**北京科技大学实验报告**

学院：计算机与通信工程学院 专业： 通信工程 班级： 通信1701

姓名： 胡成成 学号： 41724260 实验日期： 2019年 5月 20 日

**实验名称：数据库系统开发——学生管理信息系统（查询部分）**

**实验目的：**

熟练掌握Transact-SQL语言，熟练掌握数据查询中的分组、统计、计算和组合的操作方法。掌握用Transact-SQL命令创建视图、使用视图和删除视图的方法。

1、掌握SELECT语句的基本语法和查询条件表示方法。

2、掌握连接查询的表示。

3、掌握子查询和嵌套查询的表示。

4、掌握 SELECT 语句的统计函数（AVG、SUM、MAX、MIN、COUNT）的使用方法。

5、掌握 SELECT 语句的 GROUPBY 和 ORDERBY 子句的作用和使用方法。

6、熟悉视图的概念和作用，掌握视图的创建、查询和修改方法。

**实验环境：**

操作系统：Win7-1

数据库：SQL Server 2008

**实验内容：**

* 基本背景

建立一个学生管理信息系统，主要包括学生可以 多个课程，学生的每个课程有相应的成绩；

学生有学号，姓名等属性，课程有课程号与课程名等属性等。构建好数据库后开始进行相关查询测试。

* 基本功能

1、使用Transact-SQL语言实现复杂查询。

2、使用Transact-SQL语言定义视图。

* 数据库结构

表1 学生信息表（Student）的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段说明 | 字段名称 | 数据类型 | 必填项 | 主键 | 备注 |
| 1 | 学号 | StuNo | char(8) | Y | Y |  |
| 2 | 姓名 | StuName | nvarchar(64) | Y |  |  |
| 3 | 性别 | Sex | nvarchar(4) | Y |  |  |
| 4 | 出身日期 | Birthday | date |  |  |  |
| 5 | 专业编号 | MajorNo | nvarchar(4) |  |  |  |
| 6 | 籍贯 | Address | nvarchar(256) |  |  |  |
| 7 | 入学时间 | EnTime | date |  |  |  |

表2 课程表（Course）的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段说明 | 字段名称 | 数据类型 | 必填项 | 主键 | 备注 |
| 1 | 课程号 | CNo | char(6) | Y | Y |  |
| 2 | 课程名 | CName | nvarchar(64) | Y |  |  |
| 3 | 学分 | Credit | int |  |  | 缺省值2 |
| 4 | 学时数 | ClassHour | int |  |  | 缺省值32 |

表3 学习成绩表（SC）的结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段说明 | 字段名称 | 数据类型 | 必填项 | 主键 | 备注 |
| 1 | 学号 | StuNo | char(8) | Y | Y |  |
| 2 | 课程号 | CNo | char(6) | Y | Y |  |
| 3 | 成绩 | Score | decimal(18, 2) |  |  |  |

**实验步骤：**

1、在学生管理信息系统数据库StudentMIS中，根据自己在实验2的数据库里增加的数据，使用SQL server 2008中的“查询分析器”输入Transact-SQL查询语句，实现以下数据查询操作：

（1）查询选修了某一课程（例如：数据库技术）的学生学号和姓名；

select Student.StuNo,StuName from Student,SC,Course where Student.StuNo = SC.StuNo and Course.CNo = SC.CNo and CName='SX'

（2）查询某一课程的成绩高于某个学生（例如：张三）的学生的学号和成绩；

select Student.StuNo,Score from Student,SC where Student.StuNo = SC.StuNo and CNo='000001' and Score>(select Score from Student,SC where Student.StuNo = SC.StuNo and StuName='LZH')

（3）查询某一专业中比另外一个专业某一学生年龄小的学生；

select StuName from Student where MajorNo='TX' and YEAR(Birthday)>YEAR((select Birthday from Student where StuName='ZTM' and MajorNo ='J'))

（4）查询没有选修某一课程的学生姓名；

select Distinct StuName from Student ,SC,Course where Student.StuNo = SC.StuNo and Course.CNo = SC.CNo and Course.CNo ! = '000001'

（5）查询所有被学生选修了的课程号；

select distinct CNo from SC where StuNo is not null

（6）查询选修某一课程的人数。

select COUNT(StuNo) as 'number' from SC,Course where Course.CNo = SC.CNo and CName='SX'

（7）查询某一专业女学生的姓名、出生日期以及籍贯；

select StuName,Birthday,Address from Student where MajorNo='TX'

（8）查询所有姓李的学生的个人信息；

select \* from Student where StuName Like 'L%'

（9）查询课程名为“数据库技术”的平均成绩、最高分、最低分；

select AVG(Score),MAX(Score),MIN(Score) from SC,Course where Course.CNo = SC.CNo group by CName having CName='SX'

（10）查询成绩为空的学生姓名；

select StuName from Student,SC where Student.StuNo = SC.StuNo and CNo is null

（11）查询所有与学生“张三”有相同选修课程的学生信息；

select \* from Student,SC where Student.StuNo = SC.StuNo and CNo in (select CNo from Student,SC where Student.StuNo = SC.StuNo and StuName='HCC')

（12）查询年龄介于18岁~22岁之间的学生信息；

select \* from Student where Year(GETDATE())-YEAR(Birthday) between 18 and 22

（13）查询选修了某一课程的学生学号及其成绩，并按成绩降序排列；

select SC.StuNo,Score from SC,Course where Course.CNo = SC.CNo and CName='SX' order by Score Desc

（14）查询全体学生信息，要求查询结构按专业号升序排列，同一专业学生按年龄降序排列；

select \* from Student order by MajorNo ,year(getDate())-year(Birthday) Desc

（15）查询选修了3门及以上课程的学生的学号和姓名；

select SC.StuNo,StuName from SC,Student where Student.StuNo = SC.StuNo group by SC.StuNo,StuName having COUNT(CNo)>2

（16）统计每个学生选修课程的门数；

select StuNo,COUNT(CNo) from SC group by StuNo

（17）查询年龄大于男学生平均年龄的女学生姓名和年龄。

select StuName,(year(getDate())-year(Birthday)) as Age from Student where Sex='F' and year(getDate())-year(Birthday)>(select AVG(year(getDate())-year(Birthday)) from Student where Sex='M')

2、在学生管理信息系统数据库StudentMIS中，使用SQL server 2008中的“查询分析器”的Transact-SQL命令定义如下视图：

（1）建立某一专业（例如：通信工程专业）的学生视图；

Create View Student\_TX as select StuNo,StuName,Sex, Birthday, MajorNo, Address, EnTime From Student where MajorNo=’TX’

（2）由学生、课程和成绩三个表，定义某一专业（例如：通信工程专业）的学生成绩视图，其属性包括学号、姓名、课程名、课程名和成绩；

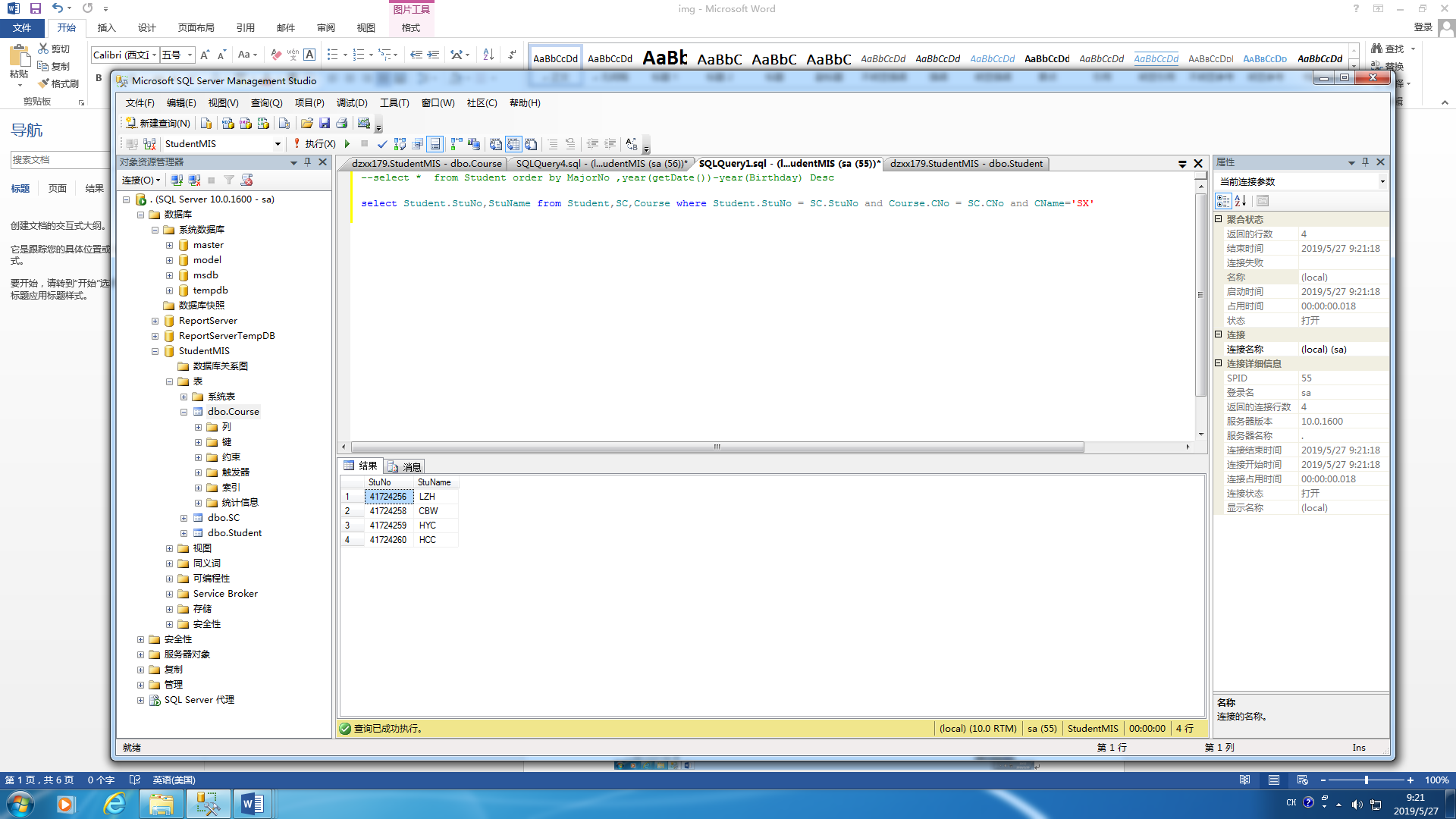
Create View SC\_Course(学号，姓名，课程名，课程号，成绩) as select Student.StuNo,StuName,CNo,SC.CName,Score From Student,Course,SC where MajorNo=’TX’

（3）查看以上定义的2个视图，并删除该视图。

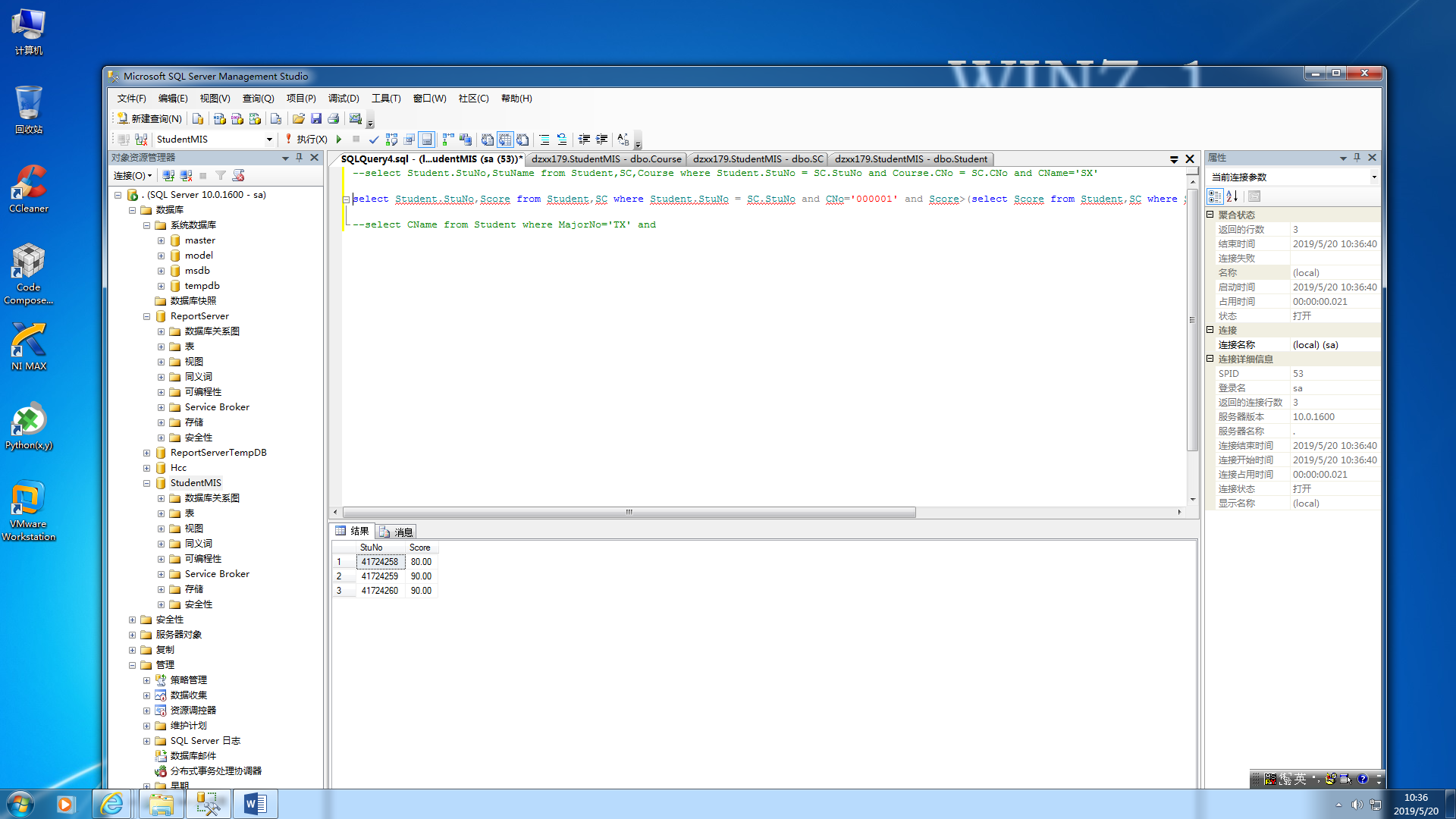
Drop VIEW Student\_TX, SC\_Course

**实验结果与分析：**

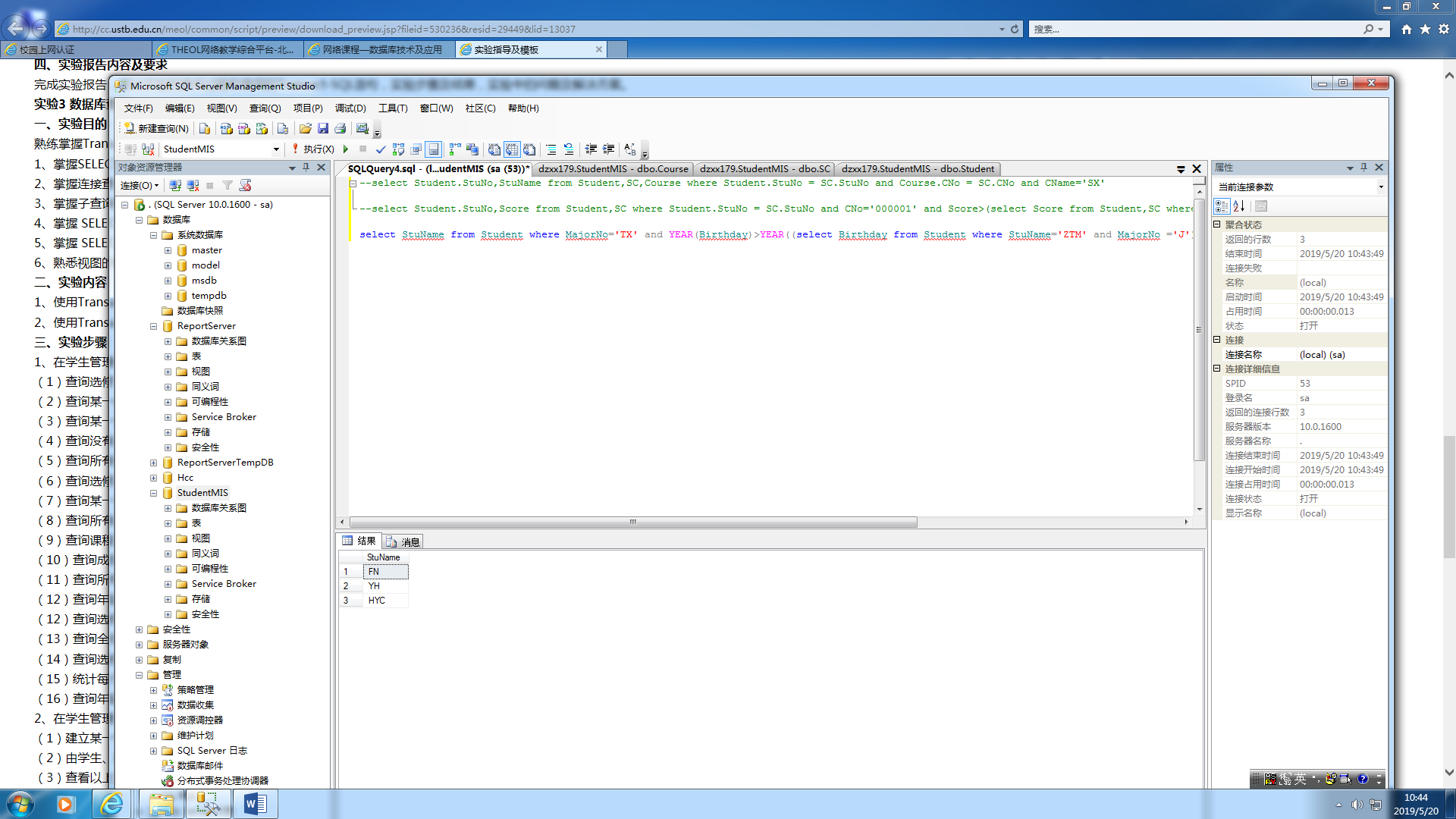
1. 查询结果：
2. 查询选修了某一课程（例如：数据库技术）的学生学号和姓名；



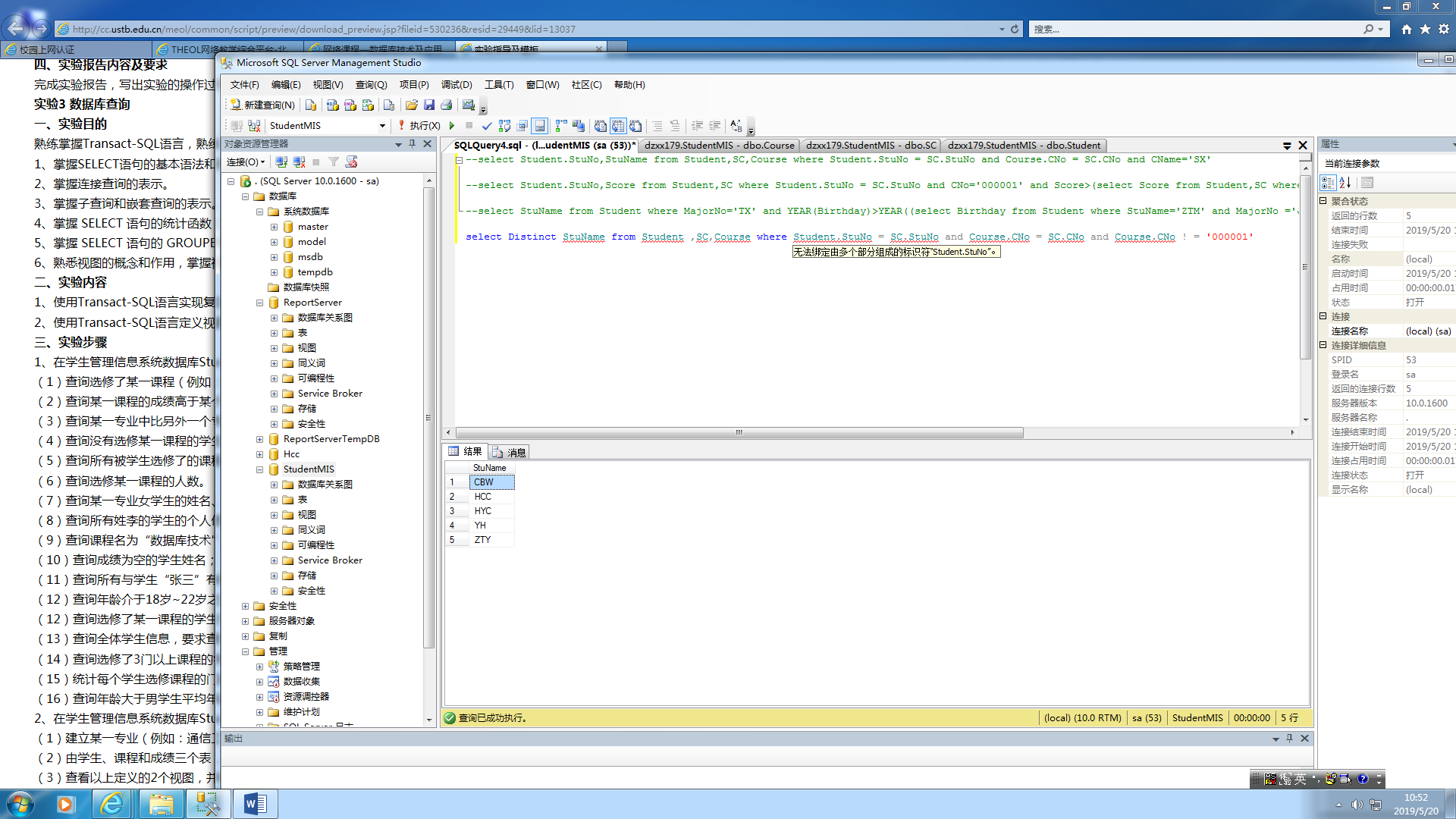
1. 查询某一课程的成绩高于某个学生（例如：张三）的学生的学号和成绩；



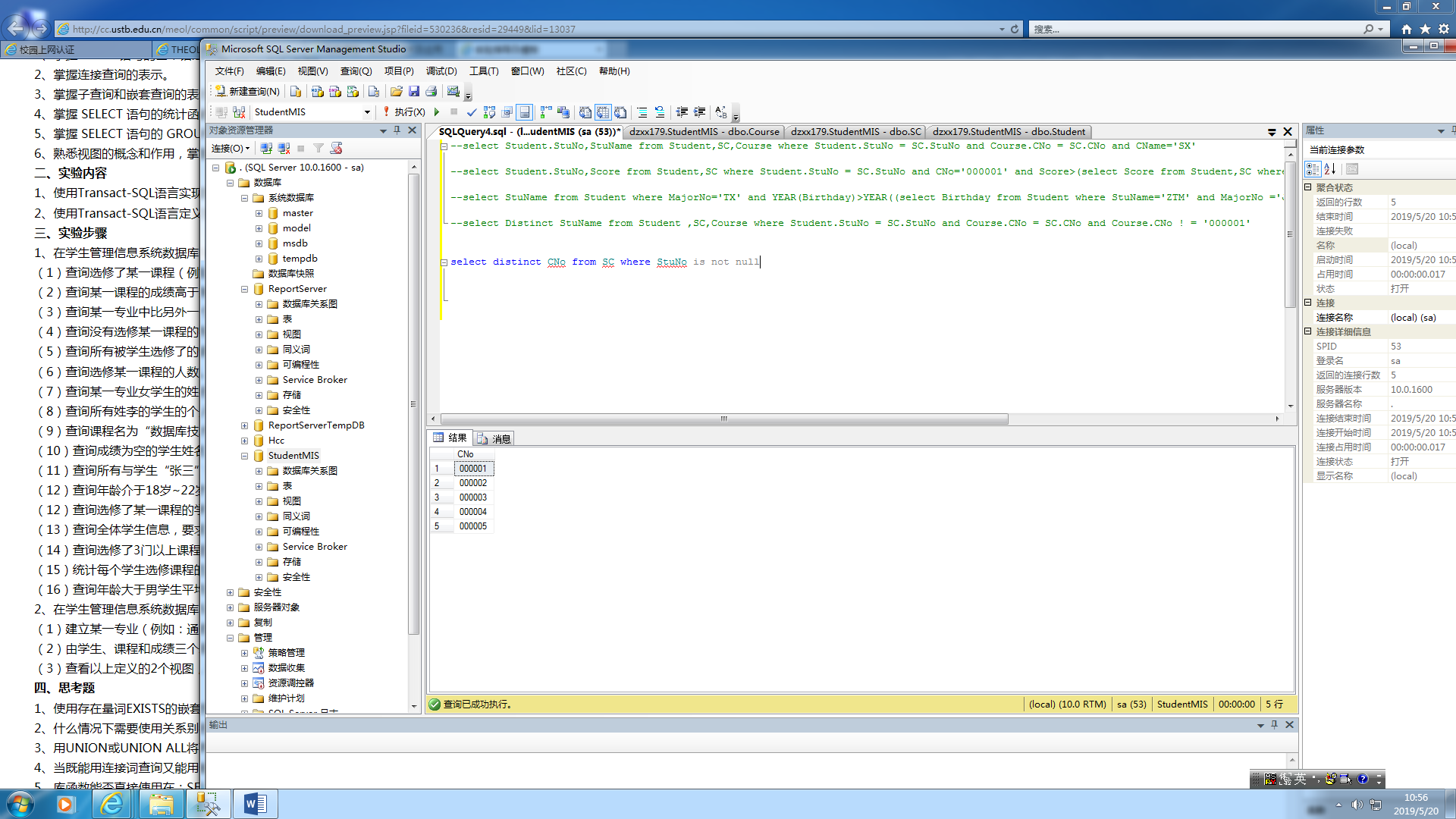
1. 查询某一专业中比另外一个专业某一学生年龄小的学生；



1. 查询没有选修某一课程的学生姓名；



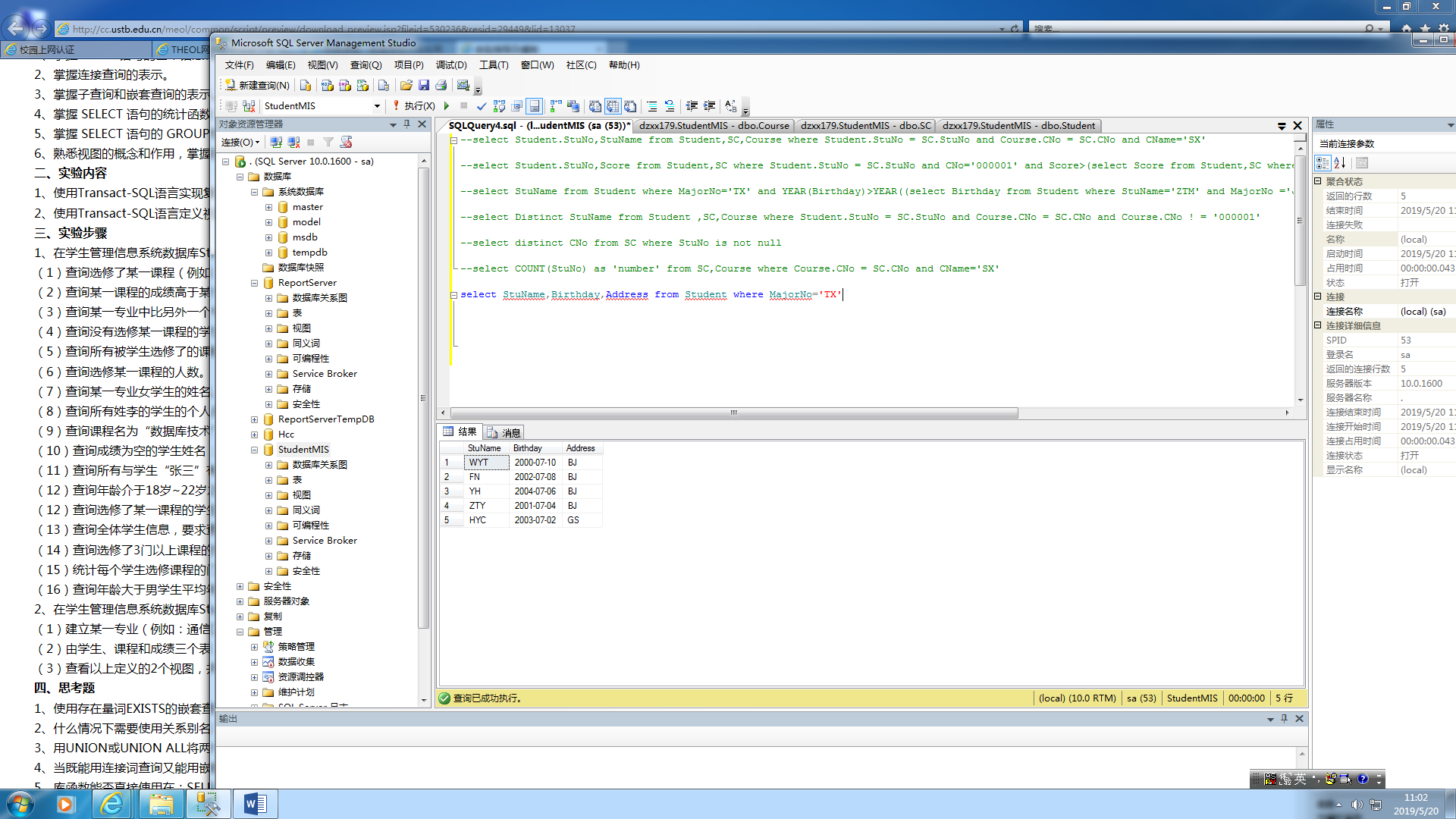
1. 查询所有被学生选修了的课程号；



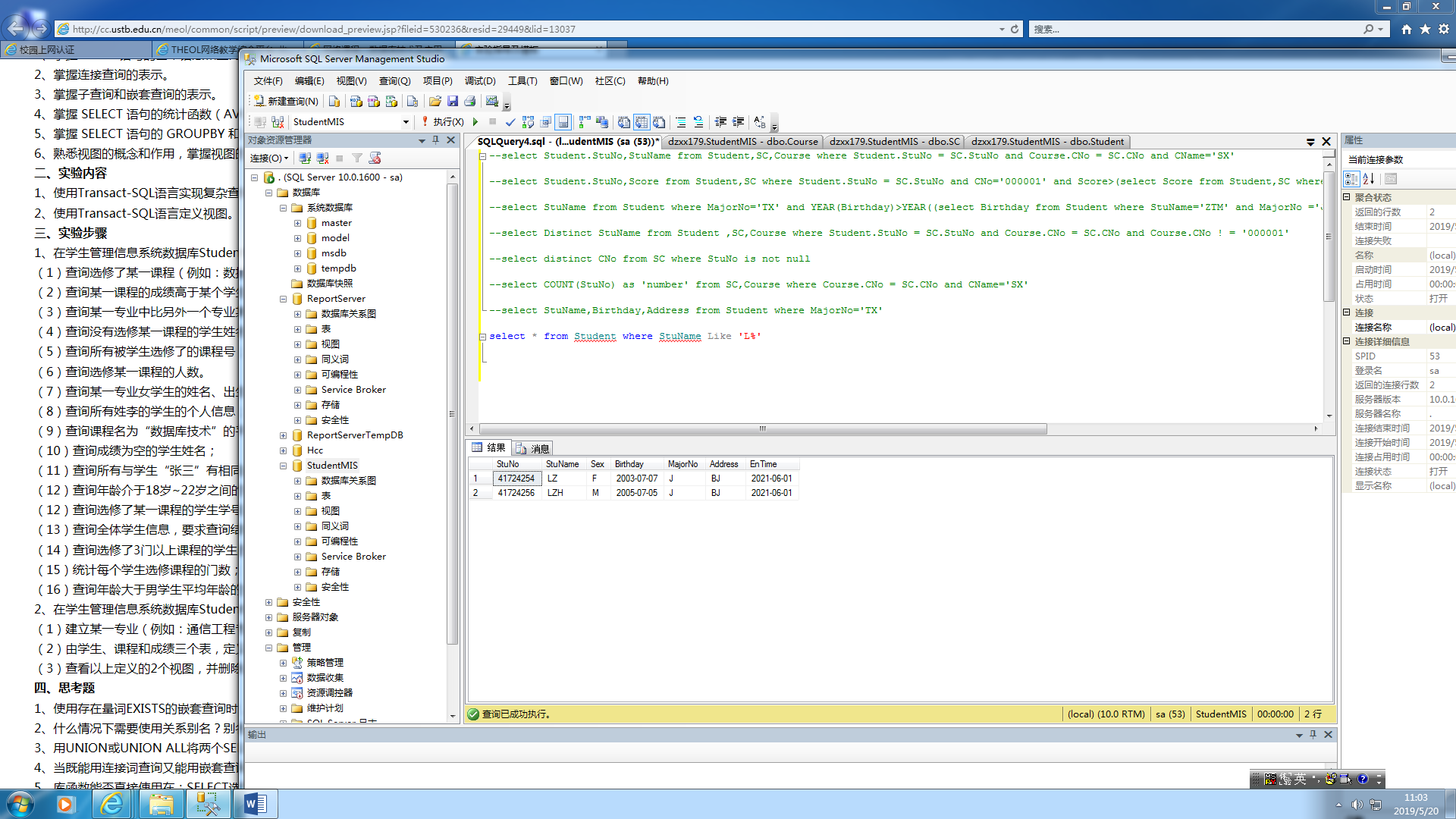
1. 查询选修某一课程的人数。



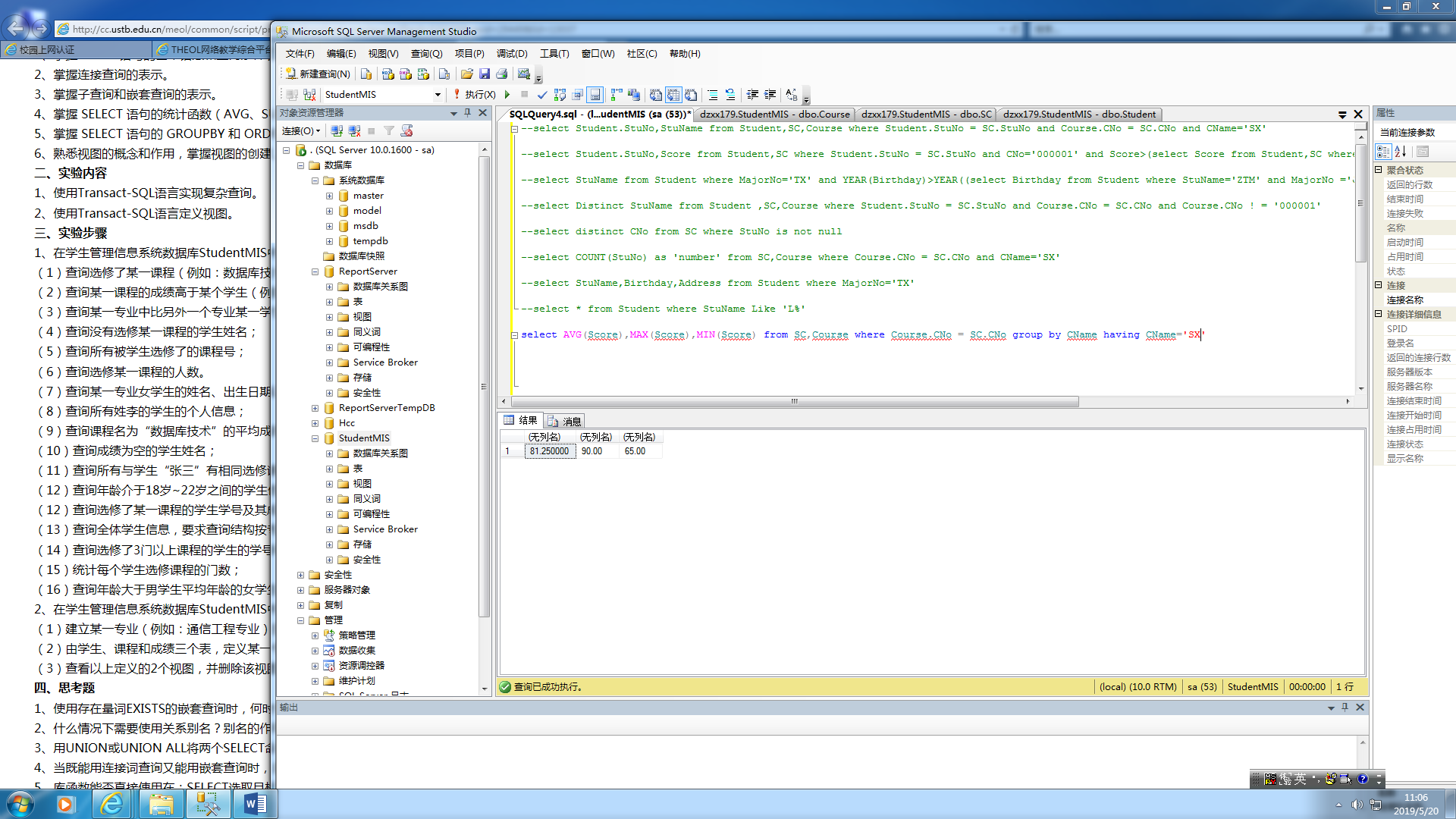
1. 查询某一专业女学生的姓名、出生日期以及籍贯；



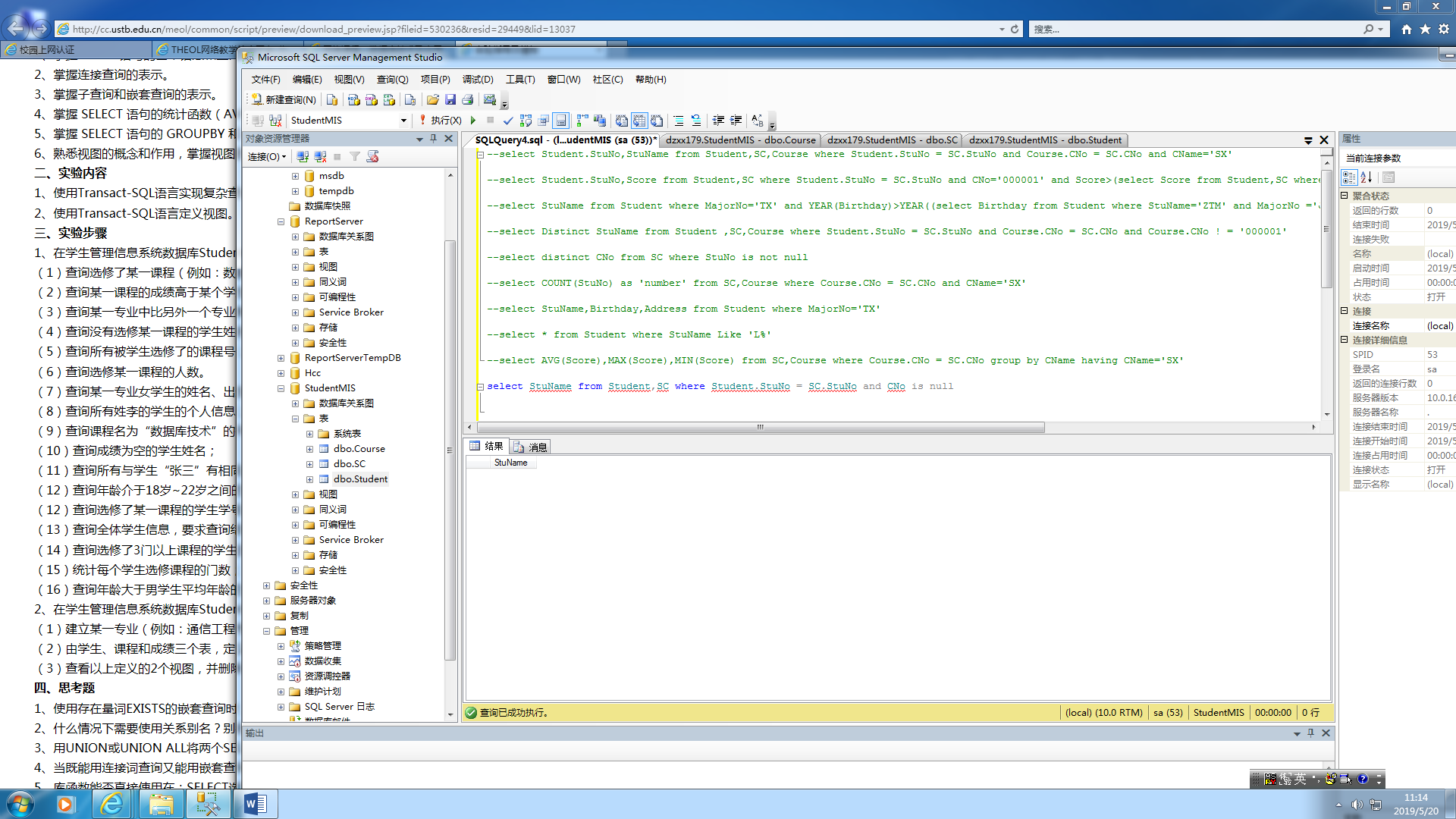
1. 查询所有姓李的学生的个人信息；



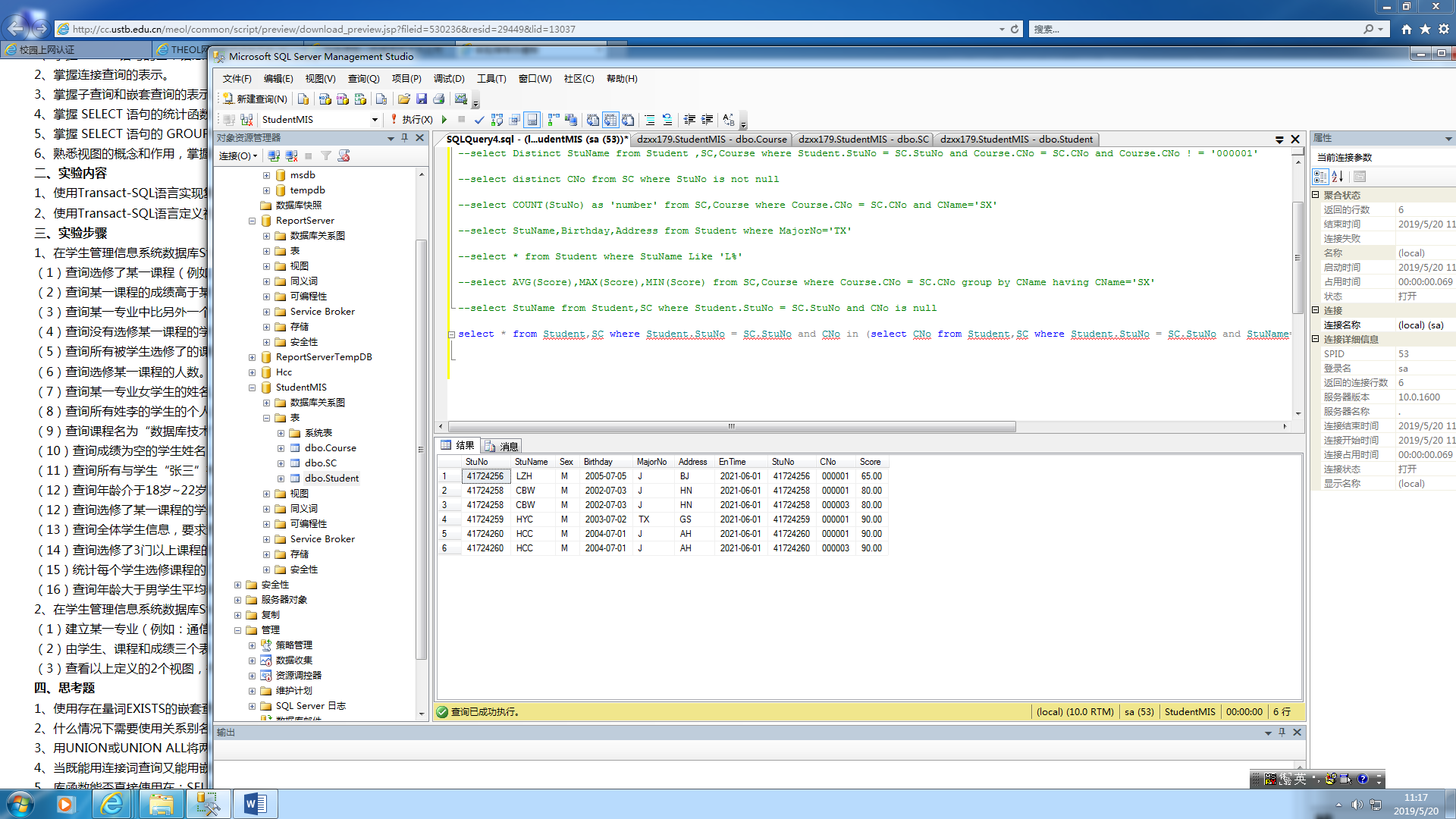
1. 查询课程名为“数据库技术”的平均成绩、最高分、最低分；



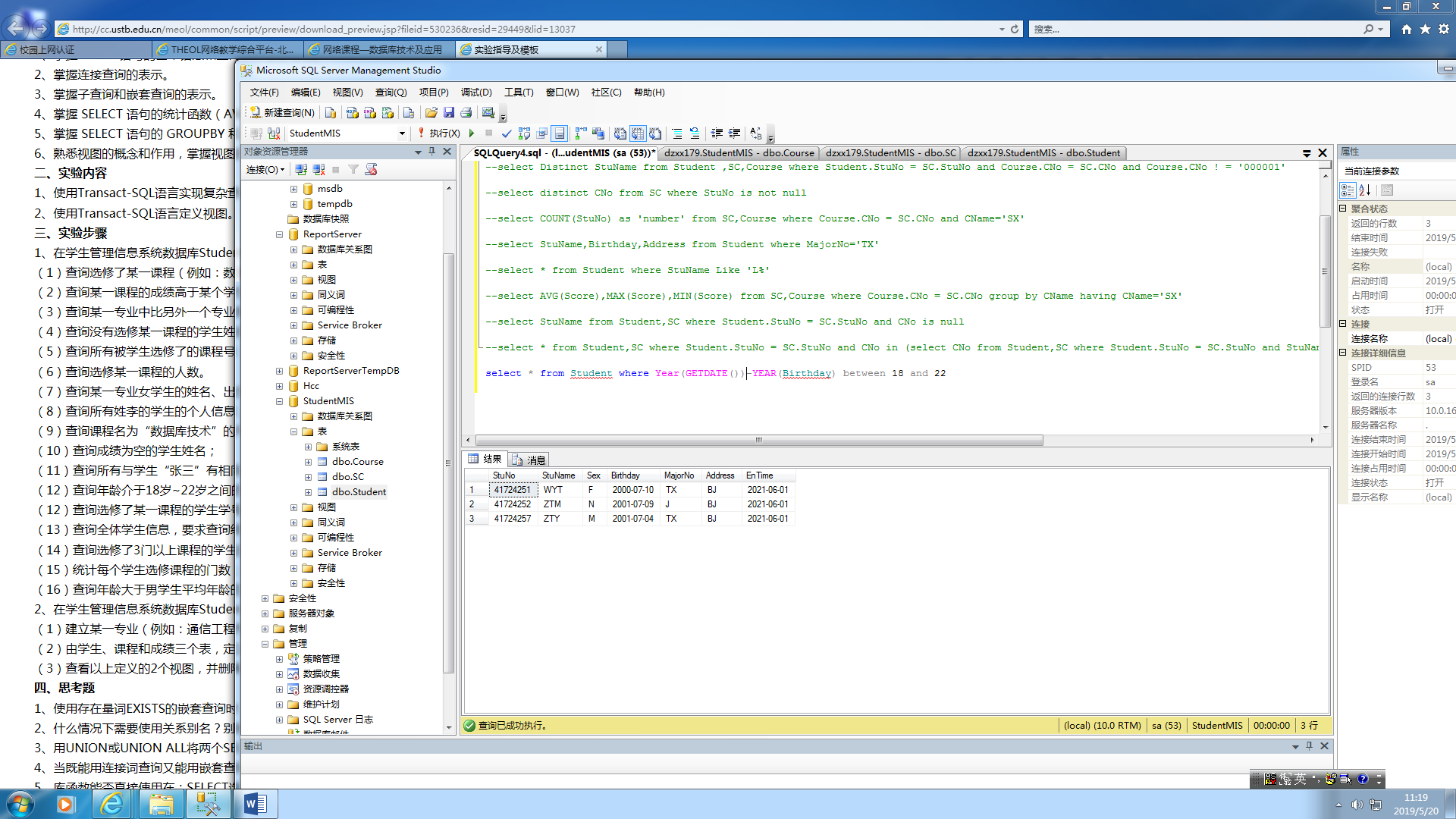
1. 查询成绩为空的学生姓名；



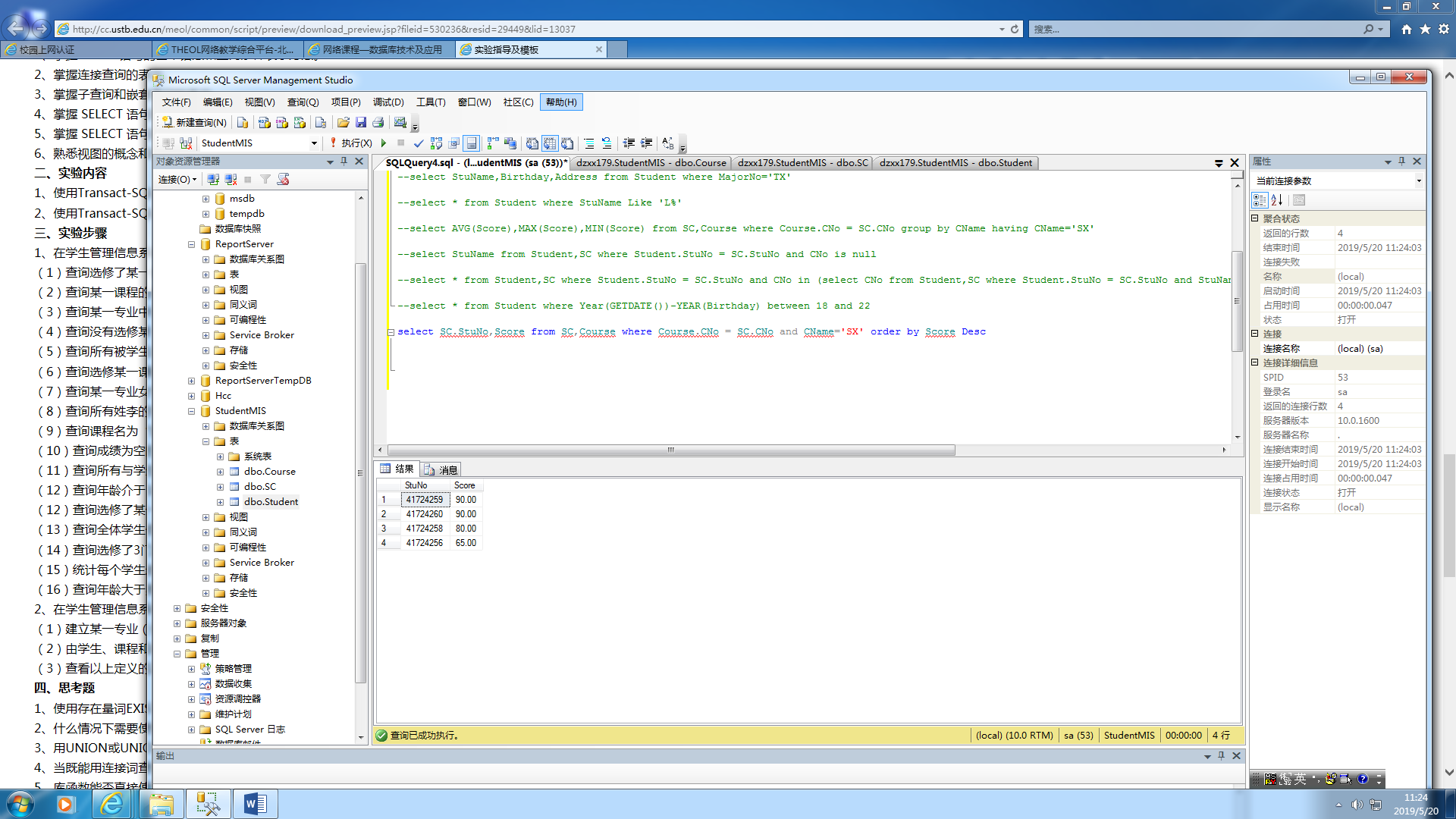
1. 查询所有与学生“张三”有相同选修课程的学生信息；



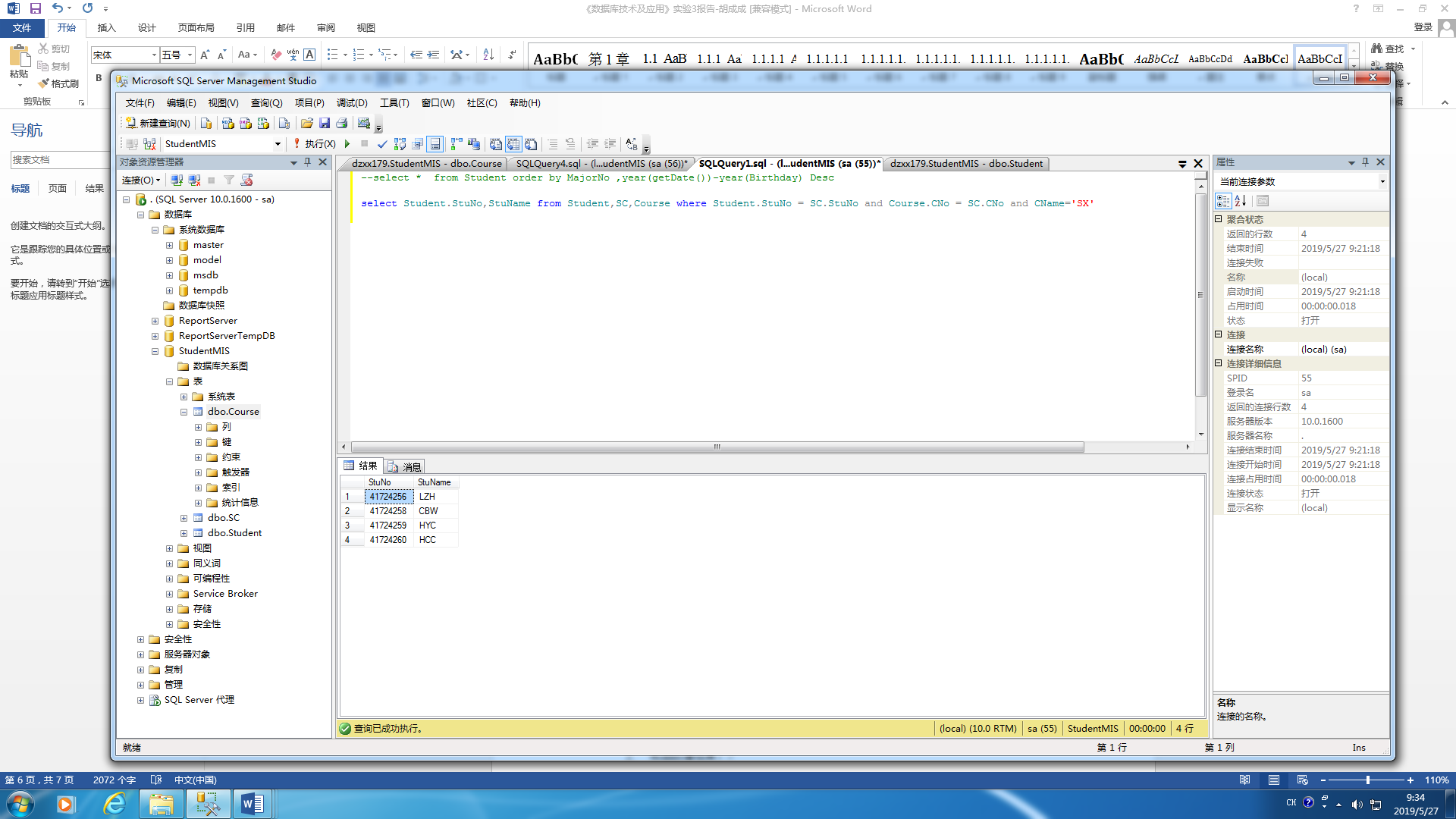
1. 查询年龄介于18岁~22岁之间的学生信息；



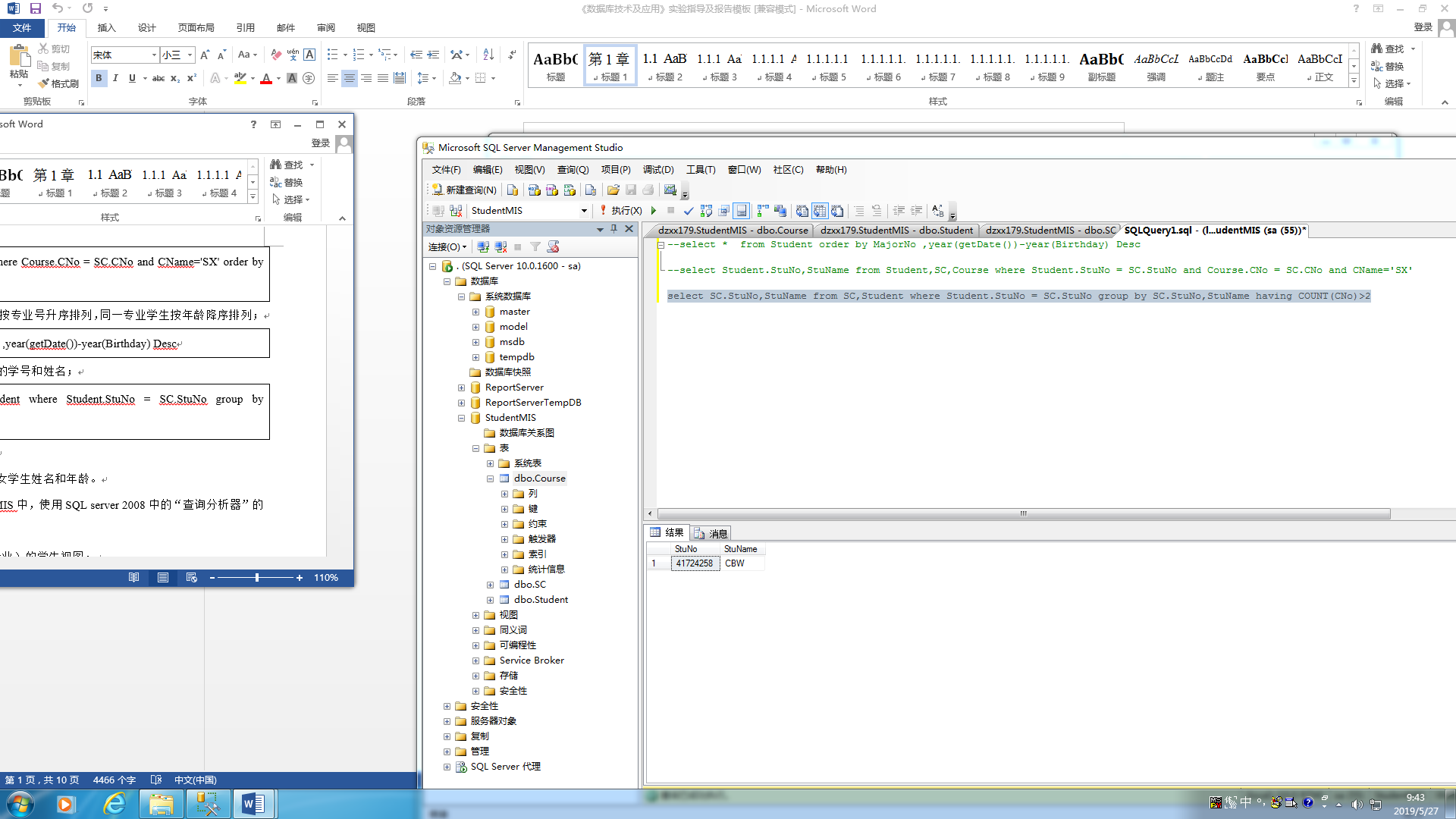
1. 查询选修了某一课程的学生学号及其成绩，并按成绩降序排列；



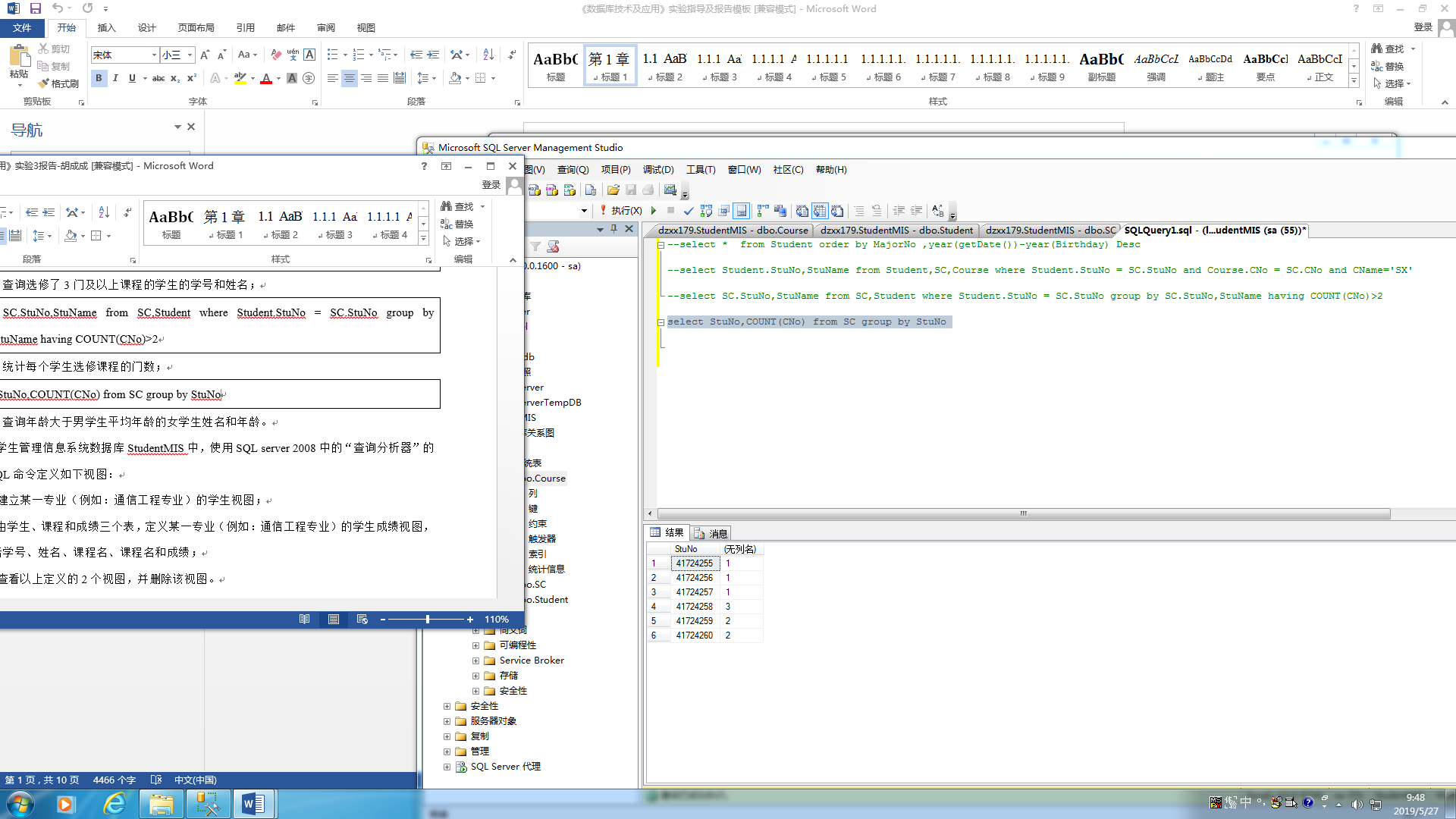
1. 查询全体学生信息，要求查询结构按专业号升序排列，同一专业学生按年龄降序排列；



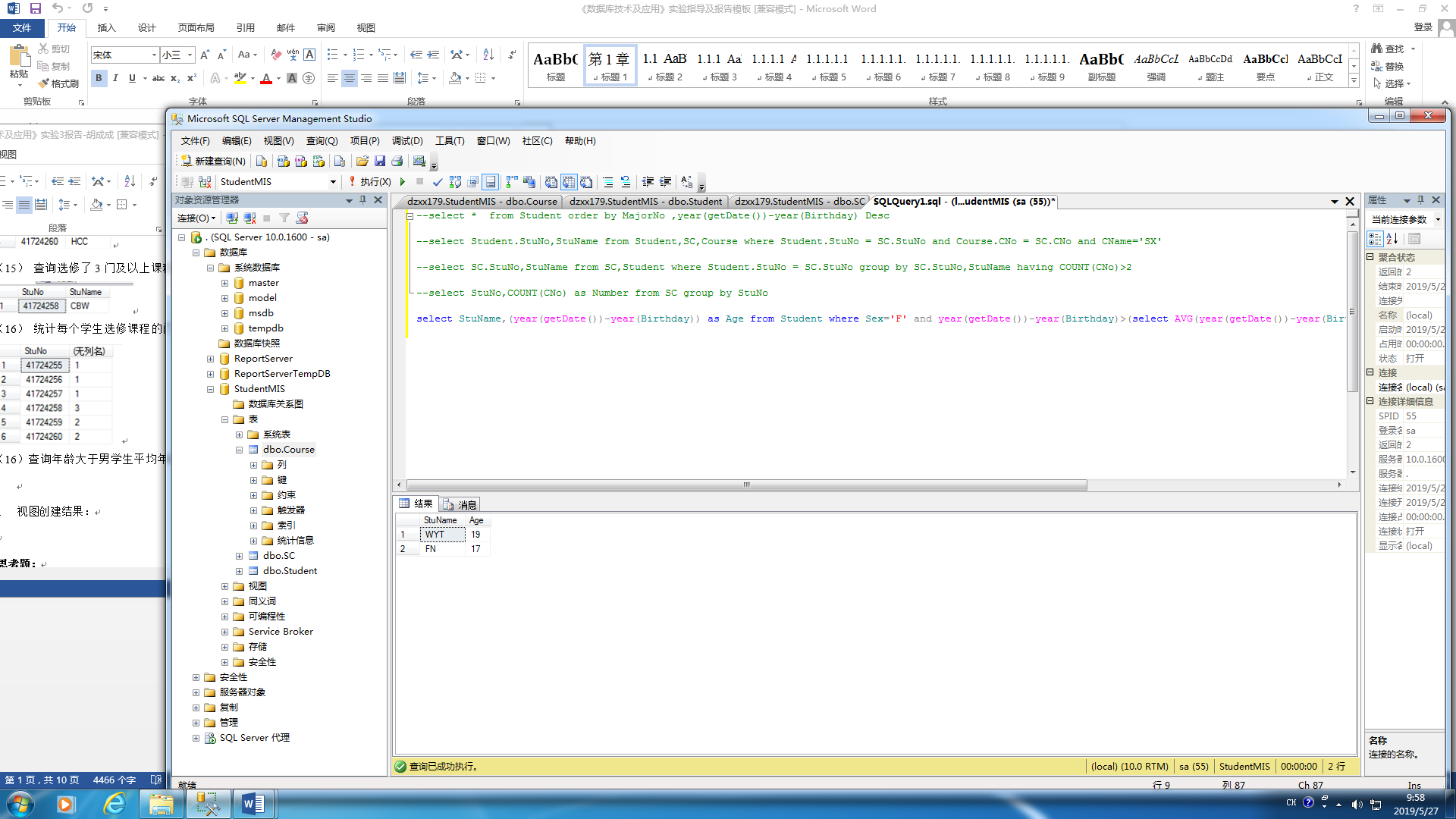
1. 查询选修了3门及以上课程的学生的学号和姓名；



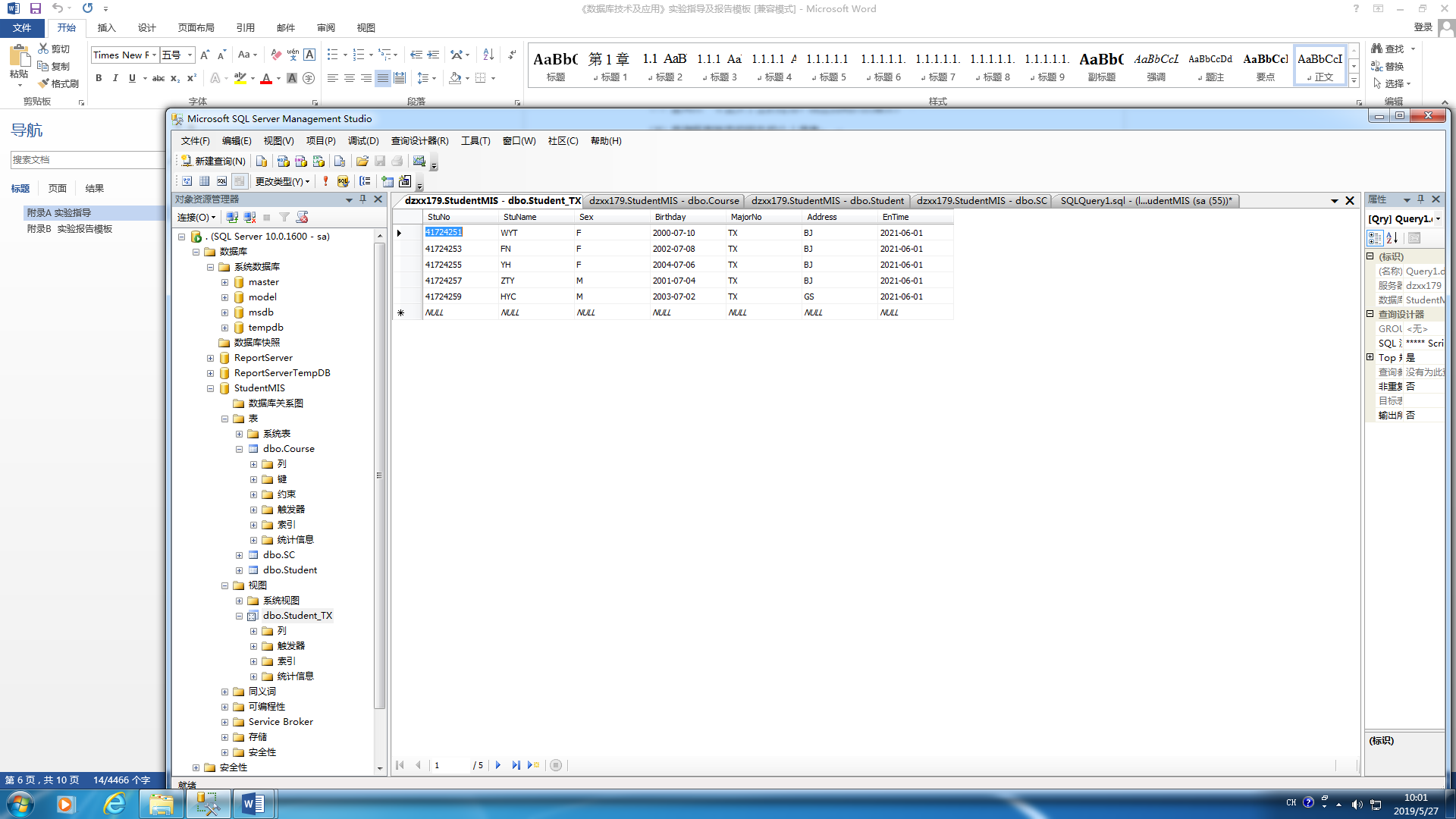
1. 统计每个学生选修课程的门数；



（17）查询年龄大于男学生平均年龄的女学生姓名和年龄。



1. 视图创建结果：
2. 建立某一专业（例如：通信工程专业）的学生视图；

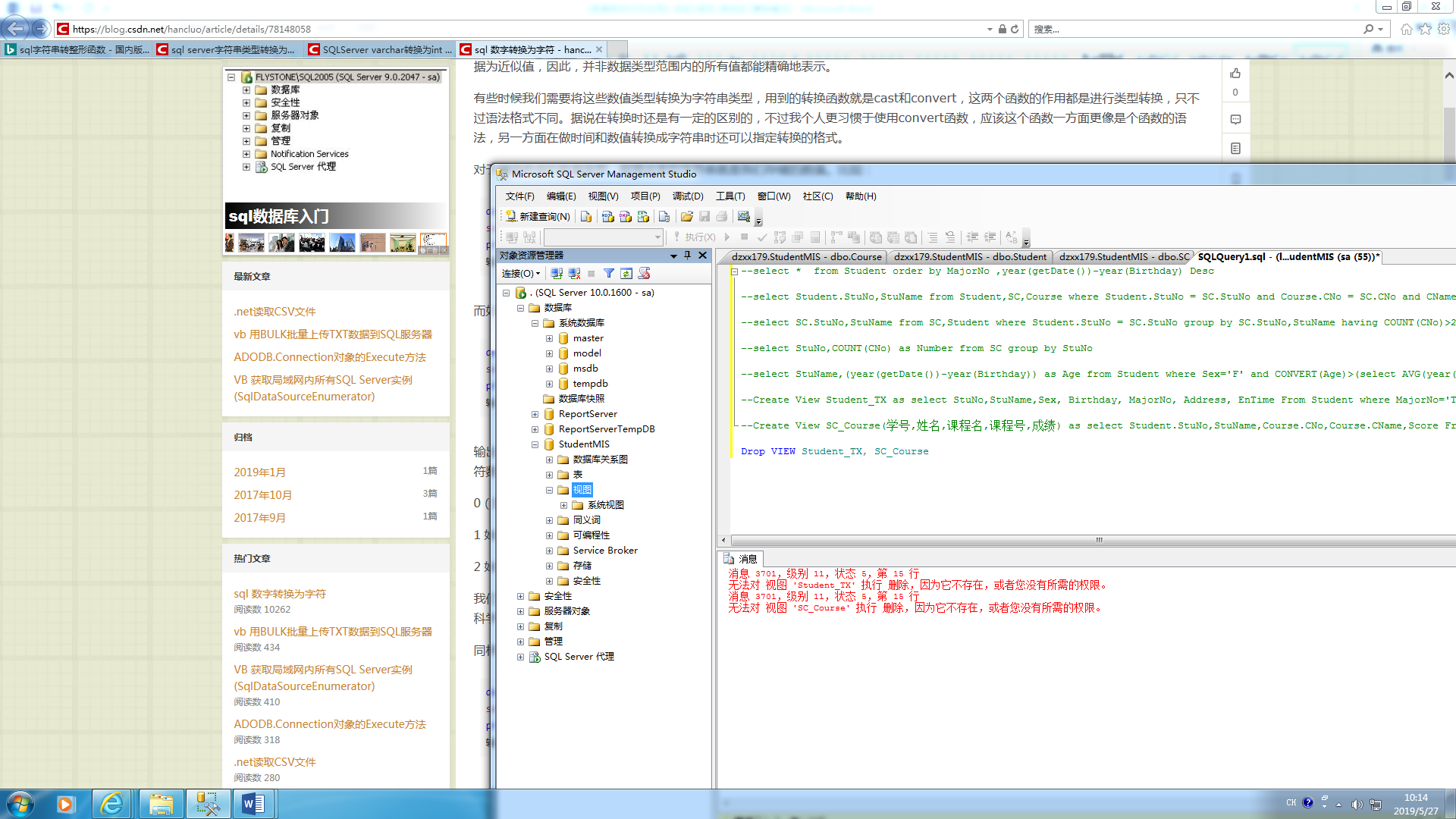
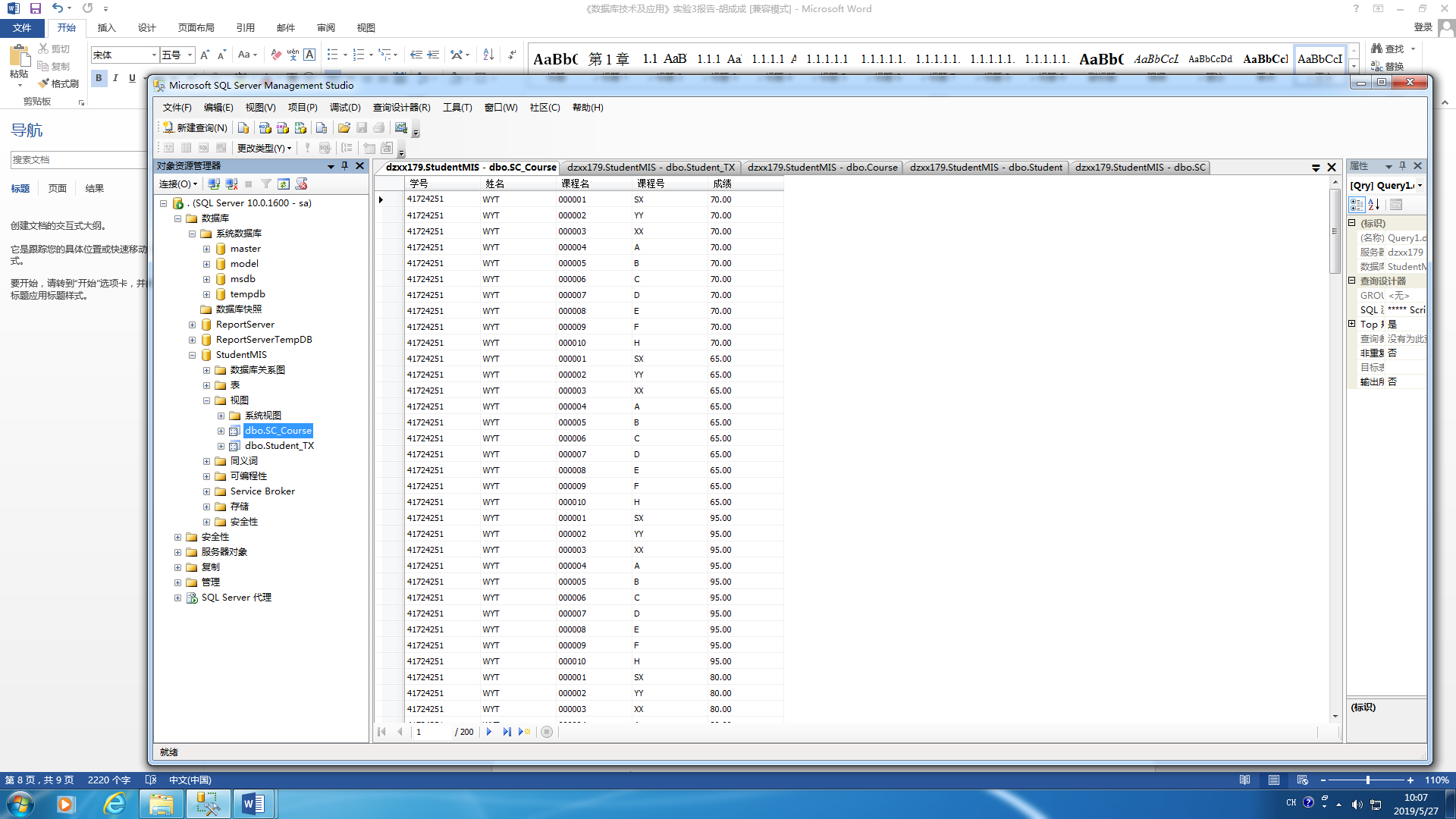


1. 由学生、课程和成绩三个表，定义某一专业（例如：通信工程专业）的学生成绩视图，其属性包括学号、姓名、课程名、课程名和成绩；



注：视图表格太长，截取了一部分

（3）查看以上定义的2个视图，并删除该视图。



**思考题：**

1. 使用存在量词EXISTS的嵌套查询时，何时外层查询的WHERE条件为真，何时为假。

Exists内查询结果为空，返回假，否则，返回真

1. 什么情况下需要使用关系别名？别名的作用范围是什么？

最少两种情况是需要别名的：

一是当两个家伙名字相同的时候，其中一个必须用别名来区分，否则不知道谁是谁了

如：SELECT A.\* FROM TB A, TB B WHERE A.ID=B.PARENTID

二是当一个家伙没有名字时，必须加个名字给它

如：SELECT \* FROM (SELECT \* FROM TB) T

1. 用UNION或UNION ALL将两个SELECT命令结合为一个时，结果有何不同？

UNION 操作符用于合并两个或多个 SELECT 语句的结果集。UNION 内部的 SELECT 语句必须拥有相同数量的列。列也必须拥有相似的数据类型。同时，每条 SELECT 语句中的列的顺序必须相同。默认地，UNION 操作符选取不同的值。如果允许重复的值，使用 UNION ALL。

1. 当既能用连接词查询又能用嵌套查询时，应该选择哪种查询较好？为什么？

选用连接词查询。因为连接词查询，效率比嵌套查询高。

1. 库函数能否直接使用在：SELECT选取目标、HAVING子句、WHERE子句、GROUP BY列名中？

库函数可以使用在SELECT选取目标、HAVING字句、WHERE子句，不能使用在GROUP BY列名中。

6、视图如何使用？

创建用户视图，根据用户需求设计出相应视图属性信息。