

合肥工业大学

计算机与信息学院

数据库系统概论实验报告

专 业 班 级

物联网 19-2 班

学生姓名及学号

2018217819 付炎平

课程教学班号

03

任 课 教 师

沈明玉

实验指导教师

沈明玉

实 验 地 点

计算机楼第四机房

2021~2022 学年第 二 学期

实验一 数据库的创建和删除

一、实验目的

1. 了解 SQL Server 2008 数据库管理系统的操作环境；
2. 学习 SQL Server Management Studio 管理平台的使用；
3. 理解 SQL Server 数据库服务器用户登录的特点与过程；
4. 掌握使用 SQL 语句创建和删除数据库。

二、实验要求

1. 学会 SQL Server Management Studio 中利用查询窗口进行交互式 SQL 语句的编写与执行方法。
2. 用 SQL 语句创建第一个属于你自己的数据库 EDUC。
3. 注意实验过程的截图与保存，供撰写实验报告使用。

三、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:\MYDB\ 文件夹中。

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. 创建另一个数据库 TestDB，参数与 EDUC 相同。
3. 用 SQL 语句删除数据库 TestDB。
4. 将 EDUC 数据库设置为当前数据库。

四、实验步骤

1. 创建数据库 EDUC



图 1 创建数据库 EDUC

2. 创建数据库 TestDB



图 2 创建数据库 TestDB

3. 删除数据库 TestDB



图 3 删除数据库 TestDB

4. 使用数据库 EDUC

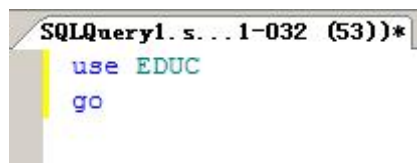


图 4 使用数据库 EDUC

五、实验总结

在本次实验过程中我学习到的内容有

- 1) 创建特定参数的数据库
- 2) 删除指定数据库
- 3) 在查询中使用数据库

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. SQL Server Management Studio 的查询窗口中，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. 修改基本表的结构。
4. 删除基本表。
5. 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

(一) 根据设计要求，用 Create table 语句定义基本表

设有简单教学管理的 6 个关系模式如下：

系(系编号、系名称、系主任、联系电话、地址)

专业(专业编号、专业名称、所属系)

教师(教师编号、姓名、所在系、电话、电子信箱)

学生(学号、姓名、性别、出生日期、所学专业)

课程(课程号、课程名、学分、先修课)

学生选课(学号、课程号、成绩)

要求：在数据库 EDUC 中，创建对应的 6 个基本表，基本表的名称和属性名称由同学自行确定。

基本表的数据结构及完整性要求：

1. 系 (系编号、系名称、系主任、联系电话、地址)

系编号：字符型（定长），长度为 4，主码

系名称：字符型（变长），长度 20，非空

系主任：字符型（定长），长度为 10

联系电话：字符型（变长），长度 20

地址：字符型（变长），长度 40

2. 专业 (专业编号、专业名称、所属系)

专业编号：字符型（定长），长度为 2，主码

专业名称：字符型（变长），长度 40，非空

所属系：字符型（定长），长度为 4，外码

3. 教师 (教师编号、姓名、所在系、电话、电子信箱)

教师编号：字符型（定长），长度为 10，主码

姓名：字符型（定长），长度 10，非空

所在系：字符型（定长），长度为 4，外码

电话：字符型（变长），长度 20

电子信箱：字符型（变长），长度 30

4. 学生 (学号、姓名、性别、出生日期、所学专业)

学号：字符型（定长），长度为 10，主码

姓名：字符型（变长），长度 20，非空，唯一

性别：字符型（定长），长度为 2，性别只能为‘男’或‘女’

出生日期：日期型（datetime）

所学专业：字符型（定长），长度为 2，外码

5. 课程 (课程号、课程名、学分、先修课)

课程号：字符型（定长），长度为 4，主码

课程名：字符型（变长），长度 30

学分：数值型（长度为 2，含 1 位小数）

先修课：字符型（定长），长度为 4，外码

6. 学生选课 (学号、课程号、成绩)

学号：字符型（定长），长度为 10，外码

课程号：字符型（定长），长度为 4，外码

成绩：数值型（长度为 4，含 1 位小数），成绩必须在：0~100 分之间。

本表的主码为：（学号，课程号）。

（二）用 Alter table 语句修改基本表的结构，以及完整性约束条件

操作要求：

1. 为学生表增加属性：班级：字符型（变长），长度 20。
2. 为课程表添加一条完整性约束条件：课程名不允许为空值。
3. 删除学生表中的班级属性。
4. 删除学生表中对学生姓名的唯一值限制。

（三）用 Drop table 语句删除基本表

1. 创建一个新的基本表
2. 删除该表

四、实验步骤

1. 创建基本表

1) 创建系表

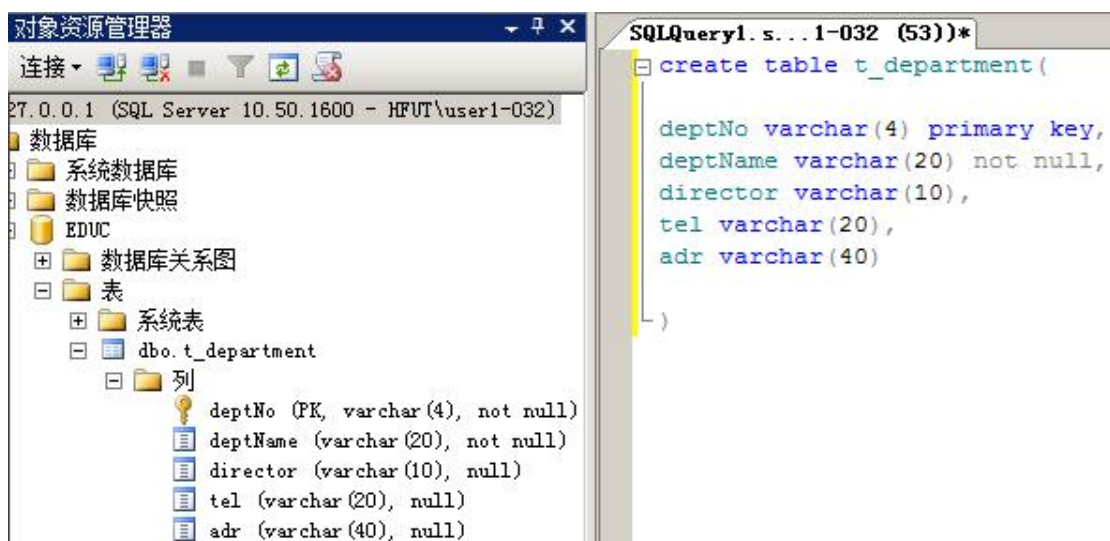


图 1.1 创建系表

2) 创建专业表

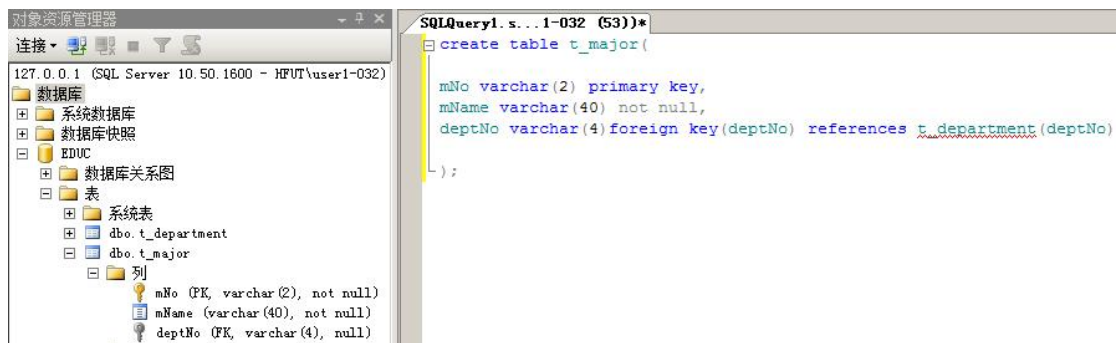


图 1.2 创建专业表

3) 创建教师表

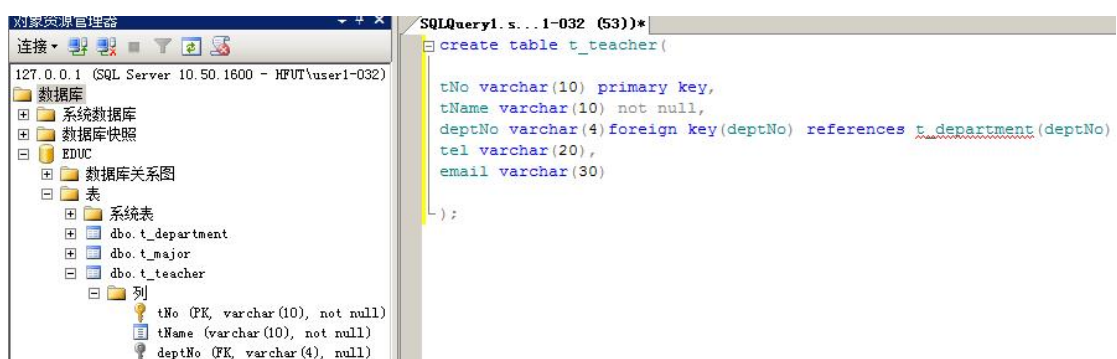


图 1.3 创建教师表

4) 创建学生表

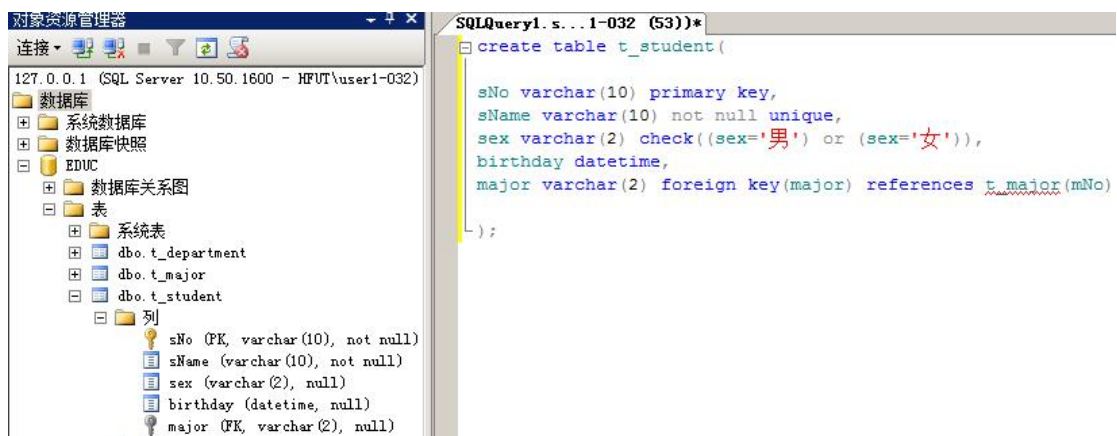


图 1.4 创建学生表

5) 创建课程表

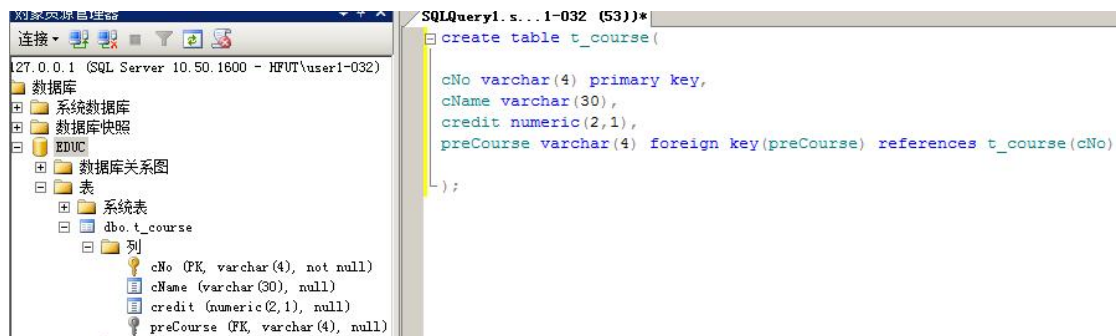


图 1.5 创建课程表

6) 创建选课表

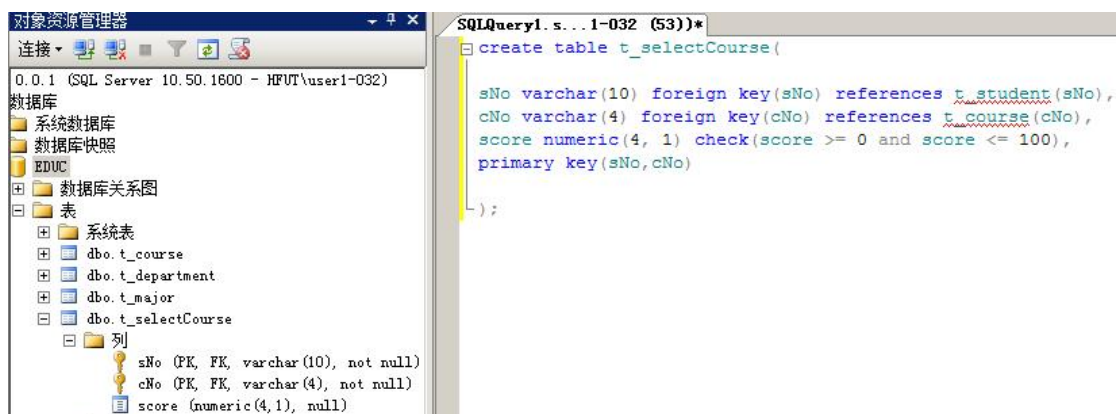


图 1.6 创建选课表

2.修改表结构

1) 为学生表增加属性：班级：字符型（变长），长度 20。

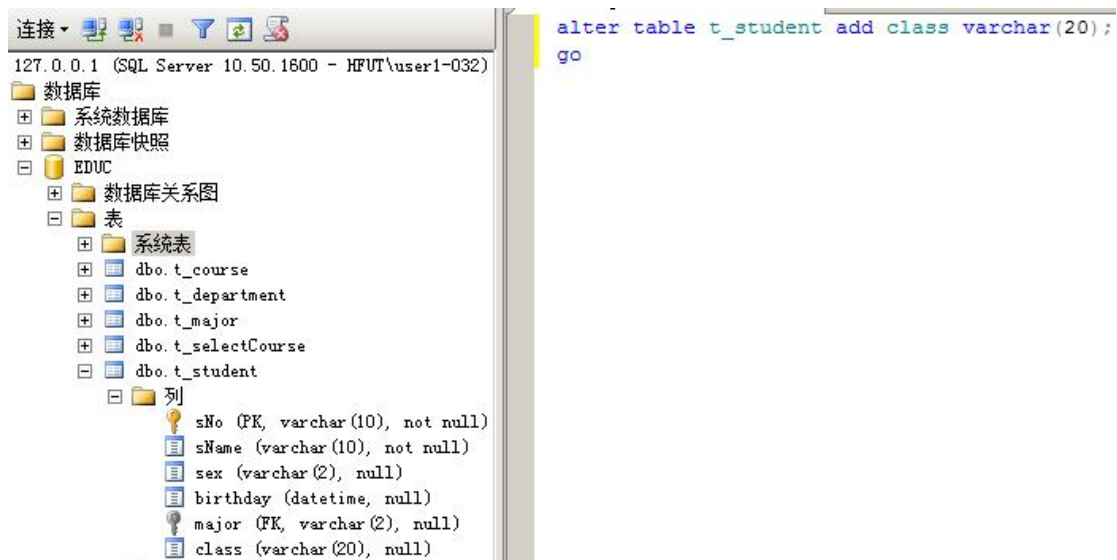


图 2.1 修改学生表结构

2) 为课程表添加一条完整性约束条件：课程名不允许为空值。

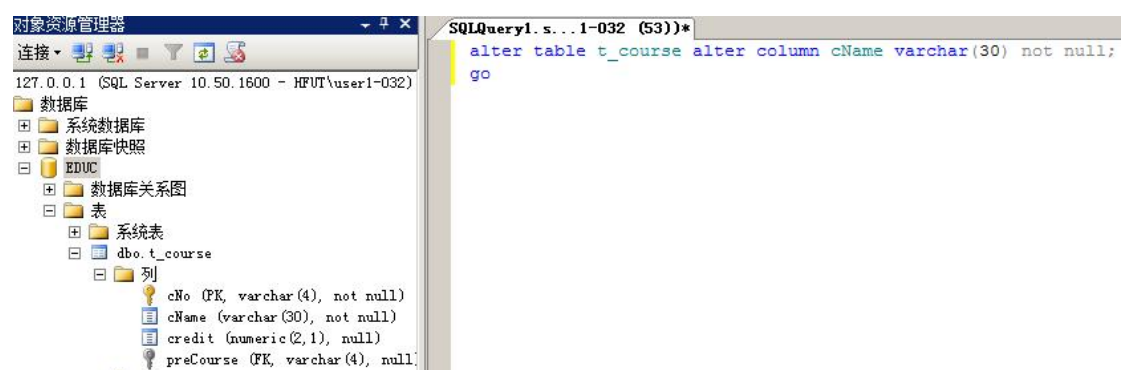


图 2.2 修改课程表结构

3) 删除学生表中的班级属性

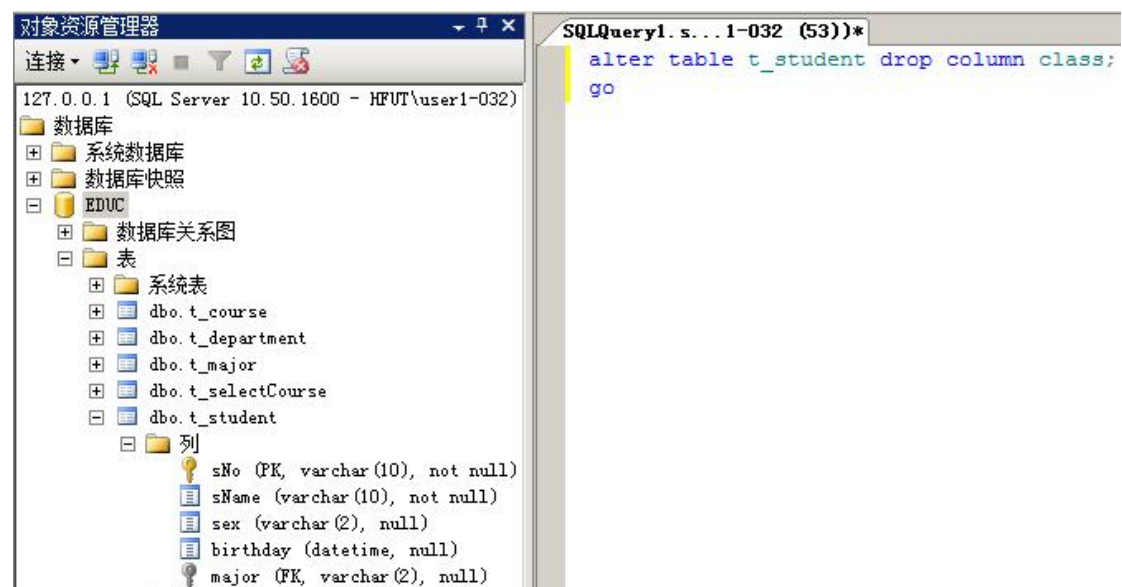


图 2.3 修改学生表结构

4) 删除学生表中对学生姓名的唯一值限制。

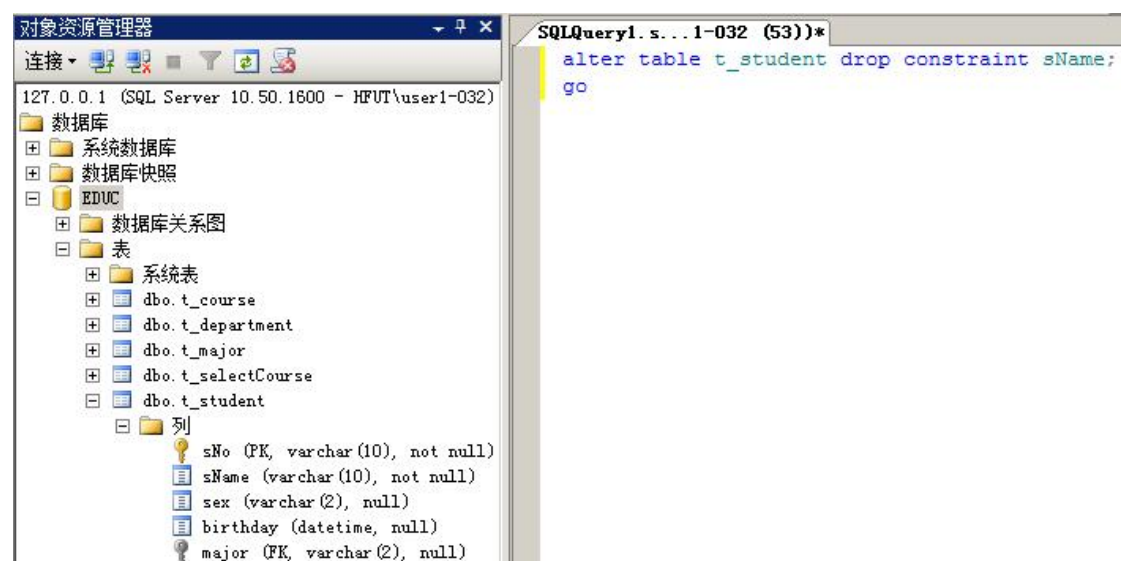


图 2.4 修改学生表结构

3. 删除基本表

1) 创建一个新的基本表

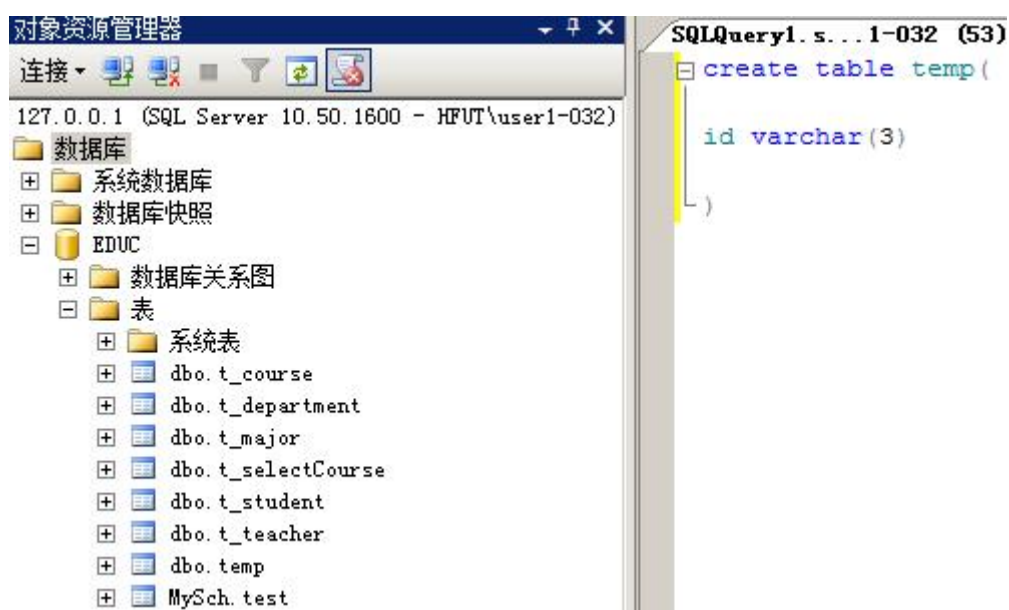


图 3.1 创建基本表

2) 删除基本表

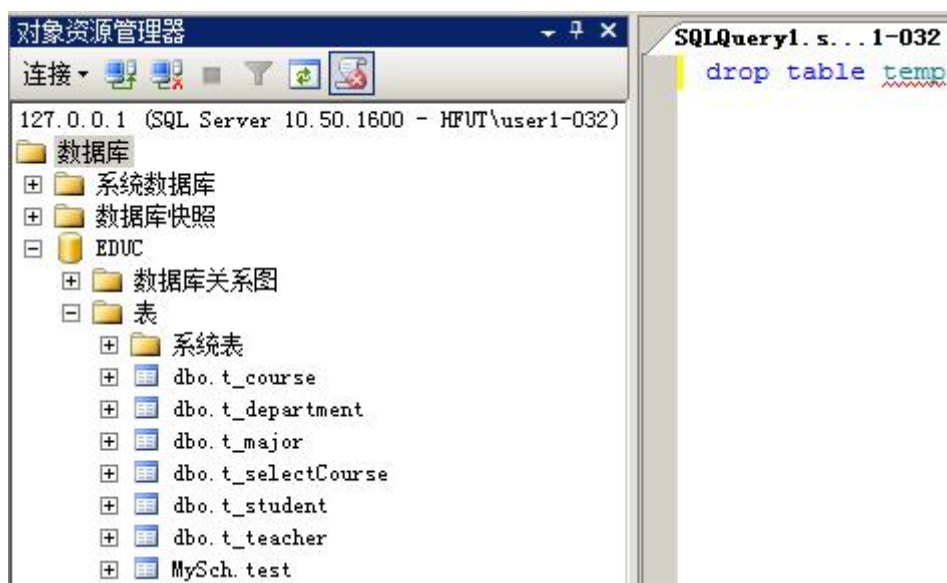


图 3.2 删除基本表

五、实验总结

通过本次实验，我掌握了：

- 1) 创建基本表并为表设置完整性约束
- 2) 修改表结构以及完整性约束
- 3) 删除基本表

实验三 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) 应用场景：在教务管理系统中，经常需要通过学生的姓名查询学生的基本信息，学生人数大概在 10000~20000 人，为了提高这类查询的效率，需要为学生表的学生姓名创建一个普通降序索引，索引名称为：Stusname。

用 SQL 语句完成上述任务。

(2) 应用场景：在应用中，学生表与专业表经常会一起使用，比如在查询学生信息时会同时需要了解学生所学的专业信息。故这两个表经常需要进行连接，为了提高连接效率，需要为学生表的专业编号创建一个普通升序索引，索引名称为：Stuspro。

用 SQL 语句完成上述任务。

(3) 应用场景：学生选课表 SC 是 EDUC 数据库中数据量最大的表，数据的变化也最频繁，实际应用中数据操作的条件以：Where Sno=' 20200001' and Cno=' c101' 的形式居多，为了提高查询效率，需要为 SC 表的学号和课程号创建一个组合索引，其中学号为升序，课程号为降序。索引名称为：SCsno_cno。

用 SQL 语句完成上述任务。

(4) 应用场景：学校教学管理规定，不同的课程使用不同的课程编号，且课程名称也应该不同，为此在创建基本表时已经为课程表的课程名定义了一个完整性约束规则：课程名必须唯一，若系统自动创建的唯一索引消失了，需要重新为课程名创建一个唯一索引，索引名为：Unicname。

用 SQL 语句完成上述任务。

2. 用 SQL 语句删除索引 Stuspro。

四、实验步骤

1. 用 SQL 语句创建所要求的索引

在 SQL Server Management Studio 的查询窗口中，将 EDUC 设置为当前数据库，依次定义实验内容所要求的 4 个索引。

语句参考：

```
create index Stusname on student(sno desc)
create index Stuspro on student(sproc asc)
create index SCsno_cno on selectCourse(sno, cno desc)
```

create index Unicname on course(cname)

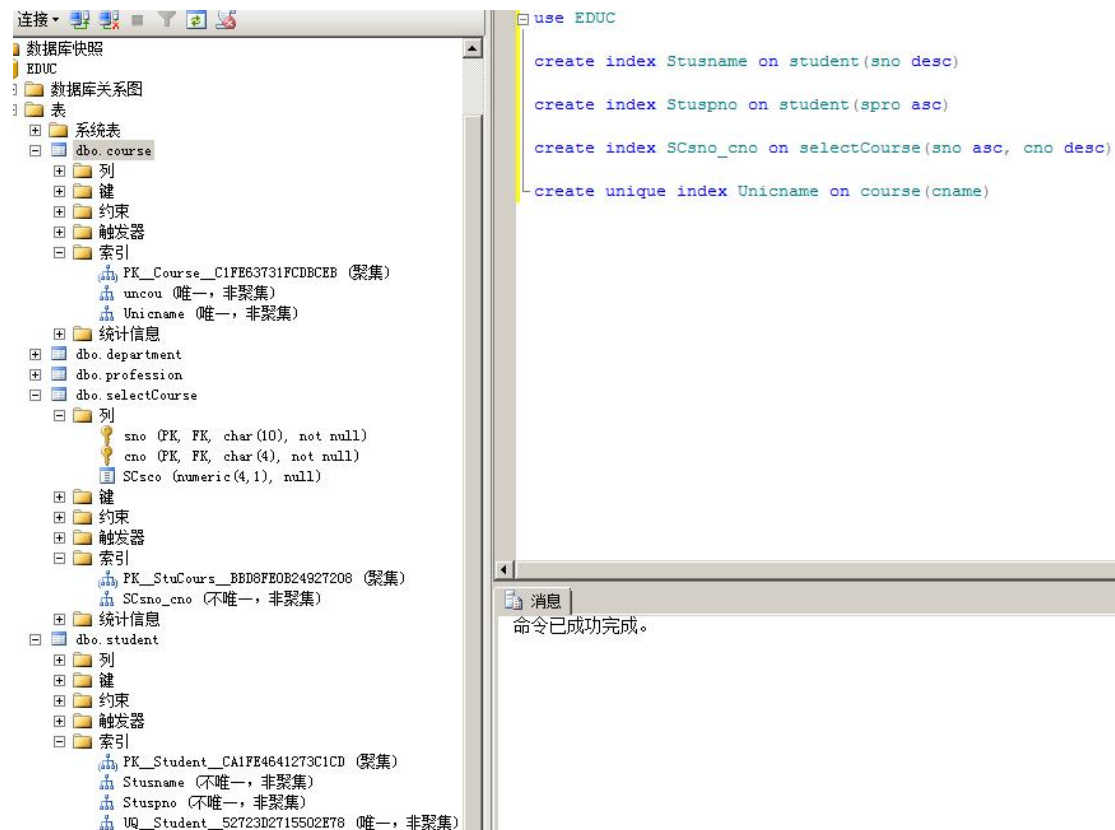


图 1 创建索引结果

2. 用 SQL 语句删除索引 Stuspro。

语句参考：

drop index Stuspro on student

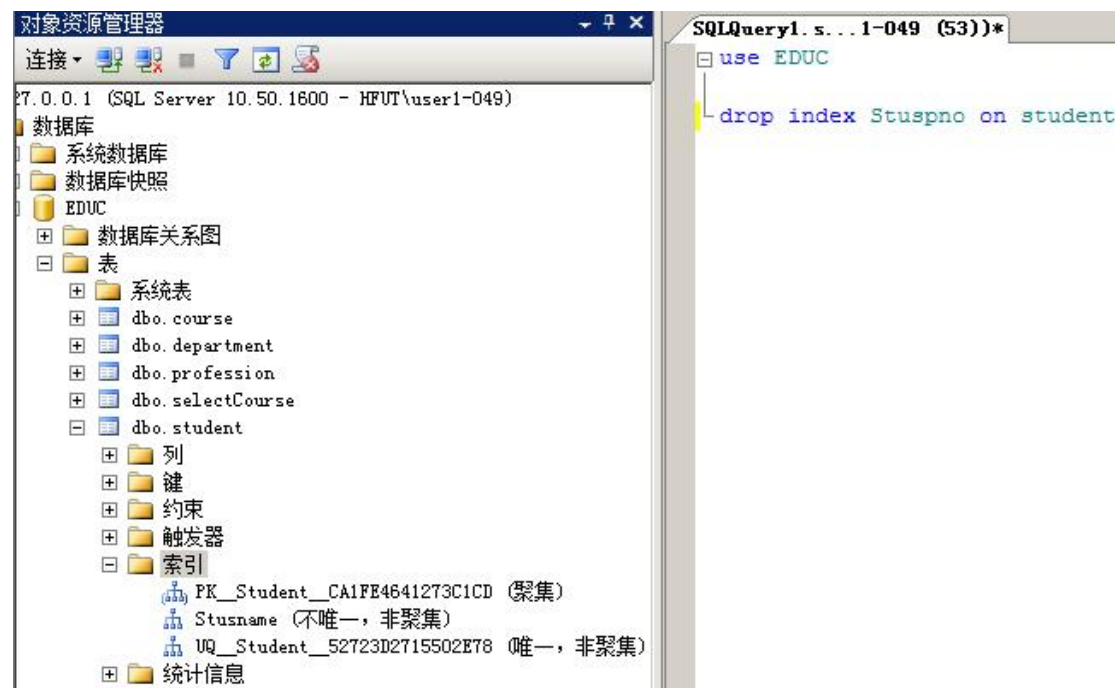


图 2 删除索引结果

五、实验总结

在本次实验过程中，我学到的内容有

- 1) 创建普通升序/降序索引
- 2) 创建组合索引
- 3) 创建唯一索引
- 4) 删除索引

另外，借此次实验，我回顾了索引的功能：索引的主要作用是为了加快查询速度。当数据库中的数据量非常庞大时，若采用普通的查询步骤会非常耗时；通过给属性增添索引，可以使得在检索过程中使用基于内容的查询，而不必只进行逐行顺序查询，大大加快查询速度。常见的索引类型有：顺序索引、B+树索引、哈希索引、位图索引等。

实验四 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、实验要求

1. 元组要求，系信息表的元组不少于 4 个，专业表不少于 8 个，学生表不少于 15 个，课程表不少于 10 个，选课表 SC 的元组不少于 20。
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. 数据插入操作

依次将准备好的数据用 Insert Into 语句将数据插入相应的基本表：

- (1) 正确数据的插入，这些数据应该不会违反完整性约束
- (2) 注意基本表的先后插入顺序，体会参照完整性。特别要注意课程表的数据插入与其他基本表的不同。
- (3) 对每一个基本表，插入错误的的数据，这些数据应该是事先准备好，且违反了基本表的某一个约束条件的。

2. 数据修改

用 Update 语句完成下列任务：

- (1) 修改某位学生的专业号，要求包括：修改语句成功执行和修改操作被拒绝两种情形。
- (2) 修改某门课学生的成绩，学生、课程及成绩由学生自行确定。
- (3) 将选课表中 50~59 之间的成绩整体增加 10 分。
- (4) 将“Java 语句”课程的先修课改为“c209”。（可根据表中数据调整条件数据）

3. 数据删除

用 Delete from 语句完成下列任务：

- (1) 先在学生表中插入一个专业号为空值的元组。
- (2) 删除学生表中所有未设定专业号的学生。

四、实验步骤

1. 数据插入操作

1) 向系表中插入数据, 均满足完整性约束



图 1.1 向系表中插入数据

2) 向专业表中插入数据, 均满足完整性约束



图 1.2 向专业表中插入数据

3) 向学生表中插入数据, 破坏实体完整性



图 1.3 向学生表中插入数据（破坏实体完整性）

4) 向学生表中插入数据，破坏参照完整性

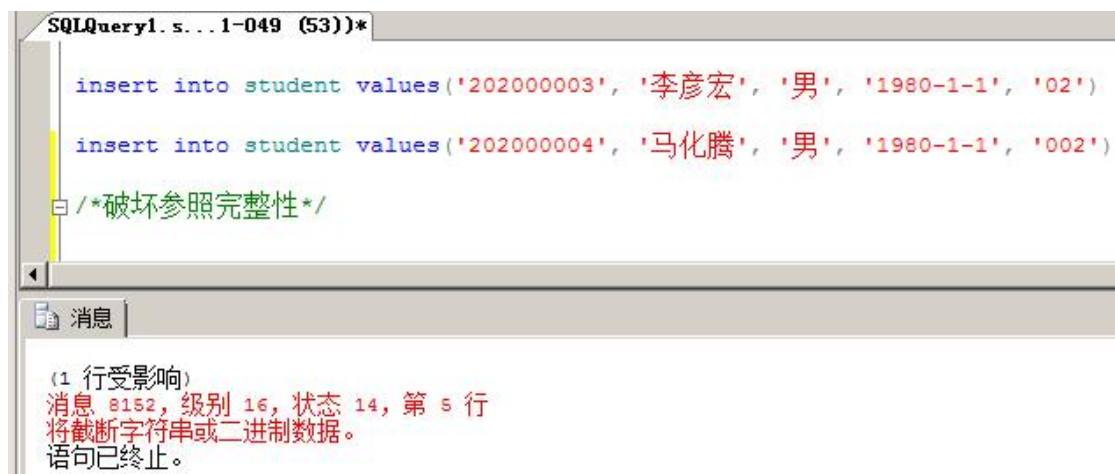


图 1.4 向学生表中插入数据（破坏参照完整性）

5) 向表中插入数据，



图 1.5 向学生表中插入数据（破坏用户定义完整性）

6) 学生表插入结果

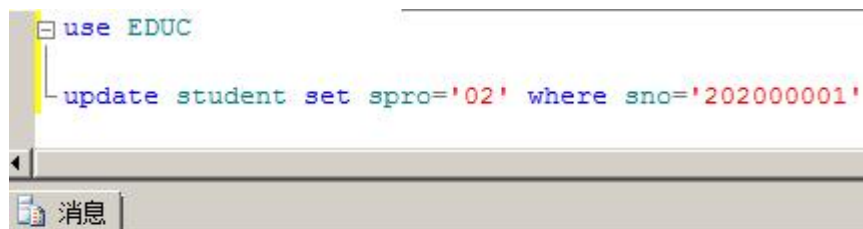

```
select * from student
```

	sno	sname	Ssex	sdate	spro
1	202000001	图灵	男	1900-01-01 00:00:00.000	01
2	2020000010	张姐	女	1990-01-01 00:00:00.000	05
3	2020000011	韩梅梅	女	1990-01-01 00:00:00.000	06
4	2020000012	李雷	男	1990-01-01 00:00:00.000	06
5	2020000013	拜登	男	1900-01-01 00:00:00.000	07
6	2020000014	瓦妮莎	女	1990-01-01 00:00:00.000	07
7	2020000015	娜塔莎	女	1990-01-01 00:00:00.000	08
8	2020000016	福尔摩斯	男	1990-01-01 00:00:00.000	08
9	202000002	吴恩达	男	1980-01-01 00:00:00.000	01
10	202000003	李彦宏	男	1980-01-01 00:00:00.000	02
11	202000004	马化腾	男	1980-01-01 00:00:00.000	02
12	202000005	熊大	男	1900-01-01 00:00:00.000	03
13	202000006	熊二	男	1900-01-01 00:00:00.000	03
14	202000007	梁思成	男	1901-01-01 00:00:00.000	04
15	202000008	贝聿铭	男	1917-04-26 00:00:00.000	04
16	202000009	扫地僧	男	1900-01-01 00:00:00.000	05

图 1.6 学生表插入结果

2. 数据修改

1) 成功修改专业号



(1 行受影响)

图 2.1 成功修改专业号

2) 拒绝修改专业号



图 2.2 拒绝修改专业号

3. 数据删除

1) 在学生表中插入一个专业号为空值的元组

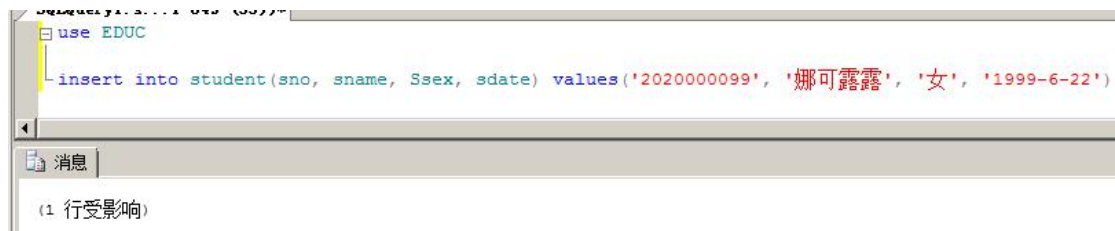


图 3.1 插入语句及结果

- 2) 删除学生表中所有未设定专业号的学生

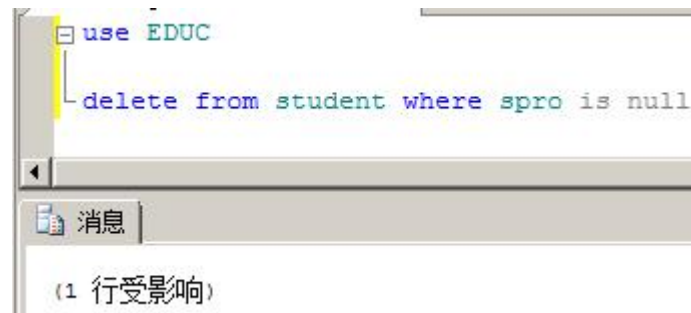


图 3.2 删除语句及结果

五、实验总结

在本次实验中，我掌握到的知识有：

- 1) 掌握了 insert、update、delete 等数据操纵语言的用法
- 2) 体会到了数据更新操作对完整性约束的影响

借此实验，我回顾了数据库完整性的内容，主要包括：

- 1) 实体完整性：与主码相关
- 2) 参照完整性：与外码相关
- 3) 用户定义完整形：与用户定义的 unique、not null 等相关

实验五 数据库的分离与附加

一、实验目的

理解数据库为何要进行分离和附加？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

当 SQL Server 服务器运行时，该服务器上所有的数据库自动处于运行状态，而运行中的数据库文件是无法进行数据库文件的拷贝的。实验任务完成后，往往需要将数据库文件，以及在实验过程中产生的各种资料拷贝到 U 盘保存起来。

本实验的目的就是将需要拷贝的数据库（如 EDUC）从数据库服务器上分离出来，使其处于非运行状态，从而可以正常进行数据库文件的拷贝。

2. 数据库附加

数据库附加是为了将某个数据库的文件附加在数据库服务器上，使其处于运行状态。比如：学生在上一次实验结束后，通过数据库分离，将数据库的文件拷贝在自己的 U 盘上，下一次实验时对 U 盘上的数据库文件进行附加，该数据库中原有的数据对象及数据可以正常使用。

四、实验步骤

1. 数据库分离

在对象窗口中找到需要分离的数据库（如 EDUC），右击数据库名，在菜单中点击分离：

检验结果：



图 1 数据库分离

2. 数据库附加

在对象窗口中，右击数据库，点击“附加”，在对象窗口中，右击数据库，点击“附加”：
出现数据库附加窗口：点击“添加”选择需要添加的数据库文件：



图 2 数据库附加

检验结果：



图 3 数据库分离结果

五、实验总结

通过本次实验，我掌握了从数据库管理系统中分离和附加数据库。

实验六 数据查询

一、实验目的

1. 熟悉 SQL Server 交互式使用 Select 语句的方法；
2. 掌握执行和检查 SELECT 语句结果的方法；
3. 熟练运用单表查询、多表查询、嵌套查询等完成实际查询要求。

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

1. 单表查询

- (1) 查询全体学生的信息；
- (2) 根据专业编号查询学生的学号、性别和年龄；
- (3) 查询未设定先修课的所有课程的信息；
- (4) 查询选修了‘C101’（学生自学设定课程号）号课程，且成绩达到 80 分的学号。

2. 多表连接查询

- (1) 查询“计算机系”全体学生的学号、姓名、专业名称；
- (2) 查询非“软件工程”专业，年龄小于 20 的学生信息；
- (3) 查询先修课是“操作系统”的所有课程的信息；
- (4) 查询至少选修了‘c110’和‘c210’（课程号由学生自学确定！）课程的学生学号和姓名；
- (5) 查询未选修“计算机网络”课程的学生学号、姓名、性别和专业号；
- (6) 查询未选修任何课程的学生学号和姓名。
- (7) 查询未被学生选修的课程号、课程名、先修课。
- (8) 用派生关系查询平均成绩达到 90 分的学生学号、姓名和平均成绩。
- (9) 查询学生的学号，只要先修课是“c331”的课程他们都选修了。
- (10) 查询未参加课程“c209”考试的学生名单（学号、姓名、专业号）。

四、实验步骤

1. 单表查询

- 1) 查询全体学生的信息

select * from student

	sno	sname	Ssex	sdate	spro
1	202000001	图灵	男	1900-01-01 00:00:00.000	02
2	2020000010	张姐	女	1990-01-01 00:00:00.000	05
3	2020000011	韩梅梅	女	1990-01-01 00:00:00.000	06
4	2020000012	李雷	男	1990-01-01 00:00:00.000	06
5	2020000013	拜登	男	1900-01-01 00:00:00.000	07
6	2020000014	瓦妮莎	女	1990-01-01 00:00:00.000	07
7	2020000015	娜塔莎	女	1990-01-01 00:00:00.000	08
8	2020000016	福尔摩斯	男	1990-01-01 00:00:00.000	08
9	202000002	吴恩达	男	1980-01-01 00:00:00.000	01
10	202000003	李彦宏	男	1980-01-01 00:00:00.000	02
11	202000004	马化腾	男	1980-01-01 00:00:00.000	02
12	202000005	熊大	男	1900-01-01 00:00:00.000	03
13	202000006	熊二	男	1900-01-01 00:00:00.000	03
14	202000007	梁思成	男	1901-01-01 00:00:00.000	04
15	202000008	贝聿铭	男	1917-04-26 00:00:00.000	04
16	202000009	扫地僧	男	1900-01-01 00:00:00.000	05

图 1.1 全体学生信息

- 2) 根据专业编号查询学生的学号、性别和年龄

SQLQuery1. s...1-049 (53))*

```
use EDUC
select sno, sname, (YEAR(GETDATE()) - YEAR(sdate)) from student where spro='01'
```

	sno	sname	(无列名)
1	202000002	吴恩达	40

图 1.2 专业号为 01 的学生的学号、性别和年龄

- 3) 查询未设定先修课的所有课程的信息

SQLQuery1. s...1-049 (53))*

```
use EDUC
select * from course where cpre is null
```

	cno	cname	csco	cpre
1	1001	高等数学	6.0	NULL
2	1002	线性代数	3.0	NULL
3	1007	汇编语言	1.0	NULL

图 1.3 未设定先修课的所有课程的信息

4) 查询选修了' C101' (学生自学设定课程号) 号课程且成绩达到 80 分的学号

```

use EDUC
select sno from selectCourse where cno='1001' and SCsco > 80

```

	sno
1	202000001
2	202000011

图 1.4 选修了 '1001' 号课程且成绩达到 80 分的学号

2. 多表连接查询

1) 查询“计算机系”全体学生的学号、姓名、专业名称

```

use EDUC
select s.sno, s.sname, p.pname from student s, profession p, department d where d.dname='计算机与信息系' and s.spro = p.pno and p.pdepartment = d.dno

```

	sno	sname	pname
1	202000001	图灵	物联网工程
2	202000002	吴思达	计算机科学与技术
3	202000003	李彦宏	物联网工程
4	202000004	马化腾	物联网工程

图 2.1 计算机系全体学生的学号、姓名、专业名称

2) 查询非“物联网工程”专业，年龄小于 40 的学生信息

```

--查询非物联网工程专业， 年龄小于40的学生信息--
select sno, Sname, Ssex, (YEAR(GETDATE()) - YEAR(sdate)) age, dname, pname
from student, profession, department
where pname != '物联网工程' and (YEAR(GETDATE()) - YEAR(sdate)) < 40
and spro = pno and pdepartment = dno

```

	sno	Sname	Ssex	age	dname	pname
1	202000007	梁思成	男	30	建筑与工程系	建筑学
2	202000008	贝聿铭	男	30	建筑与工程系	建筑学
3	202000009	扫地僧	男	30	资源与环境工程系	环境工程
4	202000010	张姐	女	30	资源与环境工程系	环境工程
5	202000011	韩梅梅	女	30	资源与环境工程系	环境科学
6	202000012	李雷	男	30	资源与环境工程系	环境科学
7	202000013	拜登	男	30	英语系	外贸英语

图 2.2 非“物联网工程”专业，年龄小于 40 的学生信息

3) 查询先修课是高等数学的所有课程的信息

```

--查询先修课是高等数学的所有课程的信息--
select a.cno, a.cname, a.cscs from course a, course b
where a.cpre = b.cno and b.cname = '高等数学'

```

	cno	cname	cscs
1	1003	信号与系统	3.0

图 2.3 先修课是高等数学的所有课程的信息

- 4) 查询至少选修了 1001 和 1006 课程的学生学号和姓名

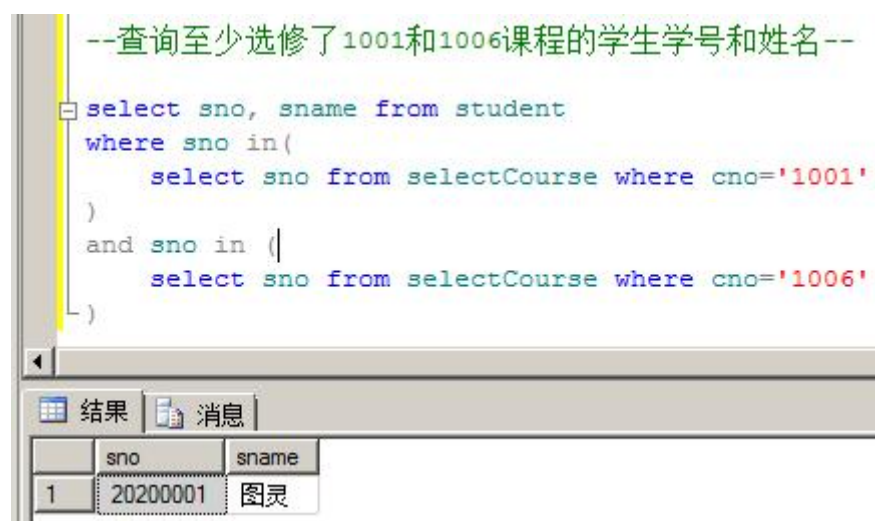


图 2.4 至少选修了 1001 和 1006 课程的学生学号和姓名

- 5) 查询未选修汇编语言课程的学生学号、姓名、专业号

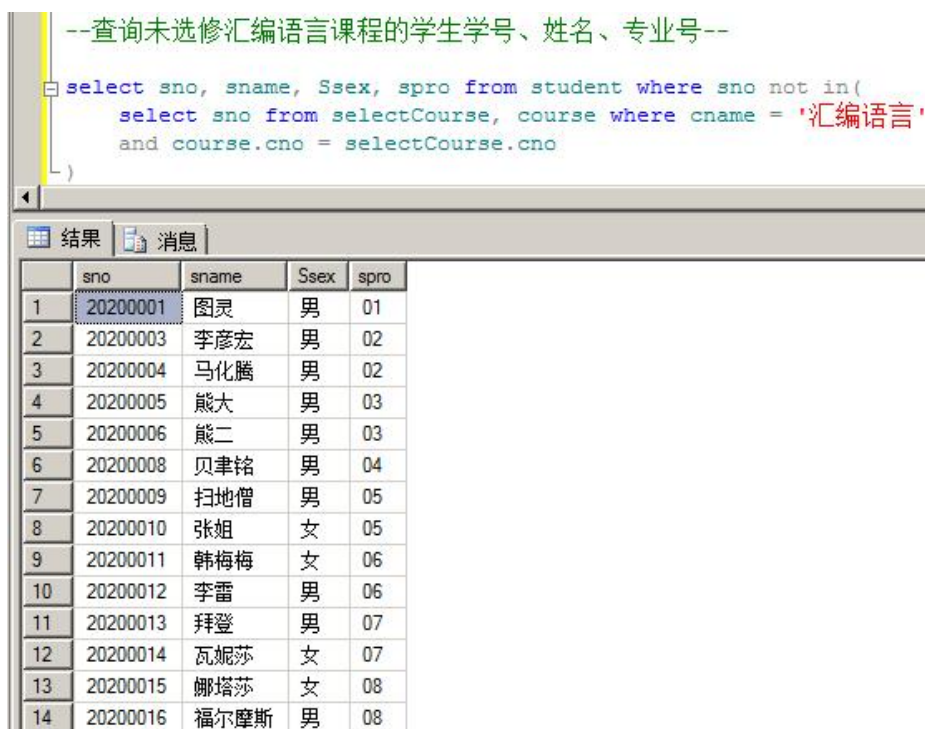


图 2.5 未选修汇编语言课程的学生学号、姓名、专业号

- 6) 查询未选修任何课程的学生学号和姓名

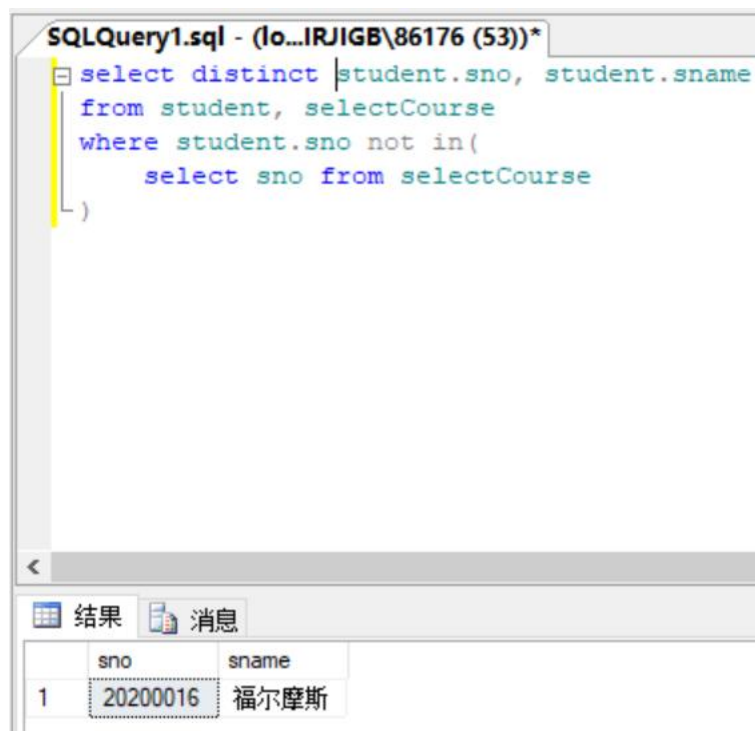


图 2.6 未选修任何课程的学生学号和姓名

- 7) 查询未被学生选修的课程号、课程名、先修课



图 2.7 未被学生选修的课程号、课程名、先修课

- 8) 用派生关系查询平均成绩达到 90 分的学生学号、姓名和平均成绩

SQLQuery1.sql - (lo...IRJIGB\86176 (53))*

```

select student.sno, student.sname, AVG(SCsco)
from student, selectCourse
where selectCourse.sno = student.sno
group by student.sno, student.sname
having AVG(SCsco) >= 90

```

<

结果 消息

	sno	sname	(无列名)
1	20200001	图灵	91.500000
2	20200005	熊大	98.000000
3	20200010	张姐	100.000000
4	20200014	瓦妮莎	90.000000
5	20200015	娜塔莎	95.000000

图 2.8 平均成绩达到 90 分的学生学号、姓名和平均成绩

- 9) 查询学生的学号，只要先修课是“1001”的课程他们都选修了

SQLQuery1.sql - (lo...IRJIGB\86176 (53))*

```

select distinct student.sno
from student, course c1, course c2
where c2.cno = '1001' and c1.cpre = c2.cno
and selectCourse.cno = c1.cno
and student.sno = selectCourse.sno

```

<

结果 消息

	sno
1	20200001

图 2.9 选修了全部先修课是“1001”的课程的学生学号

- 10) 查询未参加课程“01”考试的学生名单（学号、姓名、专业号）

SQLQuery1.sql - (lo...IRJIGB\86176 (53))*			
<pre> select distinct student.sno, student.sname, student.spro from student, selectCourse where student.sno not in(select selectCourse.sno from selectCourse where cno = '1001') </pre>			
结果 消息			
	sno	sname	spro
1	20200002	吴恩达	01
2	20200003	李彦宏	02
3	20200004	马化腾	02
4	20200005	熊大	03
5	20200006	熊二	03
6	20200007	梁思成	04
7	20200008	贝聿铭	04
8	20200009	扫地僧	05
9	20200010	张姐	05
10	20200012	李雷	06
11	20200013	拜登	07
12	20200014	瓦妮莎	07
13	20200015	娜塔莎	08
14	20200016	福尔摩斯	08

图 2.10 未参加课程“01”考试的学生名单

五、实验总结

在本次实验过程中，我掌握了以下内容

- 1) 简单的单表查询
- 2) 简单的条件查询
- 3) 多表连接查询
- 4) 嵌套查询
- 5) 初步使用聚集函数

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 查询窗口的使用，加深对 SQL 语言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) 查询每个专业的专业编号、专业名称、学生人数、学生的平均年龄。
- 3) 查询选修了课程的学生人数。
- 4) 查询选修了课程“c218”，且成绩超过该课程平均成绩的学生学号、姓名。
- 5) 查询每门课的课程号、选课人数、平均成绩、最高分和最低分。
- 6) 查询选修课程超过 2 门课的学生学号和姓名。

四、实验步骤

- 1) 查询学生的总人数

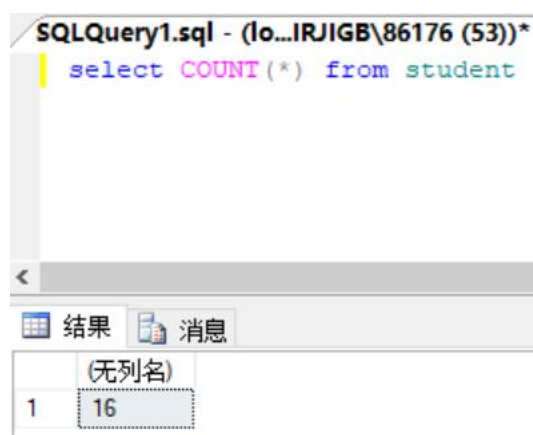


图 1 学生总人数

- 2) 查询每个专业的专业编号、专业名称、学生人数、学生的平均年龄

SQLQuery1.sql - (10...IKRJGB\86176 (56)) SQLQuery1.sql - (10...IKRJGB\86176 (53))

```

use EDUC
select profession.pno,profession.pname,count(student.sno) as 学生人数 ,
      avg(datediff(year,sdate,getdate())) as 年龄
from student,profession
where pno = student.spro
group by profession.pno,profession.pname

```

结果 消息

	pno	pname	学生人数	年龄
1	01	计算机科学与技术	1	41
2	02	物联网工程	2	41
3	03	土木工程	3	41
4	04	建筑学	2	31
5	05	环境工程	2	31
6	06	环境科学	2	31
7	07	外贸英语	2	36
8	08	商务英语	2	41

图 2 每个专业的专业编号、专业名称、学生人数、学生的平均年龄

3) 查询选修了课程的学生人数

```

select COUNT(*) as nums
from student
where student.sno in(
    select sno from selectCourse
)

```

结果 消息

	nums
1	15

图 3 选修了课程的学生人数

4) 查询选修了课程“1001”，且成绩超过该课程平均成绩的学生学号、姓名

```

select student.sname, student.sno
from student, selectCourse, course
where course.cno = selectCourse.cno
    and course.cno = '1001' and student.sno = selectCourse.sno
    and SCsco >= any(
        select AVG(SCsco) from selectCourse, course
        where course.cno = selectCourse.cno and course.cno = '1001'
    )

```

结果 消息

	sname	sno
1	图灵	20200001

图 4 选修了课程“1001”，且成绩超过该课程平均成绩的学生学号、姓名

5) 查询每门课的课程号、选课人数、平均成绩、最高分和最低分

```

select distinct course.cno, COUNT(course.cno) as nums,
        AVG(SCsco) as average, MAX(SCsco) as maximum, MIN(SCsco) as minimum
from course, selectCourse
where course.cno = selectCourse.cno
group by course.cno

```

	cno	nums	average	maximum	minimum
1	1006	2	91.000000	93.0	89.0
2	1010	2	98.500000	100.0	97.0
3	1001	2	62.000000	90.0	34.0
4	1007	2	61.500000	63.0	60.0
5	1008	2	53.000000	54.0	52.0
6	1005	2	97.000000	99.0	95.0
7	1009	2	50.500000	59.0	42.0
8	1004	2	91.500000	93.0	90.0
9	1002	2	86.500000	95.0	78.0
10	1003	2	82.000000	92.0	72.0

图 5 每门课的课程号、选课人数、平均成绩、最高分和最低分

6) 查询选修课程超过 2 门课的学生学号和姓名

```

select distinct student.sno, student.sname
from student, selectCourse
where student.sno = selectCourse.sno
group by student.sno, student.sname
having COUNT(student.sno) >= 2

```

	sno	sname
1	20200001	图灵
2	20200002	吴恩达
3	20200003	李彦宏
4	20200004	马化腾
5	20200005	熊大

图 6 选修课程超过 2 门课的学生学号和姓名

五、实验总结

在本次实验中，我掌握了以下内容：

- 1) 使用聚集函数进行查询
- 2) group by 后面的属性必须出现在 select 后
- 3) where 后不能有聚集函数

实验八 视图的定义与使用

一、实验目的

1. 熟悉和掌握对数据表中视图的查询操作和 SQL 语句的使用；
2. 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；
3. 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、实验内容

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；
- 2) 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；
- 3) 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；
- 4) 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；
- 5) 定义一个学生表的行列子集视图 V_S：该视图包含学生的学号、姓名、专业号；
- 6) 定义一个视图 V_SC：该视图包含学号、姓名、课程号、成绩。

2. 使用视图（下列操作只允许对视图进行）

- 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。
- 3) 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。
- 4) 查询 2000 年出生的学生学号和姓名。
- 5) 将学生“张三”的专业号改为“03”（数据视实际数据而定）。
- 6) 利用视图 V_SC 将学号为“20180001”、课程号为“c111”的成绩改为 85 分。

四、实验步骤

1. 定义视图

- 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


```
create view V_SC_G as
select student.sno, student.sname, course.cno,
       course.cname as course_name, selectCourse.SCsco
from student, selectCourse, course
where student.sno = selectCourse.sno and selectCourse.cno = course.cno
```

消息
命令已成功完成。

图 1.1 定义视图 V_SC_G

- 2) 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

```
create view V_YEAR(Sno,Sname,Syear)
as
select student.sno, student.sname,
       YEAR(GETDATE())-datediff(year, student.sdate, getdate())
       as 出生年份
from student
```

消息
命令已成功完成。

图 1.2 定义一个反映学生出生年份的视图 V_YEAR

- 3) 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；

```
create view V_AVG_S_G(sno,count,avg)
as
select selectCourse.sno, count(*) as 选修课数量,
       avg(selectCourse.SCsco) as 平均成绩
from selectCourse group by selectCourse.sno
```

消息
命令已成功完成。

图 1.3 定义视图 V_AVG_S_G

- 4) 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


```
create view V_AVG_C_G(cno, count, avggrade)
as
select selectCourse.cno, count(selectCourse.sno) as 选修人数,
      AVG(selectCourse.SCsco) as 平均成绩
from selectCourse, student, Course
where selectCourse.sno=student.sno and selectCourse.cno=Course.cno
group by selectCourse.cno;
```

消息
命令已成功完成。

图 1.4 定义视图 V_AVG_C_G

- 5) 定义一个学生表的行列子集视图 V_S: 该视图包含学生的学号、姓名、专业号

```
create view V_S(Sno, Sname, major)
as
select student.sno, student.sname, student.spro from student
```

消息
命令已成功完成。

图 1.5 定义一个学生表的行列子集视图 V_S

- 6) 定义一个视图 V_SC: 该视图包含学号、姓名、课程号、成绩

```
create view V_SC(Sno, Sname, class, socre)
as
select student.sno, student.sname, selectCourse.cno, selectCourse.SCsco
from student, selectCourse
where student.sno = selectCourse.sno
```

消息
命令已成功完成。

图 1.6 定义一个视图 V_SC

2. 使用视图

- 1) 查询以上所建的视图结果

select *	from	V SC G
select *	from	V YEAR
select *	from	V AVG S G
select *	from	V AVG C G
select *	from	V S
select *	from	V SC

	sno	sname	cno	course_name	SCsco
4	20200012	李雷	1002	线性代数	78.0
5	20200003	李彦宏	1003	信号与系统	92.0
6	20200003	李彦宏	1008	计算机组成原理	52.0
7	20200007	梁思成	1007	汇编语言	60.0
8	20200004	马化腾	1004	数据结构	93.0
9	20200004	马化腾	1009	软件工程	59.0
10	20200015	娜塔莎	1005	离散数学	95.0
11	20200009	扫地僧	1009	软件工程	42.0

	Sno	Sname	Syear
1	20200001	图灵	1980
2	20200002	吴恩达	1980
3	20200003	李彦宏	1980
4	20200004	马化腾	1980
5	20200005	熊大	1980
6	20200006	熊二	1980
7	20200007	梁思成	1990
8	20200008	贝聿铭	1990

图 2.1 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

select distinct	V SC G.sno	sname	SCsco
from	V SC G,V AVG S G		
where	V AVG S G.avg>90	and	V SC G.sno=V AVG S G.sno

	sno	sname	SCsco
1	20200001	图灵	90.0
2	20200001	图灵	93.0
3	20200005	熊大	97.0
4	20200005	熊大	99.0
5	20200010	张姐	100.0
6	20200015	娜塔莎	95.0

图 2.2 平均成绩为 90 分及以上的学生学号、姓名和成绩

3) 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

```

select distinct V_SC.G.sno, V_AVG.C.G.cno, V_SC.G.SCsco, avggrade
from V_SC.G, V_AVG.C.G
where V_AVG.C.G.cno=V_SC.G.cno
and V_SC.G.SCsco>V_AVG.C.G.avggrade

```

	sno	cno	SCsco	avggrade
1	20200001	1001	90.0	62.000000
2	20200002	1002	95.0	86.500000
3	20200003	1003	92.0	82.000000
4	20200004	1004	93.0	91.500000
5	20200005	1005	99.0	97.000000
6	20200001	1006	93.0	91.000000
7	20200002	1007	63.0	61.500000
8	20200008	1008	54.0	53.000000
9	20200004	1009	59.0	50.500000
10	20200010	1010	100.0	98.500000

图 2.3 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

- 4) 查询 2000 年出生的学生学号和姓名

```

select V_YEAR.sno, student.sname
from V_YEAR, student
where V_YEAR.Syear=2000 and V_YEAR.sno=student.sno

```

sno	sname
-----	-------

图 2.4 2000 年出生的学生学号和姓名

- 5) 将学生“图灵”的专业号改为“03”（数据视实际数据而定）

```

update V_S set major = '03'
where V_S.Sname = '图灵';

```

消息

(1 行受影响)

图 2.4 将学生“图灵”的专业号改为“03”

- 6) 利用视图 V_SC 将学号为“20200001”、课程号为“1001”的成绩改为 85 分

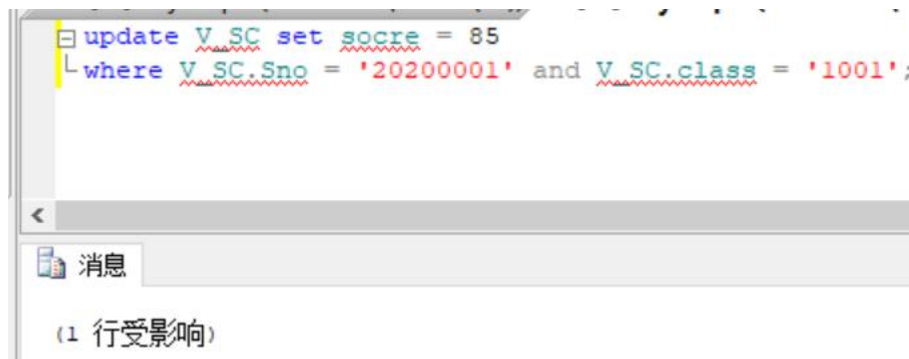


图 2.6 将学号为“20200001”、课程号为“1001”的成绩改为 85 分

五、实验总结

在本次实验中，我掌握了

- 1) 视图的定义和创建
- 2) 使用视图进行增删改查等操作

此外，借此次实验我再次回顾了视图的概念和作用；视图是一张虚表，它不存放数据，只存放视图的定义，视图可以简化用户的操作，只展示一部分数据给用户，起到一定的安全性作用。

实验九 用户及其权限管理

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。数据库的数据访问只允许合法的数据库用户在权限控制下进行。

使用者必须先获得一个合法的身份（数据库用户），每个用户只有在得到授权才能进行与其权限相符的数据操作，角色是权限的集合，当某个角色被授予用户后，该用户即可获得该角色的所有权限。

SQL Server 是以数据库服务器方式启动的，以数据库服务器用户（登录名）身份登录，为了使某个服务器用户登录成功后能够以数据库用户的身份进行数据库操作，需要建立登录名（服务器用户）与数据库用户间的映射关系，每个登录名可以关联到多个数据库，但只能关联特定数据库中的一个用户。

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 Mylog，设置登录初始口令
- 2、创建一个 EDUC 数据库用户 user1，使其关联登录名 Mylog
- 3、创建一个 EDUC 数据库角色 role1
- 4、对用户及角色授权
- 5、验证用户授权
- 6、收回用户权限

四、实验步骤

- 1、创建登录帐号 Mylog，设置登录初始口令

```
create login Mylog with password = '1234'
```

- 2、创建一个 EDUC 数据库用户 u1，使其关联登录名 Mylog

```
create user u1 for login Mylog with default_schema = DBO
```

```
create user u2 for login Mylog
```

- 3、创建一个 EDUC 数据库角色 r1

```
create role r1
```

- 4、对用户及角色授权

- 1) 将创建基本表的权限授予 r1

语句参考: grant create table to r1

2) 将查询学生表的权限授予角色 r1。

语句参考: grant select on student to r1

3) 将插入选课表和修改成绩的权限授予角色 r1。

语句参考: grant select,update(SCsco) on selectCourse to r1

4) 将角色 r1 授予用户 u1。

语句参考: exec sp_addrolemember 'r1','u1'

5) 将查询、插入和删除课程表的权限授予用户 user1。

语句参考: grant select,insert,delete on course to u1

5、验证用户授权

(1) 选用 SQL Server 用户身份验证方式, 以 Mylog 登录名重新登录



图 1 用 SQL Server 身份验证

(2) 将 EDUC 设置为当前数据库

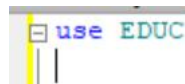


图 2 使用 EDUC

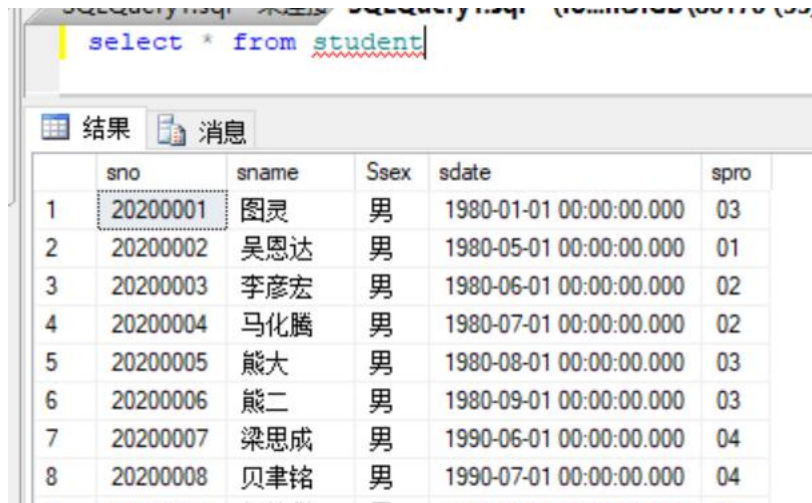
(3) 创建一个基本表, 结构由学生自行确定, 检验该用户是否具有创建基本表的权限

```
create table temp
(sno nchar(10) primary key,
sname nvarchar(20) not null unique,
ssex nchar(2) check (ssex = '男' or ssex = '女'),
-)
```



图 3 创建基本表

- (4) 查询学生表，检验用户是否已拥有查询学生表的权限

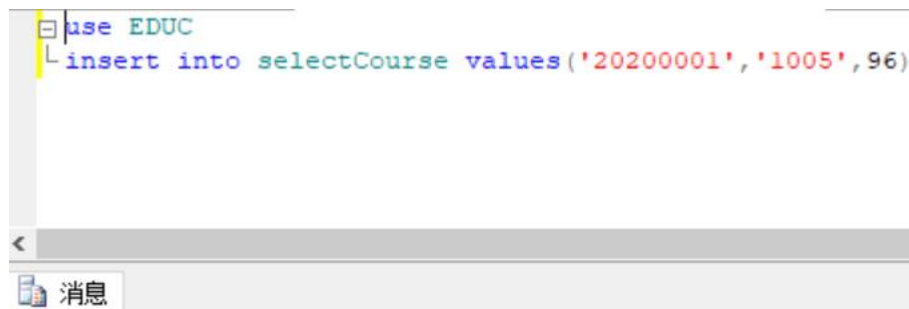


The screenshot shows a SQL query window with the command `select * from student`. Below the command, there are tabs for '结果' (Results) and '消息' (Messages). The '结果' tab is active, displaying a table with 8 rows of student data. The columns are `sno`, `sname`, `Ssex`, `sdate`, and `spro`.

	sno	sname	Ssex	sdate	spro
1	20200001	图灵	男	1980-01-01 00:00:00.000	03
2	20200002	吴恩达	男	1980-05-01 00:00:00.000	01
3	20200003	李彦宏	男	1980-06-01 00:00:00.000	02
4	20200004	马化腾	男	1980-07-01 00:00:00.000	02
5	20200005	熊大	男	1980-08-01 00:00:00.000	03
6	20200006	熊二	男	1980-09-01 00:00:00.000	03
7	20200007	梁思成	男	1990-06-01 00:00:00.000	04
8	20200008	贝聿铭	男	1990-07-01 00:00:00.000	04

图 4 查询学生表

- (5) 向选课表中插入一个元组，检验用户是否拥有该权限



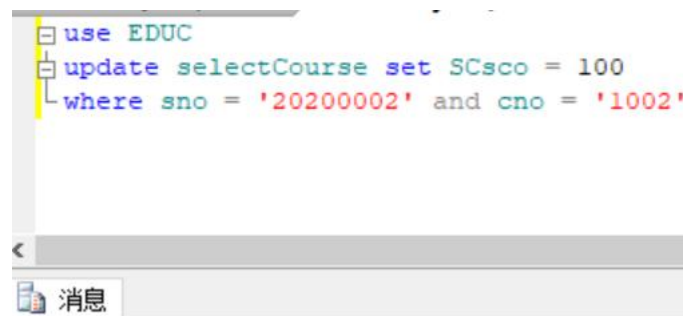
The screenshot shows a SQL query window with the command `use EDUC` and `insert into selectCourse values ('20200001', '1005', 96)`. Below the command, there is a tab for '消息' (Messages).

```
use EDUC
insert into selectCourse values ('20200001', '1005', 96)
```

(1 行受影响)

图 5 向选课表中插入数据

- (6) 修改某学生、某门课的成绩，检验用户是否具有修改成绩的权限



The screenshot shows a SQL query window with the command `use EDUC`, `update selectCourse set SCsco = 100`, and `where sno = '20200002' and cno = '1002'`. Below the command, there is a tab for '消息' (Messages).

```
use EDUC
update selectCourse set SCsco = 100
where sno = '20200002' and cno = '1002'
```

(1 行受影响)

图 6 修改选课表

- (7) 操作课程表，检验用户是否具有：查询、插入和删除的权限

```
use EDUC
select * from course
```

	cno	cname	cscd	cpre
1	1001	高等数学	6.0	NULL
2	1002	线性代数	3.0	NULL
3	1003	信号与系统	3.0	1001
4	1004	数据结构	4.0	1002
5	1005	离散数学	4.0	1004
6	1006	Java技术	2.0	NULL
7	1007	汇编语言	1.0	NULL
8	1008	计算机组成原理	4.0	1007
9	1009	软件工程	3.0	1006
10	1010	毕业设计	6.0	1009

图 7.1 查询课程表

```
use EDUC
insert into course values('1011','操作系统',5,'1002')
```

消息

(1 行受影响)

图 7.2 向课程表插入数据

```
use EDUC
delete from course where cname = '操作系统'
```

消息

(1 行受影响)

图 7.3 从课程表中删除数据

6、收回用户权限

(1) 收回角色 r1 查询学生表的权限

1) 以系统用户的身份重新登录

```
grant select on student to r1
```

2) 收回角色 r1 查询学生表的权限

```
revoke select on student from r1
```

(2) 收回用户 u1 删除课程表元组的权限

```
revoke delete on class from u1
```

(3) 验证权限收回是否有效

```
select * from student
```

```
insert into course values('1011','操作系统',5,'1002')
```

```
delete from course where cname = '操作系统'
```


五、实验总结

在本次实验过程中，我掌握了：

- 1) 用户、角色的创建
- 2) 给用户授予权限
- 3) 收回用户的权限
- 4) 对数据库的安全性有了更深入的理解