选做：设计学籍管理系统

一、需求分析

学籍数据库的有关语义如下：

1. 一个系可以若干专业，每个专业每年可以招若干班，每个班有若干学生。
2. 系里对每个专业每年都制订了教学计划，学生每年必须按照教学计划修完一定学分的课程（必修课、限选课和任选课），如2000年入学的学生大三上学期必修课30学分，限选课10学分，任选课6学分。
3. 系里的教师可以给多个班带课，但是不能给一个班带多门课程。
4. 一门课程最多允许学生一次补考，学生达到如下条件之一的被开除：
    - 一学期不及格的必修课学分超过10个；
    - 不及格必修课学分累计超过30个；
    - 不及格选修课学分累计超过20个；

1.查询学生所选修的课程及成绩，并给出必修课平均成绩和选修课平均成绩；

2.查某一个学生被哪些教师教过课；

3.查询应被开除的学生（假定差2学分即被开除）。

信息要求

需要从数据库中获取 系、教学计划、学生、教师、课程的信息

数据要求：

- 系：系号、名称
- 班：班号、所在系
- 学生：学号、姓名、所在班、课程分数
- 课程：课程号、课程名、课程学分、课程性质
- 教师：姓名，教课班级、教课课程、所属系

处理要求

需要完成的处理功能：

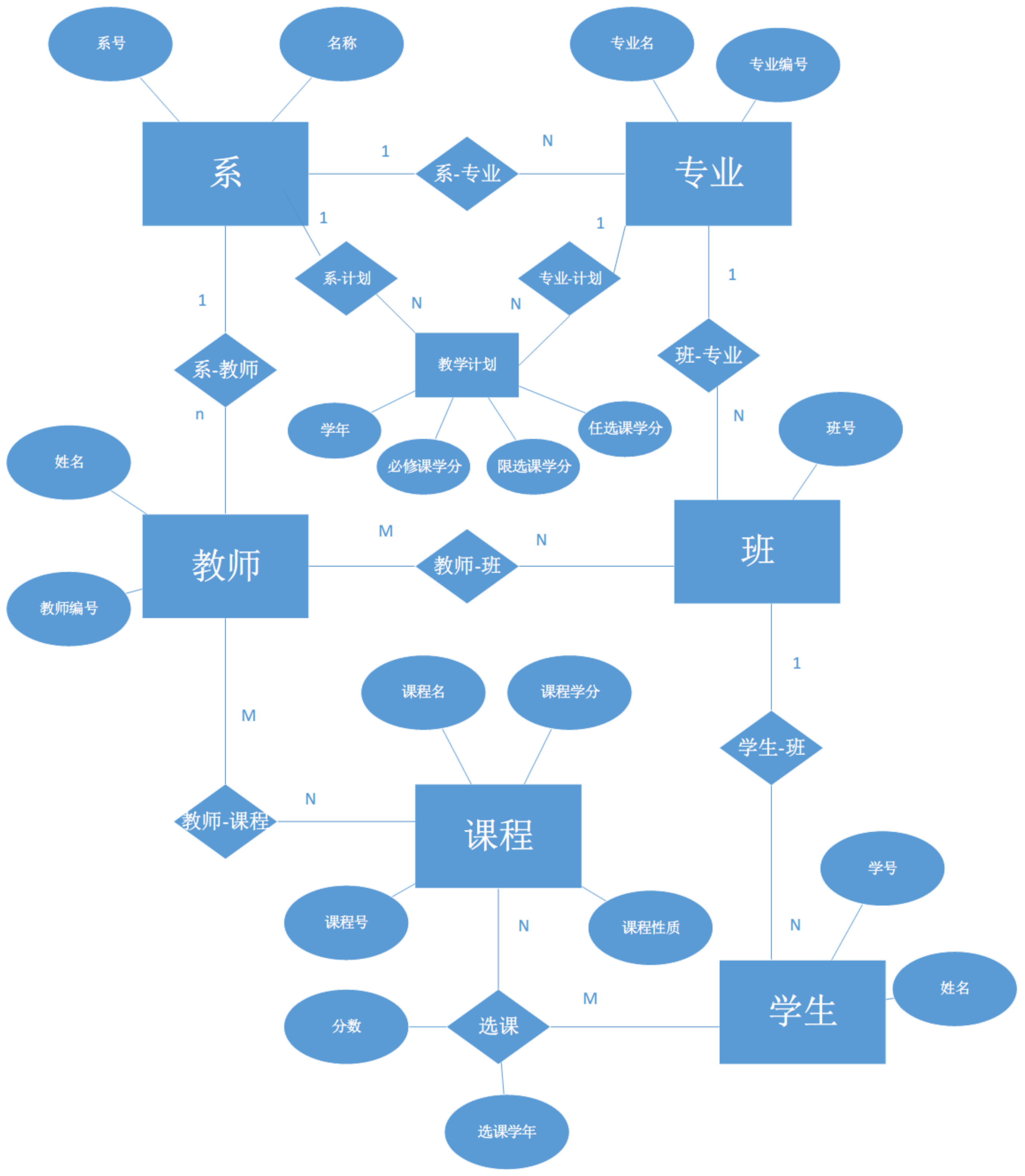
- 查询学生所选修的课程及成绩，并给出必修课平均成绩和选修课平均成绩；
- 查某一个学生被哪些教师教过课；
- 查询应被开除的学生（假定差2学分即被开除）

安全性和完整性要求

- 相关属性组的非空性和唯一性
- 系里的教师可以给多个班带课，但是不能给一个班带多门课程

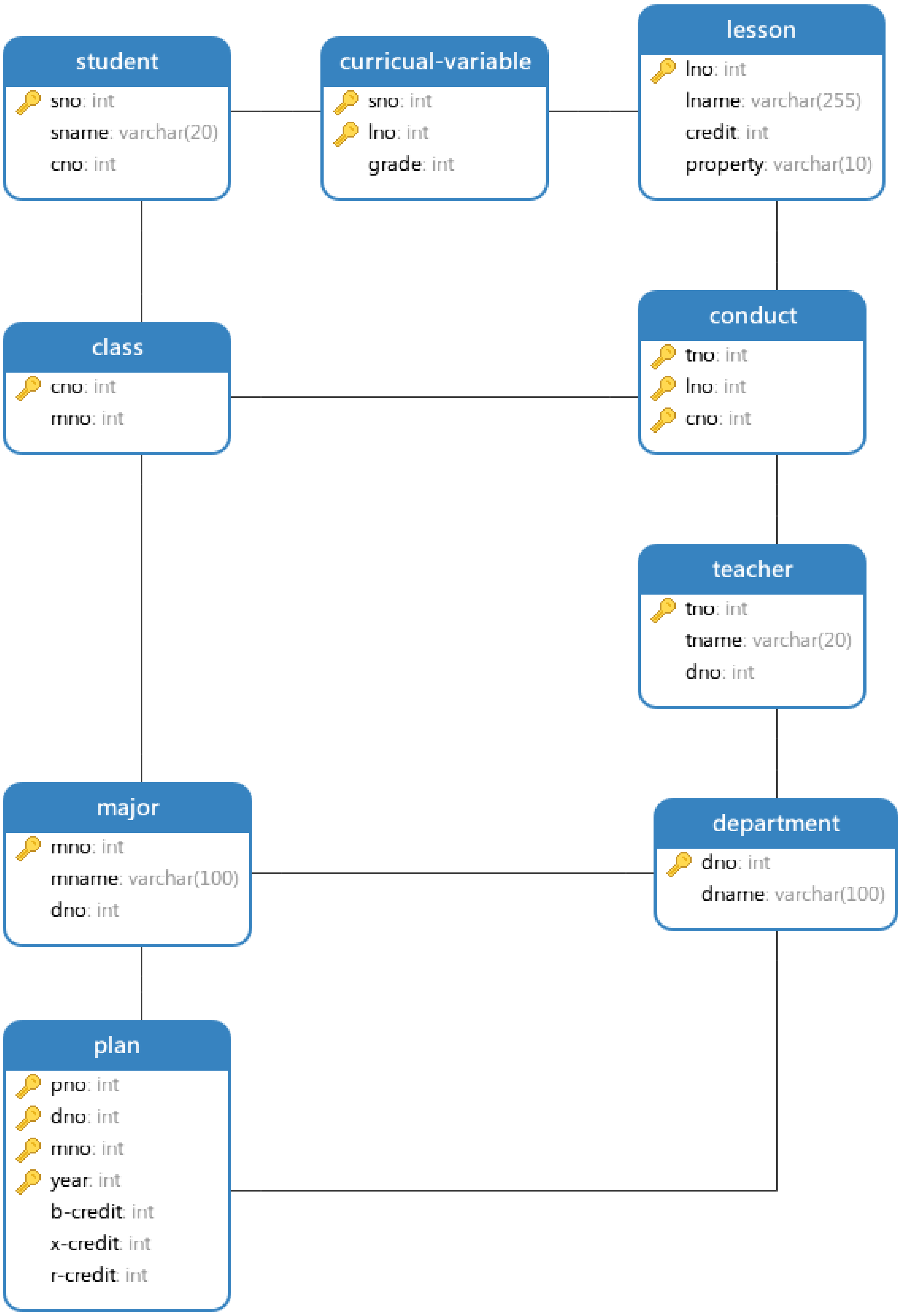
二、概念结构设计

根据需求，设计学籍管理系统 E-R 图如下：



三、逻辑结构设计

将 E-R 图转换为关系模式如下



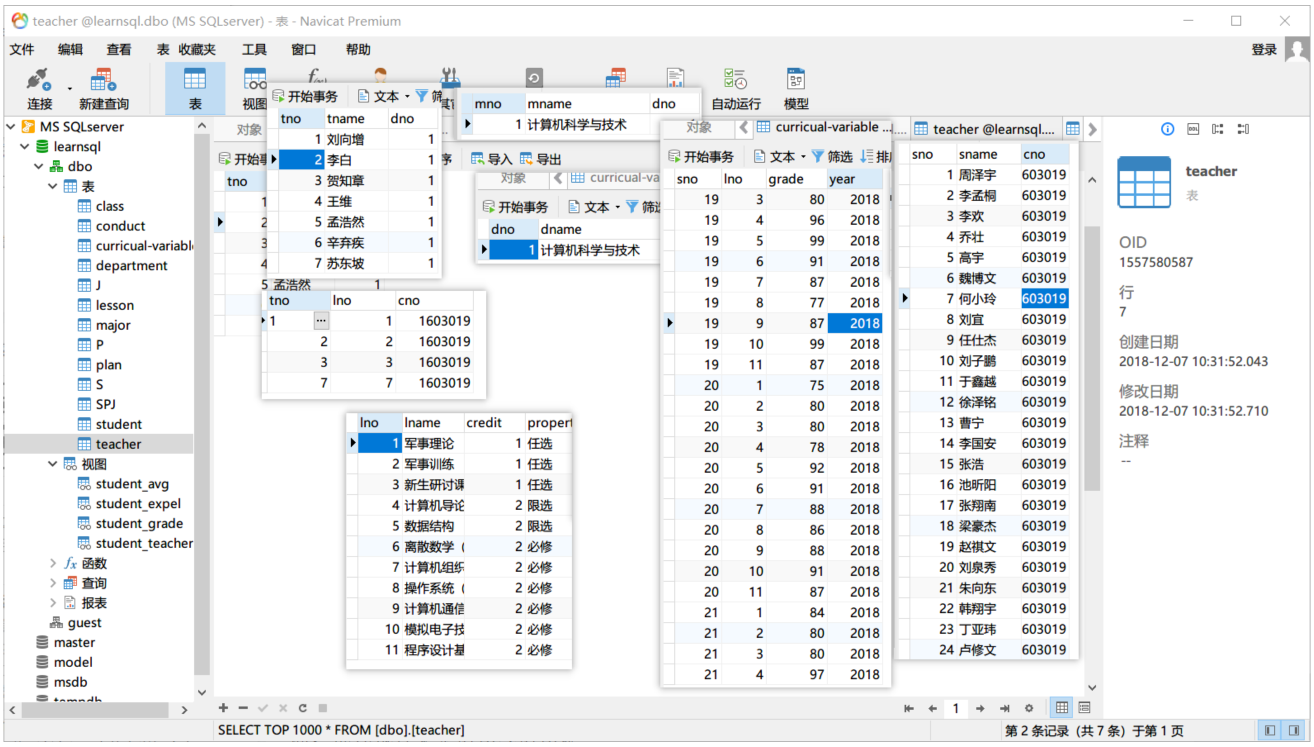
四、详细实现及源代码

根据建立的关系模式，编写建表 sql 语句

```sql
CREATE TABLE [department] (
[dno] int NOT NULL,
[dname] varchar(100) NOT NULL,
PRIMARY KEY ([dno])
)
GO
CREATE TABLE [major] (
[mno] int NOT NULL,
[mname] varchar(100) NOT NULL,
[dno] int NULL,
PRIMARY KEY ([mno])
)
GO
CREATE TABLE [class] (
[cno] int NOT NULL,
[mno] int NOT NULL,
PRIMARY KEY ([cno])
)
GO
CREATE TABLE [student] (
[sno] int NOT NULL,
[sname] varchar(20) NOT NULL,
[cno] int NOT NULL,
PRIMARY KEY ([sno])
)
GO
CREATE TABLE [teacher] (
[tno] int NOT NULL,
[tname] varchar(20) NOT NULL,
[dno] int NOT NULL,
PRIMARY KEY ([tno])
)
GO
CREATE TABLE [lesson] (
[lno] int NOT NULL,
[lname] varchar(255) NOT NULL,
[credit] int NOT NULL,
[property] varchar(10) NOT NULL,
PRIMARY KEY ([lno])
)
GO
CREATE TABLE [curricual-variable] (
[sno] int NOT NULL,
[lno] int NOT NULL,
[grade] int NOT NULL,
PRIMARY KEY ([sno], [lno])
)
GO
CREATE TABLE [plan] (
[pno] int NOT NULL,
[dno] int NOT NULL,
[mno] int NOT NULL,
[year] int NOT NULL,
[b-credit] int NOT NULL,
[x-credit] int NOT NULL,
[r-credit] int NOT NULL,
PRIMARY KEY ([pno], [dno], [mno], [year])
)
GO
CREATE TABLE [conduct] (
[tno] int NOT NULL,
[lno] int NOT NULL,
[cno] int NOT NULL,
PRIMARY KEY ([lno], [cno], [tno])
)
GO

ALTER TABLE [major] ADD CONSTRAINT [major-department] FOREIGN KEY ([dno]) REFERENCES [department] ([dno])
GO
ALTER TABLE [class] ADD CONSTRAINT [class-major] FOREIGN KEY ([mno]) REFERENCES [major] ([mno])
GO
ALTER TABLE [student] ADD CONSTRAINT [student-class] FOREIGN KEY ([cno]) REFERENCES [class] ([cno])
GO
ALTER TABLE [teacher] ADD CONSTRAINT [teacher-department] FOREIGN KEY ([dno]) REFERENCES [department] ([dno])
GO
ALTER TABLE [curricual-variable] ADD CONSTRAINT [cur-stu] FOREIGN KEY ([sno]) REFERENCES [student] ([sno])
GO
ALTER TABLE [curricual-variable] ADD CONSTRAINT [cur-les] FOREIGN KEY ([lno]) REFERENCES [lesson] ([lno])
GO
ALTER TABLE [plan] ADD CONSTRAINT [plan-dep] FOREIGN KEY ([dno]) REFERENCES [department] ([dno])
GO
ALTER TABLE [plan] ADD CONSTRAINT [plan-maj] FOREIGN KEY ([mno]) REFERENCES [major] ([mno])
GO
ALTER TABLE [conduct] ADD CONSTRAINT [cond-teacher] FOREIGN KEY ([tno]) REFERENCES [teacher] ([tno])
GO
ALTER TABLE [conduct] ADD CONSTRAINT [cond-class] FOREIGN KEY ([cno]) REFERENCES [class] ([cno])
GO
ALTER TABLE [conduct] ADD CONSTRAINT [cond-lesson] FOREIGN KEY ([lno]) REFERENCES [lesson] ([lno])
GO
```

执行建表语句，并编写测试数据：


teacher @learnsql.dbo (MS SQLserver) - 表 - Navicat Premium

tno	tname	dno
1	刘向增	1
2	李白	1
3	贺知章	1
4	王维	1
5	孟浩然	1
6	辛弃疾	1
7	苏东坡	1

mno	mname	dno
1	计算机科学与技术	

dno	dname
1	计算机科学与技术

tno	lno	cno
1	1	1603019
2	2	1603019
3	3	1603019
7	7	1603019

lno	lname	credit	propert
1	军事理论	1	任选
2	军事训练	1	任选
3	新生研讨课	1	任选
4	计算机导论	2	限选
5	数据结构	2	限选
6	离散数学（	2	必修
7	计算机组织	2	必修
8	操作系统（	2	必修
9	计算机通信	2	必修
10	模拟电子技	2	必修
11	程序设计基	2	必修

curricual-variable ...

sno	lno	grade	year
19	3	80	2018
19	4	96	2018
19	5	99	2018
19	6	91	2018
19	7	87	2018
19	8	77	2018
19	9	87	2018
19	10	99	2018
19	11	87	2018
20	1	75	2018
20	2	80	2018
20	3	80	2018
20	4	78	2018
20	5	92	2018
20	6	91	2018
20	7	88	2018
20	8	86	2018
20	9	88	2018
20	10	91	2018
20	11	87	2018
21	1	84	2018
21	2	80	2018
21	3	80	2018
21	4	97	2018

teacher @learnsql....

sno	sname	cno
1	周泽宇	603019
2	李孟桐	603019
3	李欢	603019
4	乔壮	603019
5	高宇	603019
6	魏博文	603019
7	何小玲	603019
8	刘宜	603019
9	任仕杰	603019
10	刘子鹏	603019
11	于鑫越	603019
12	徐泽铭	603019
13	曹宁	603019
14	李国安	603019
15	张浩	603019
16	池昕阳	603019
17	张翔南	603019
18	梁豪杰	603019
19	赵祺文	603019
20	刘泉秀	603019
21	朱向东	603019
22	韩翔宇	603019
23	丁亚玮	603019
24	卢修文	603019

teacher
表
OID
1557580587
行
7
创建日期
2018-12-07 10:31:52.043
修改日期
2018-12-07 10:31:52.710
注释
--

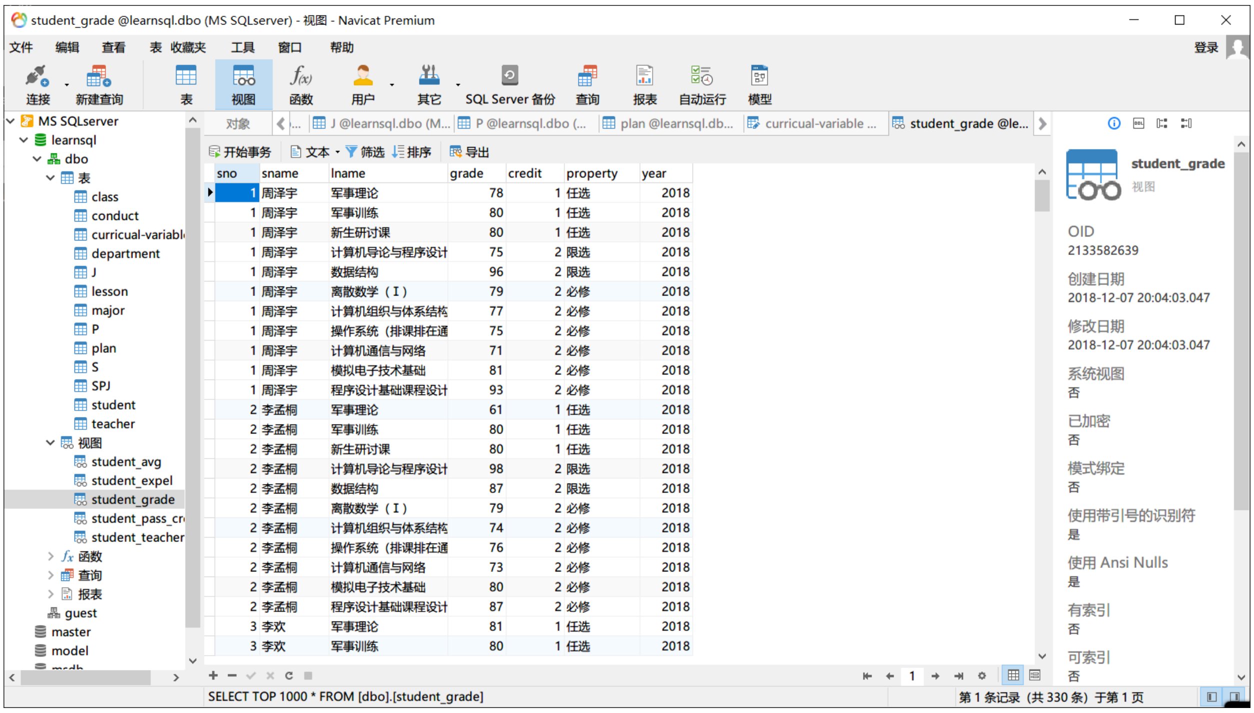
SELECT TOP 1000 * FROM [dbo].[teacher]
第 2 条记录（共 7 条）于第 1 页

可以看到，对应的基本表已经建立，并可以通过基本表建立数据

视图的建立

- 建立 `student_grade` 视图，以查询学生的各科成绩

```sql
CREATE VIEW student_grade
AS
SELECT student.sno, sname,lname,grade,credit,property,[curricual-variable].year
FROM student,[curricual-variable],lesson
WHERE student.sno = [curricual-variable].sno AND
        [curricual-variable].[lno] = lesson.lno
GO
```



sno	sname	lname	grade	credit	property	year
1	周泽宇	军事理论	78	1	任选	2018
1	周泽宇	军事训练	80	1	任选	2018
1	周泽宇	新生研讨课	80	1	任选	2018
1	周泽宇	计算机导论与程序设计	75	2	限选	2018
1	周泽宇	数据结构	96	2	限选	2018
1	周泽宇	离散数学（I）	79	2	必修	2018
1	周泽宇	计算机组织与体系结构	77	2	必修	2018
1	周泽宇	操作系统（排课排在通	75	2	必修	2018
1	周泽宇	计算机通信与网络	71	2	必修	2018
1	周泽宇	模拟电子技术基础	81	2	必修	2018
1	周泽宇	程序设计基础课程设计	93	2	必修	2018
2	李孟桐	军事理论	61	1	任选	2018
2	李孟桐	军事训练	80	1	任选	2018
2	李孟桐	新生研讨课	80	1	任选	2018
2	李孟桐	计算机导论与程序设计	98	2	限选	2018
2	李孟桐	数据结构	87	2	限选	2018
2	李孟桐	离散数学（I）	79	2	必修	2018
2	李孟桐	计算机组织与体系结构	74	2	必修	2018
2	李孟桐	操作系统（排课排在通	76	2	必修	2018
2	李孟桐	计算机通信与网络	73	2	必修	2018
2	李孟桐	模拟电子技术基础	80	2	必修	2018
2	李孟桐	程序设计基础课程设计	87	2	必修	2018
3	李欢	军事理论	81	1	任选	2018
3	李欢	军事训练	80	1	任选	2018

student_grade 视图

OID 2133582639
创建日期 2018-12-07 20:04:03.047
修改日期 2018-12-07 20:04:03.047
系统视图 否
已加密 否
模式绑定 否
使用带引号的识别符 是
使用 Ansi Nulls 是
有索引 否
可索引 否

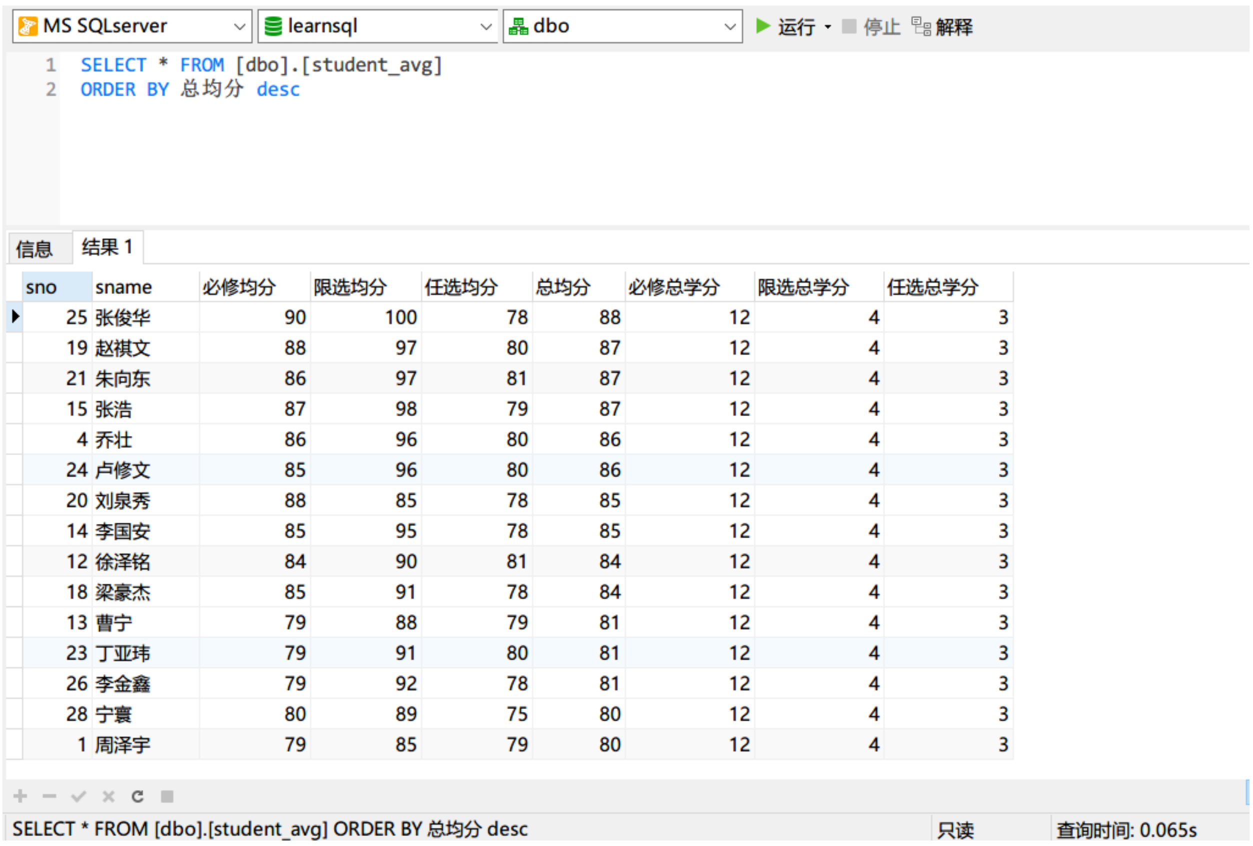
```
SELECT TOP 1000 * FROM [dbo].[student_grade]
```

第 1 条记录（共 330 条）于第 1 页

- 建立 `student_avg` 视图，查询学生的平均成绩

```
create view student_avg
AS
SELECT distinct stu.sno,stu.sname,必修均分,限选均分,任选均分,总均分,必修总学分,限选总学分,任选总学分
FROM student_grade as stu
INNER JOIN (
    SELECT sno,sname,AVG(grade) as '必修均分',SUM(credit) as '必修总学分'
    FROM student_grade
    WHERE property = '必修'
    group by sno,sname
) as b
on stu.sno = b.sno
INNER JOIN (
    SELECT sno,sname,AVG(grade) as '限选均分',SUM(credit) as '限选总学分'
    FROM student_grade
    WHERE property = '限选'
    group by sno,sname
)as x
on stu.sno = x.sno
INNER JOIN (
    SELECT sno,sname,AVG(grade) as '任选均分',SUM(credit) as '任选总学分'
    FROM student_grade
    WHERE property = '任选'
    group by sno,sname
)as r
on stu.sno = r.sno
INNER JOIN (
    SELECT sno,sname,AVG(grade) as '总均分',SUM(credit) as '总学分'
    FROM student_grade
    group by sno,sname
)as z
```

```sql
on stu.sno = z.sno
go
```



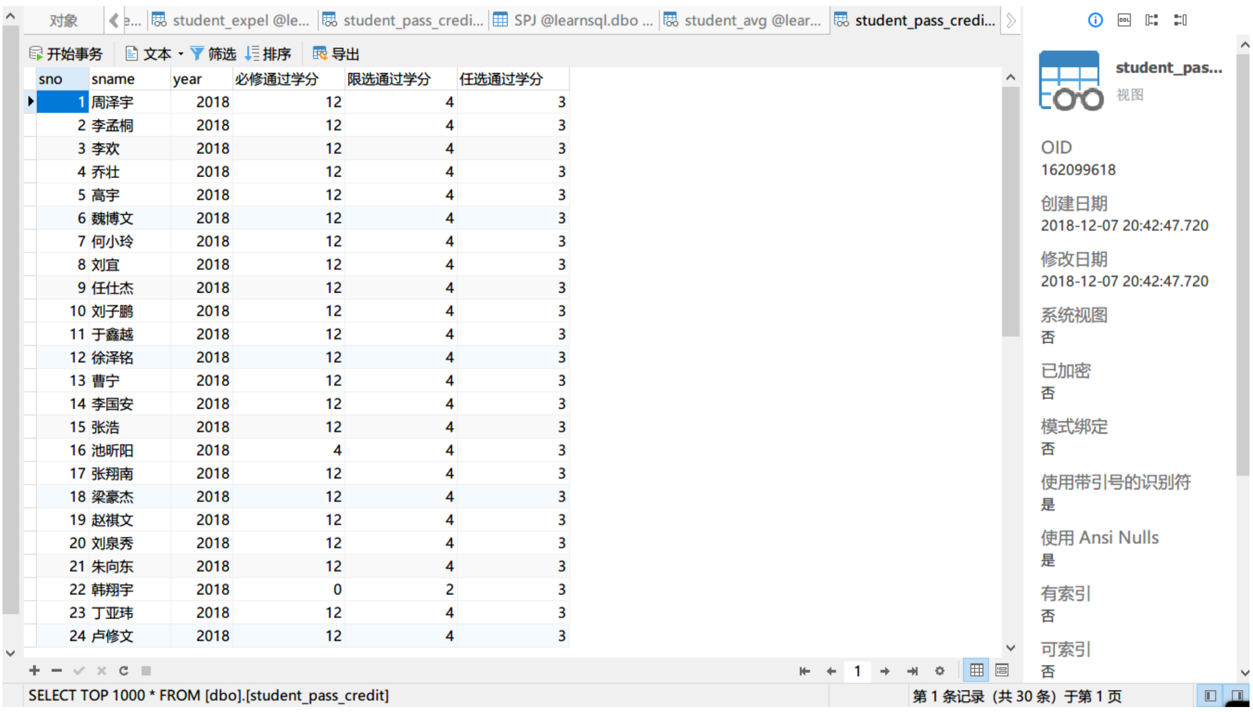
```sql
SELECT * FROM [dbo].[student_avg]
ORDER BY 总均分 desc
```

sno	sname	必修均分	限选均分	任选均分	总均分	必修总学分	限选总学分	任选总学分
25	张俊华	90	100	78	88	12	4	3
19	赵祺文	88	97	80	87	12	4	3
21	朱向东	86	97	81	87	12	4	3
15	张浩	87	98	79	87	12	4	3
4	乔壮	86	96	80	86	12	4	3
24	卢修文	85	96	80	86	12	4	3
20	刘泉秀	88	85	78	85	12	4	3
14	李国安	85	95	78	85	12	4	3
12	徐泽铭	84	90	81	84	12	4	3
18	梁豪杰	85	91	78	84	12	4	3
13	曹宁	79	88	79	81	12	4	3
23	丁亚玮	79	91	80	81	12	4	3
26	李金鑫	79	92	78	81	12	4	3
28	宁寰	80	89	75	80	12	4	3
1	周泽宇	79	85	79	80	12	4	3

SELECT * FROM [dbo].[student_avg] ORDER BY 总均分 desc　只读　查询时间: 0.065s

- 建立 `student_pass_credit` 视图，查询学生通过的学分数 （成绩>=60分视为通过）

```sql
create view student_pass_credit
AS
SELECT distinct stu.sno,stu.sname,stu.year,isNULL(bx,0) as 必修通过学分,isNULL(xx,0) as 限选通过学分,isNULL(rx,0) as 任选通过学分
FROM student_grade as stu
left JOIN (
    SELECT sno,sname,student_grade.year,SUM(credit) as bx
    FROM student_grade
    WHERE property = '必修' and grade >=60
    group by sno,sname,student_grade.year
) as b
on stu.sno = b.sno AND stu.year = b.year
left JOIN (
    SELECT sno,sname,student_grade.year,SUM(credit) as xx
    FROM student_grade
    WHERE property = '限选' and grade >=60
    group by sno,sname,student_grade.year
)as x
on stu.sno = x.sno AND stu.year = x.year
left JOIN (
    SELECT sno,sname,student_grade.year,SUM(credit) as rx
    FROM student_grade
    WHERE property = '任选' and grade >=60
    group by sno,sname,student_grade.year
)as r
on stu.sno = r.sno AND stu.year = r.year
left JOIN (
    SELECT sno,sname,student_grade.year,SUM(credit) as z
    FROM student_grade
    where  grade >=60
    group by sno,sname,student_grade.year
)AS z
on stu.sno = z.sno AND stu.year = z.year
go
```



sno	sname	year	必修通过学分	限选通过学分	任选通过学分
1	周泽宇	2018	12	4	3
2	李孟桐	2018	12	4	3
3	李欢	2018	12	4	3
4	乔壮	2018	12	4	3
5	高宇	2018	12	4	3
6	魏博文	2018	12	4	3
7	何小玲	2018	12	4	3
8	刘宜	2018	12	4	3
9	任仕杰	2018	12	4	3
10	刘子鹏	2018	12	4	3
11	于鑫越	2018	12	4	3
12	徐泽铭	2018	12	4	3
13	曹宁	2018	12	4	3
14	李国安	2018	12	4	3
15	张浩	2018	12	4	3
16	池昕阳	2018	4	4	3
17	张翔南	2018	12	4	3
18	梁豪杰	2018	12	4	3
19	赵祺文	2018	12	4	3
20	刘泉秀	2018	12	4	3
21	朱向东	2018	12	4	3
22	韩翔宇	2018	0	2	3
23	丁亚玮	2018	12	4	3
24	卢修文	2018	12	4	3

SELECT TOP 1000 * FROM [dbo].[student_pass_credit]

第 1 条记录（共 30 条）于第 1 页

student_pas...
视图

OID
162099618

创建日期
2018-12-07 20:42:47.720

修改日期
2018-12-07 20:42:47.720

系统视图
否

已加密
否

模式绑定
否

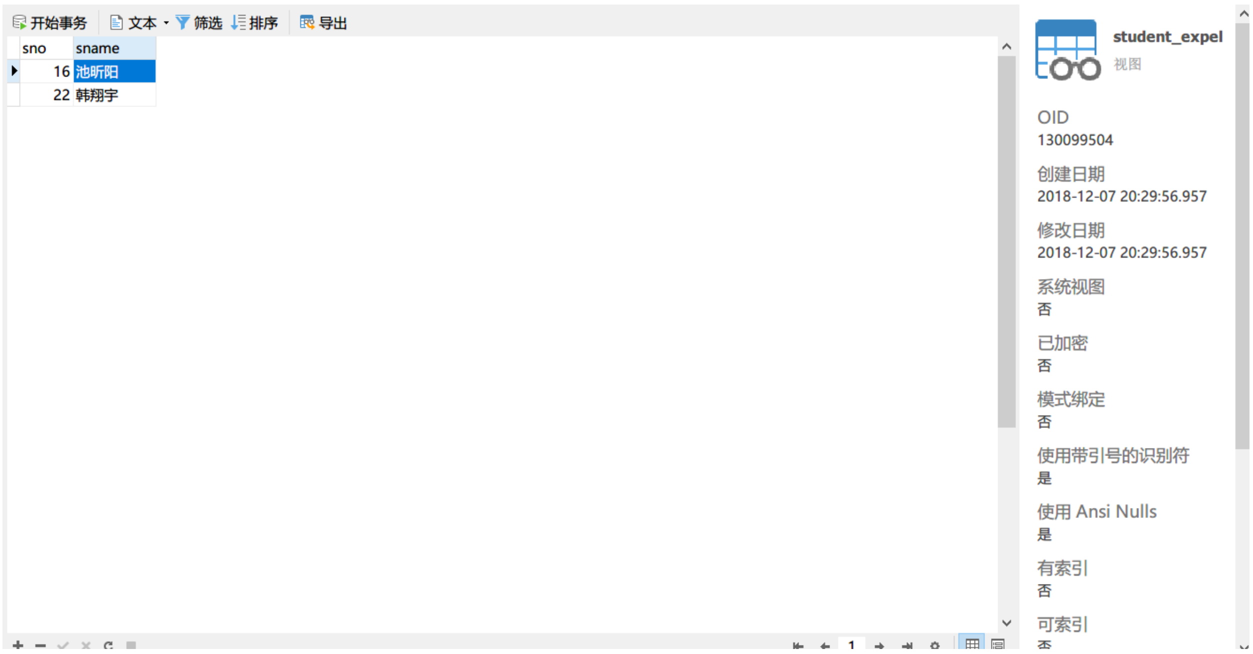
使用带引号的识别符
是

使用 Ansi Nulls
是

有索引
否

可索引
否

- 建立 `student_expel` 视图，筛选需要开除的学生



sno	sname
16	池昕阳
22	韩翔宇

student_expel
视图

OID
130099504

创建日期
2018-12-07 20:29:56.957

修改日期
2018-12-07 20:29:56.957

系统视图
否

已加密
否

模式绑定
否

使用带引号的识别符
是

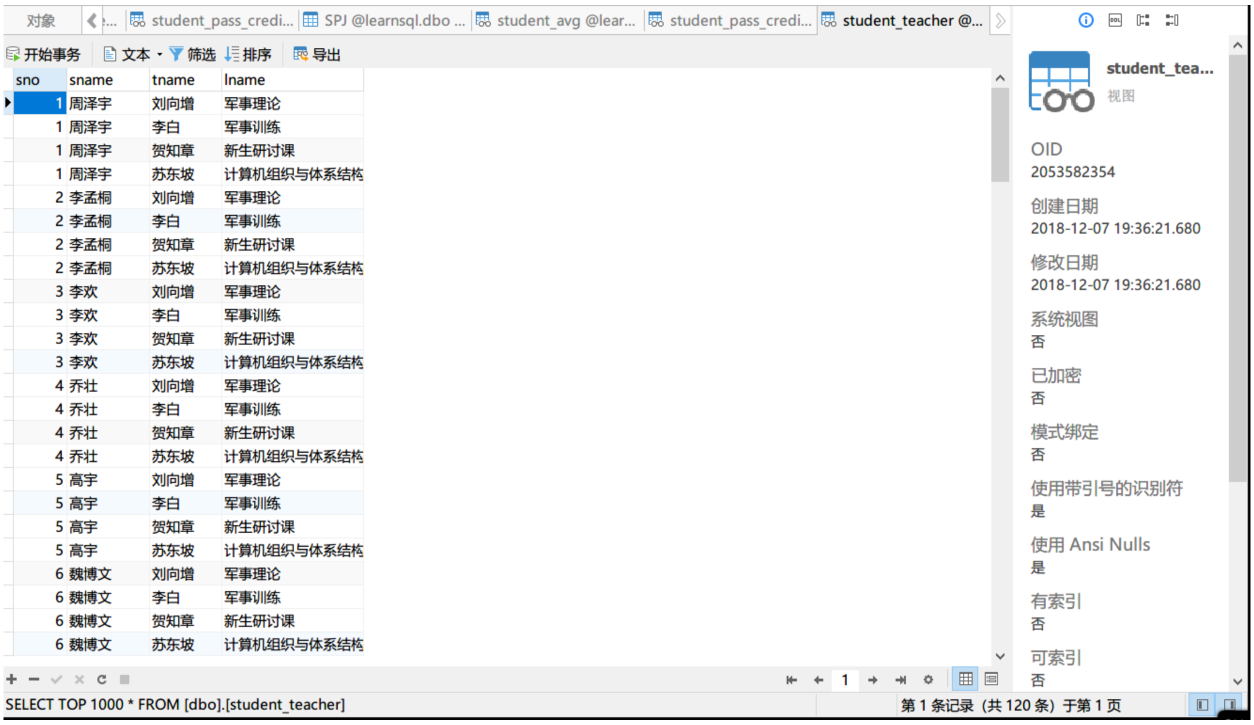
使用 Ansi Nulls
是

有索引
否

可索引
否

- 建立 student_teacher 视图，查询学生的任课老师和所教课程

```sql
CREATE view student_teacher
AS
SELECT student.sno,sname,tname,lname
FROM student,teacher,conduct,[curricual-variable],lesson
WHERE student.cno = conduct.cno AND
        teacher.tno = conduct.tno and
        conduct.lno = [curricual-variable].lno and
        student.sno = [curricual-variable].sno AND
        lesson.lno = conduct.lno
GO
```



sno	sname	tname	lname
1	周泽宇	刘向增	军事理论
1	周泽宇	李白	军事训练
1	周泽宇	贺知章	新生研讨课
1	周泽宇	苏东坡	计算机组织与体系结构
2	李孟桐	刘向增	军事理论
2	李孟桐	李白	军事训练
2	李孟桐	贺知章	新生研讨课
2	李孟桐	苏东坡	计算机组织与体系结构
3	李欢	刘向增	军事理论
3	李欢	李白	军事训练
3	李欢	贺知章	新生研讨课
3	李欢	苏东坡	计算机组织与体系结构
4	乔壮	刘向增	军事理论
4	乔壮	李白	军事训练
4	乔壮	贺知章	新生研讨课
4	乔壮	苏东坡	计算机组织与体系结构
5	高宇	刘向增	军事理论
5	高宇	李白	军事训练
5	高宇	贺知章	新生研讨课
5	高宇	苏东坡	计算机组织与体系结构
6	魏博文	刘向增	军事理论
6	魏博文	李白	军事训练
6	魏博文	贺知章	新生研讨课
6	魏博文	苏东坡	计算机组织与体系结构

student_tea...
视图

OID
2053582354
创建日期
2018-12-07 19:36:21.680
修改日期
2018-12-07 19:36:21.680
系统视图
否
已加密
否
模式绑定
否
使用带引号的识别符
是
使用 Ansi Nulls
是
有索引
否
可索引
否

SELECT TOP 1000 * FROM [dbo].[student_teacher]

第 1 条记录（共 120 条）于第 1 页

建立触发器，实现数据完整性约束

建立 `conduct_instead` 触发器，在插入任课信息前进行完整性检查，即：

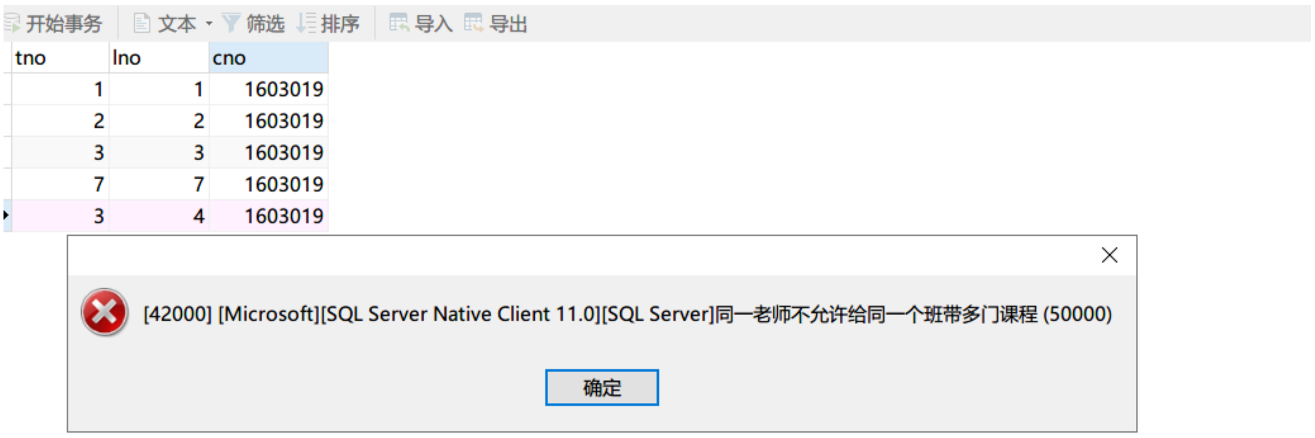
- 系里的教师可以给多个班带课，但是不能给一个班带多门课程

```sql
CREATE TRIGGER conduct_instead
on conduct
INSTEAD of update,insert
as
IF exists (
    SELECT * FROM conduct,inserted
    where conduct.cno = inserted.cno AND
          conduct.tno = inserted.tno AND
          not exists(
            SELECT * FROM inserted,deleted
            WHERE inserted.tno = deleted.tno
          )
)
    RAISERROR('同一老师不允许给同一个班带多门课程',16,10)
ELSE
```

```
BEGIN
    DELETE FROM conduct
    WHERE  exists(
        select * from deleted
        WHERE conduct.tno = deleted.tno AND
                conduct.cno = deleted.cno AND
                conduct.lno = deleted.lno
    )

    INSERT into conduct
    SELECT * FROM inserted
end
GO
```

定义触发器后，当插入或修改的数据不满足要求时，就会弹出错误警告，并取消插入或修改操作



tno	lno	cno
1	1	1603019
2	2	1603019
3	3	1603019
7	7	1603019
3	4	1603019

```
[42000] [Microsoft][SQL Server Native Client 11.0][SQL Server]同一老师不允许给同一个班带多门课程 (50000)
```