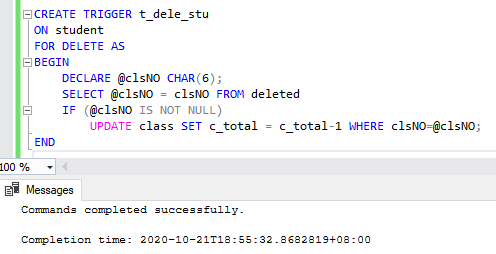
**此处虽然设置了默认值为0，但查询后发现值均为NULL，并且在后续实验过程中，发现触发器一直没有生效，最后发现用一条UPDATE语句c\_total将全部置为0后，触发器即可正常工作。**

2. 为学生表(student)分别创建：

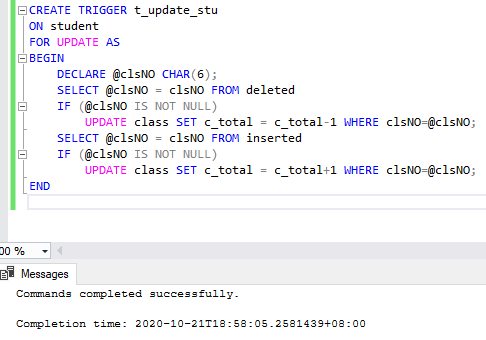
* INSERT触发器t\_inst\_stu



* DELETE触发器t\_dele\_stu



* UPDATE触发器t\_update\_stu



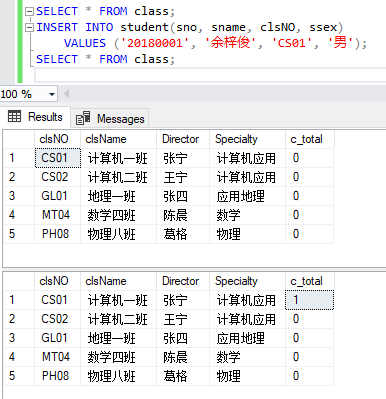
3. 体会触发器的作用：

1) 查看班级表（class）

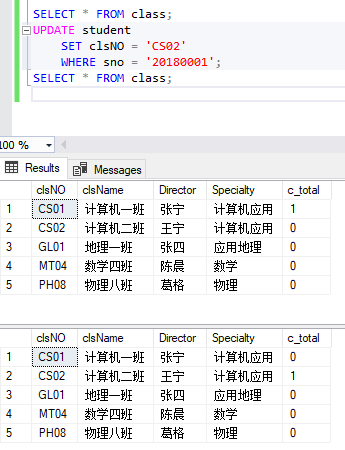
2）对学生表(student)分别插入（INSERT）、删除（DELETE）和修改（UPDATE）元组

3）再次查看班级表（class），检查其数据的变化情况，体会触发器的作用。

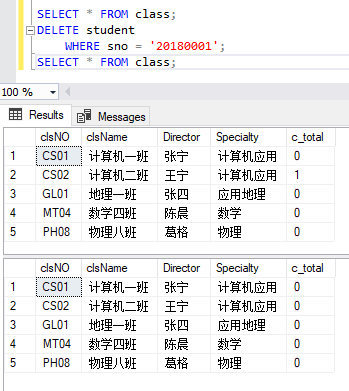
**插入：**



**更新：**



**删除：**



**五、实验总结**

通过此次实验，我学会了触发器的创建和使用，体会到了触发器对于完成自动化任务的方便和实用。同时上文提到的关于NULL和0对触发器效果的影响，也进一步加深了我对SQL Server不同数据类型的理解。

# 实验十 存储过程的创建与使用

**一、实验目的**

本实验的目的是使学生进一步掌握SQL Server存储过程的创建及使用方法，加深对SQL存储过程的理解。通过对存储过程的调用体会其作用。

**二、实验要求**

1. 掌握T-SQL创建触发器语句Create procedure的使用。

2. 完成所规定的存储过程定义要求。

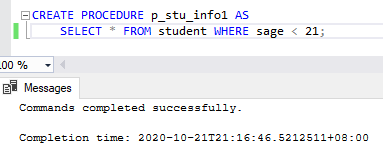
3. 调用（EXEC）所创建的过程，检查过程的执行结果。

4. 注意操作结果的截图与保存，供撰写实验报告使用。

**三、实验内容**

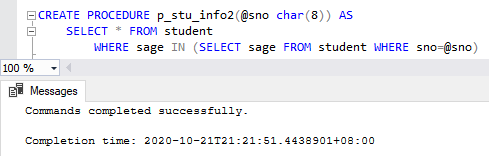
1、创建一个不带参数的存储过程p\_stu\_info1，实现对满足要求的学生基本信息的查询。

要求：所有年龄<21岁的男同学



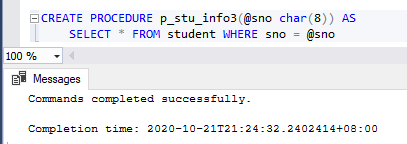
1. 创建一个带有参数的存储过程p\_stu\_info2，实现对满足要求的学生基本信息的查询。

要求：输入参数为学号，与指定学号的学生同龄的所有同学。



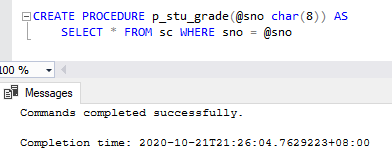
1. 创建一个存储过程p\_stu\_info3,根据输入的学号，查询某学生的基本信息。

要求：输入参数为学号。

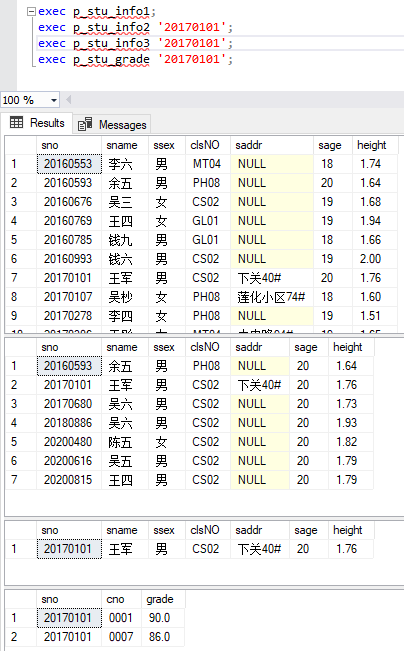


1. 创建一个存储过程p\_stu\_grade,根据输入的学号，返回其选课及其成绩。

要求：输入参数为学号。



5.使用SQL语句分别执行p\_stu\_info1、p\_stu\_info2、p\_stu\_info3和p\_stu\_grade，并查看显示结果。



**五、实验总结**

通过此次实验，我学会了T-SQL创建触发器的方法，完成了所规定的存储过程定义要求，学会了如何使用创建的触发器。

# 实验十一 用户管理操作

**一、实验目的**

本实验的目的是使学生进一步掌握SQL Server用户角色及权限的使用方法，加深SQL用户管理的理解。SQL Server的特定数据库都有自己的用户和角色（用户组），该数据库只能由它的用户或角色访问，其他用户无权访问该数据库数据，数据库系统可以通过创建和管理特定的数据库的用户和角色来保证数据库不被非法用户访问。

**二、实验要求**

1.实现对SQL Server的用户和角色管理

2.设置和管理数据操作权限

**三、实验内容**

1、创建登录帐号

2、创建用户

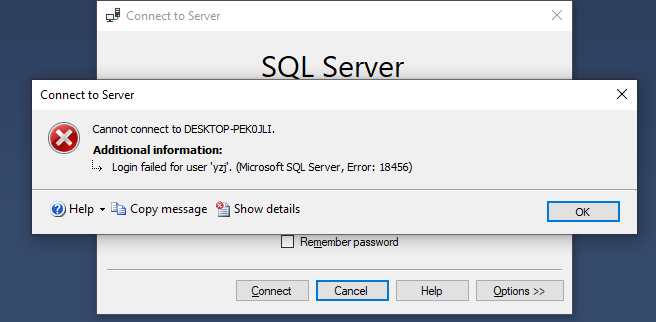
3、创建角色

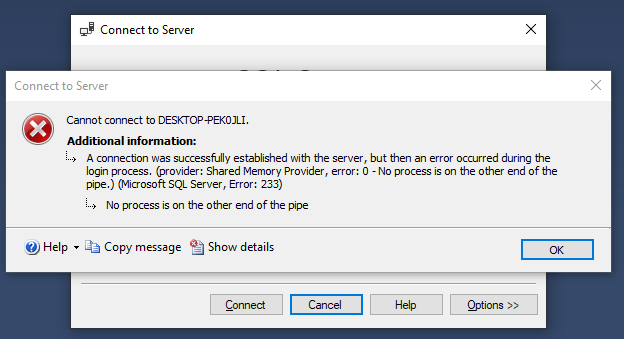
4、授权

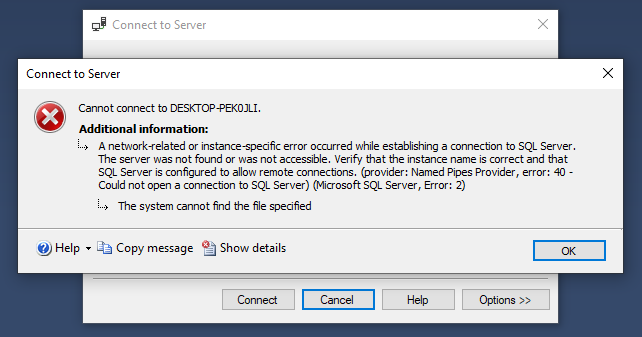
5、验证授权操作

**四、实验步骤**

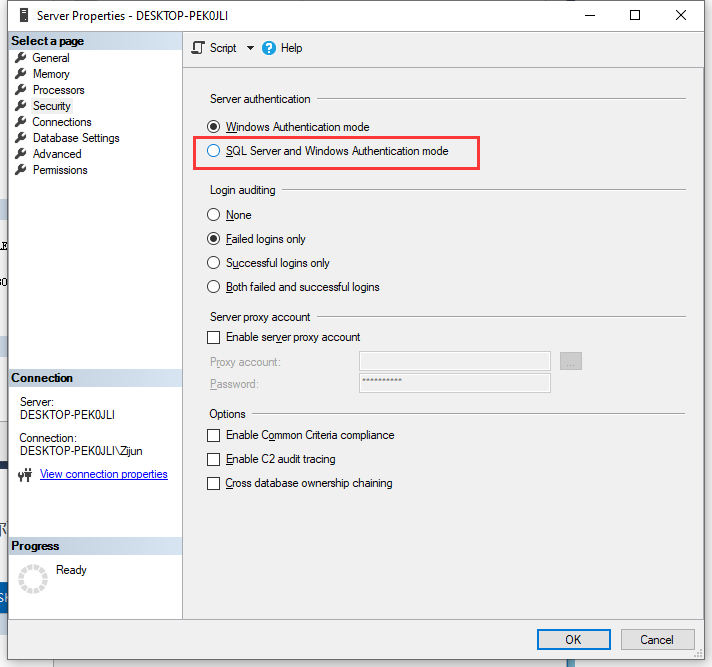
首先按照文档中的提示进行操作，使用向导进行用户的创建，在创建完登录用户后尝试登录时分别遇到了如下三种错误。

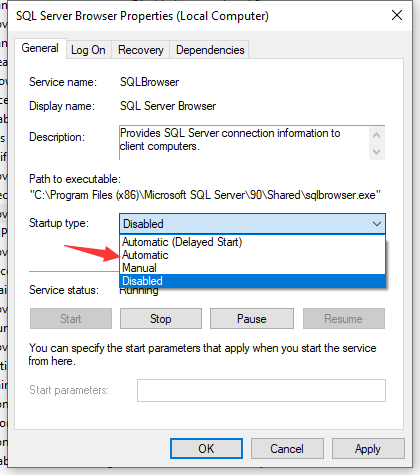




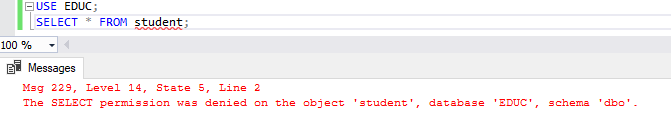


部分解决过程如下：

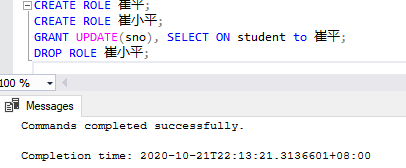


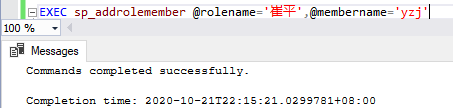


成功使用新创建的yzj用户登录数据库系统，然后尝试访问EDUC数据库，发现的确没有权限。

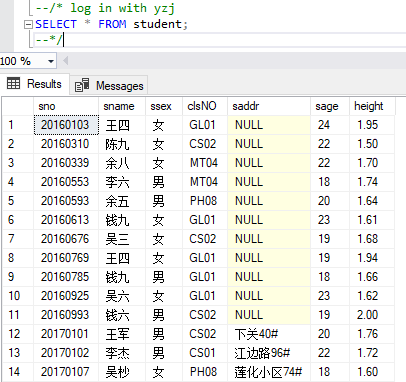


换回windows身份认证登录，创建角色“崔小平”，给予相应的权限，并将改角色赋给用户yzj。





再次用yzj登录，角色“崔平”的权限成功授予了用户yzj



**五、实验总结**

在此次实验的过程中，创建登录用户并进行登录这一任务遇到了许多不同的问题，查阅了网上的资料并尝试了众多解决方法后，成功地登录了自己创建的新用户。通过此次实验，我学会如何在SQL Server上创建新的用户，进一步掌握了SQL Server用户角色及权限的使用方法，加深了对SQL用户管理及其安全性的理解。