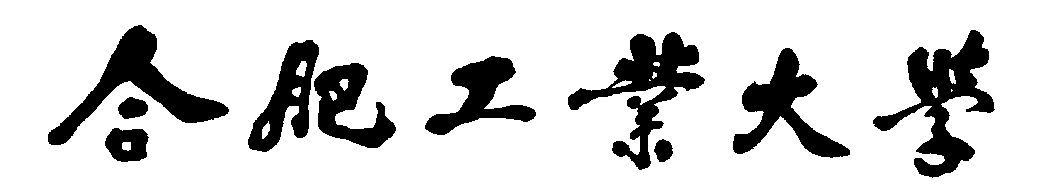
****

计算机与信息学院

数据库系统实验报告

|  |  |
| --- | --- |
| 专 业 班 级 | 计算机科学与技术 22-4班 |
| 学生姓名及学号 | 侯腾跃 2022217477 |
| 课程教学班号 | 数据库系统 |
| 任 课 教 师 | 郑淑敏、胡敏 |
| 实验指导教师 | 胡敏 |
| 实验地点 | 计算机中心第二机房 |
| 2024 ~2025 学年第 二 学期 | |

**目录**

[实验一 数据库的基本操作 2](#_Toc30818)

[一、实验目的 2](#_Toc24724)

[二、实验准备 2](#_Toc29404)

[三、实验要求 2](#_Toc31387)

[四、实验内容 3](#_Toc2572)

[实验二 触发器、存储过程 27](#_Toc94)

[一、实验目的 27](#_Toc9579)

[二、实验准备 27](#_Toc31854)

[三、实验要求 27](#_Toc19027)

[四、实验内容 28](#_Toc14212)

[五、实验总结 30](#_Toc15427)

[实验三 数据库的安全性、备份与恢复 30](#_Toc5698)

[一、实验目的 30](#_Toc10242)

[二、实验准备 30](#_Toc18999)

[三、实验要求 31](#_Toc31015)

[四、实验内容 31](#_Toc6112)

[五、实验总结 34](#_Toc13452)

# 实验一 数据库的基本操作

## 一、实验目的

1. 掌握关专业数据库管理专业统的基本命令，理解数据库专业统的基本概念。
2. 掌握使用SQL语句完成数据的定义、查询和更新等功能，熟练使用SQL语言管理数据库。

## 二、实验准备

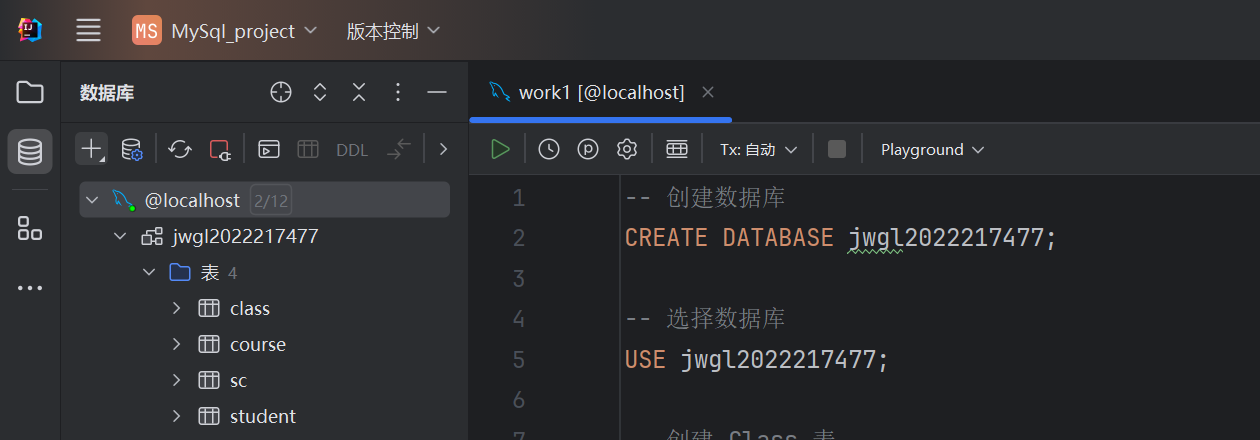
1. 实验环境： MySQ8.0版本+IntelliJ IDEA 2024.1.2熟悉相关数据库管理软件。

## 三、实验要求

1. 安装MySQ8.0 以上版本数据库与IntelliJ IDEA可视化数据库管理工具。
2. 通过IntelliJ IDEA连接数据库；
3. 创建和维护学教务管理数据库：“jwgl”+“自己学号”（后面简称jwgl），利用navicat的相关机制，创建并维护jwgl数据库。
4. 实验完成后完成实验报告，在实验报告中附上相应的代码和实验过程截图。

## 四、实验内容

**1． 创建数据库jwgl（“jwgl”+“自己学号”）**



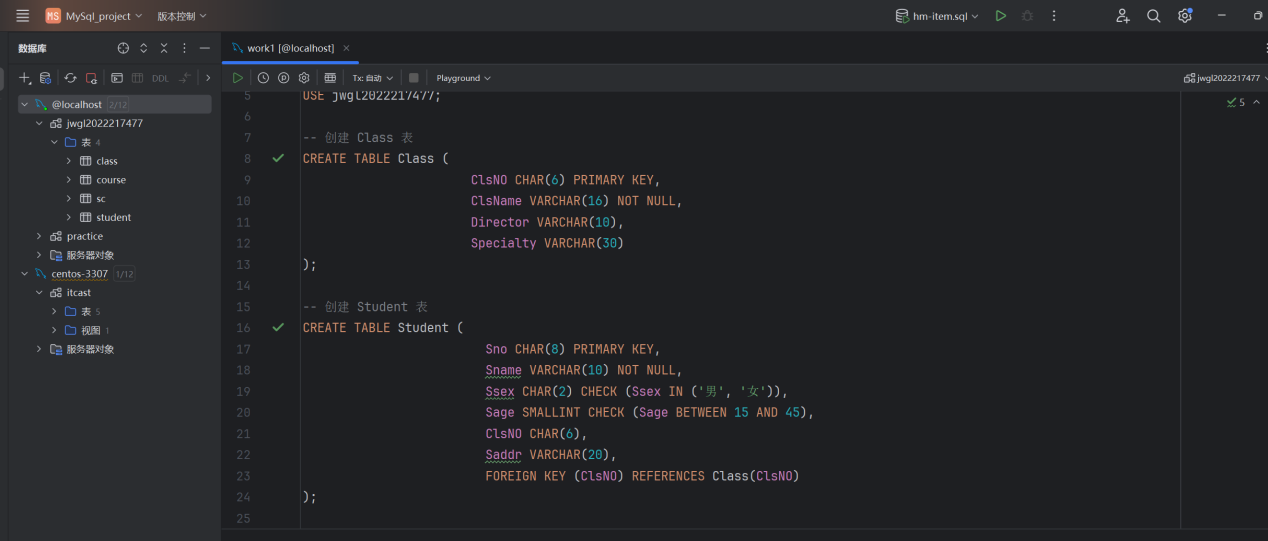
1. **在jwgl数据库中创建如下几个表，要求使用SQL命令创建。**

Class 表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| ClsNO | 班级号 | 普通编码定长字符串，长度为6 | 主码 |
| ClsName | 班级名 | 普通编码可变长字符串，长度为16 | 非空 |
| Director | 辅导员 | 普通编码可变长字符串，长度为10 |  |
| Specialty | 专业 | 普通编码可变长字符串，长度为30 |  |

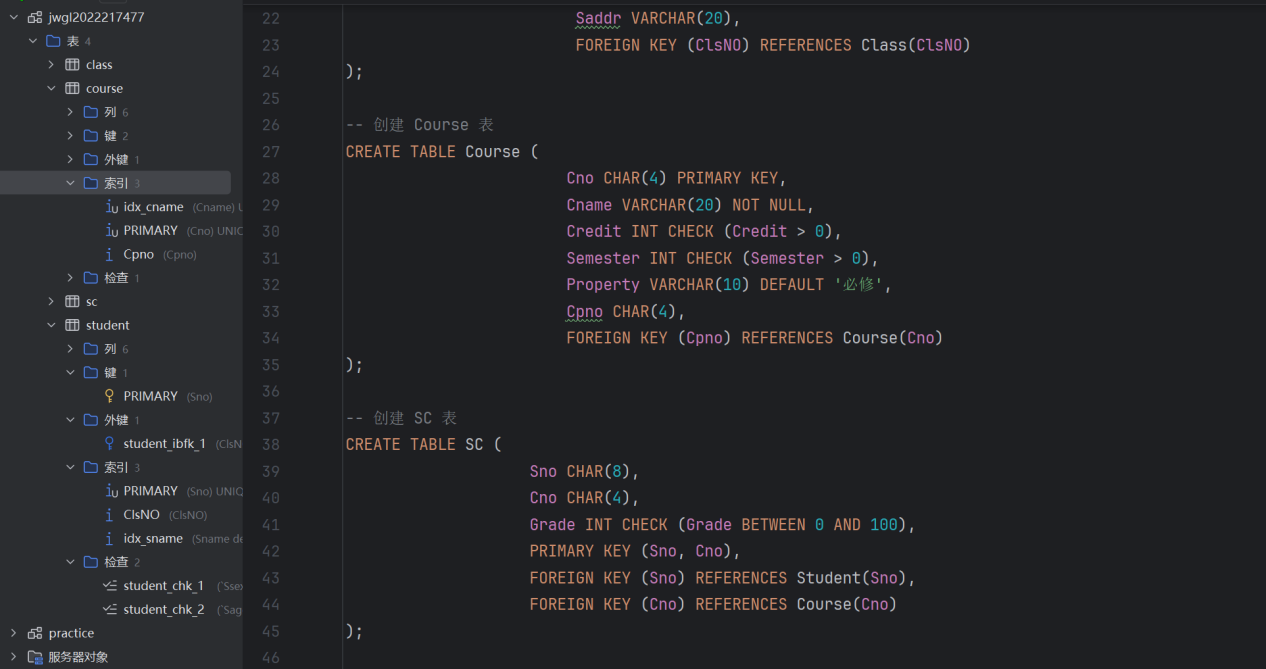
Student 表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| Sno | 学号 | 普通编码定长字符串，长度为8 | 主码 |
| Sname | 姓名 | 普通编码可变长字符串，长度为10 | 非空 |
| Ssex | 性别 | 普通编码定长字符串，长度为2 | 取‘男’或‘女’ |
| Sage | 年龄 | 微整型 | 取值15～45 |
| ClsNO | 所在班级编号 | 普通编码定长字符串，长度为6 | 外键，参照表Class |
| Saddr | 住址 | 普通编码可变长字符串，长度为20 |  |



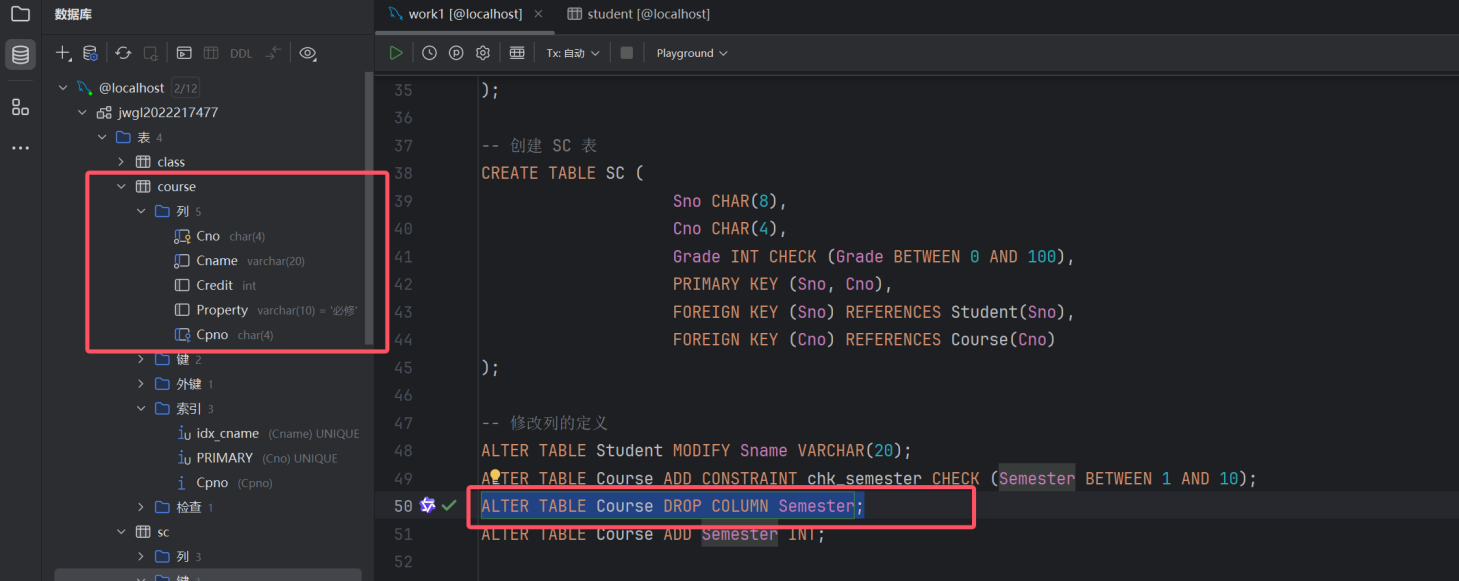
Course 表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名 | | 说明 | | 数据类型 | | | 约束 |
| Cno | | 课程号 | | 字符串，长度为4 | | | 主码 |
| Cname | | 课程名 | | 字符串，长度为20 | | | 非空 |
| Ccredit | | 学分 | | 整数 | | | 取值大于0 |
| Semester | | 学期 | | 整数 | | | 取值大于0 |
| Property | | 课程性质 | | 字符串，长度为10 | | | 默认值为必修 |
| Cpno | | 先修课 | | 字符串，长度为4 | | | 外码，参照Course(Cno) |
|  | |  | | sc 表 | | |  |
| 列名 | | 说明 | | 数据类型 | 约束 | | |
| Sno | | 学号 | | 字符串，长度为8 | 主码，引用Student的外码 | | |
| Cno | | 课程号 | | 字符串，长度为4 | 主码，引用Course的外码 | | |
| Grade | | 成绩 | | 整数 | 取值0～100 | | |

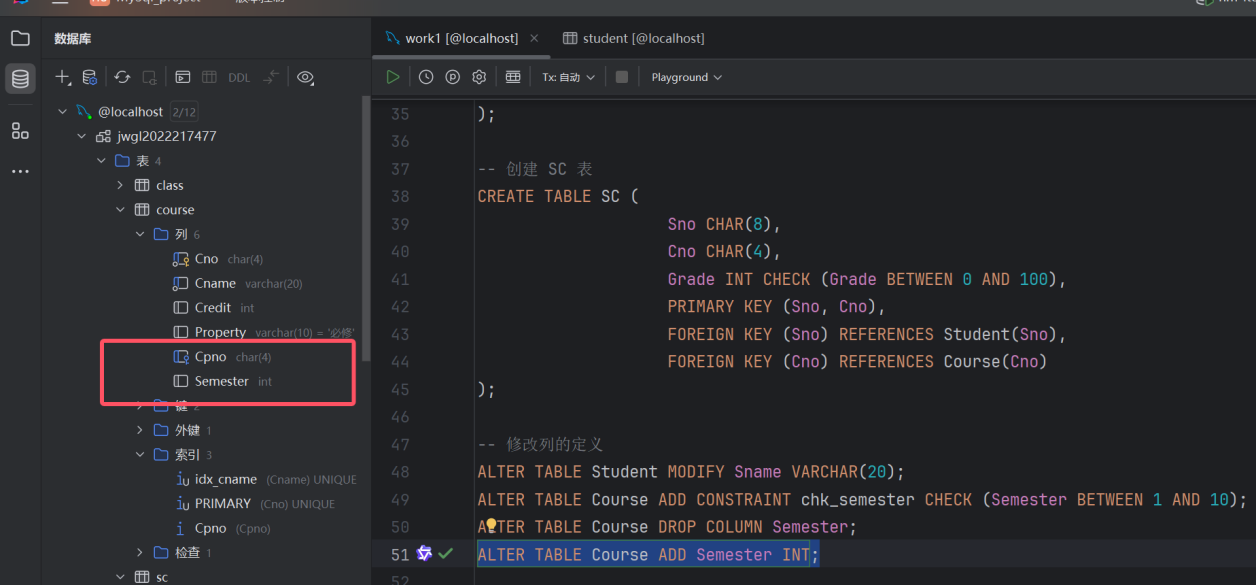


**3.修改列的定义**

1. 用SQL语句将Student表中的属性sname varchar(8)改成varchar(20)类型。
2. 用SQL语句在Course表的Semester字段添加约束，Semester的属性值在1-10之间。
3. 用SQL语句将Course表中的Semester字段删除。



1. 用SQL语句向Course表添加Semester字段。



**4.建立以下索引**

(1) 在student表的sname列上建立普通降序索引。

(2) 在course表的cname列上建立唯一索引。

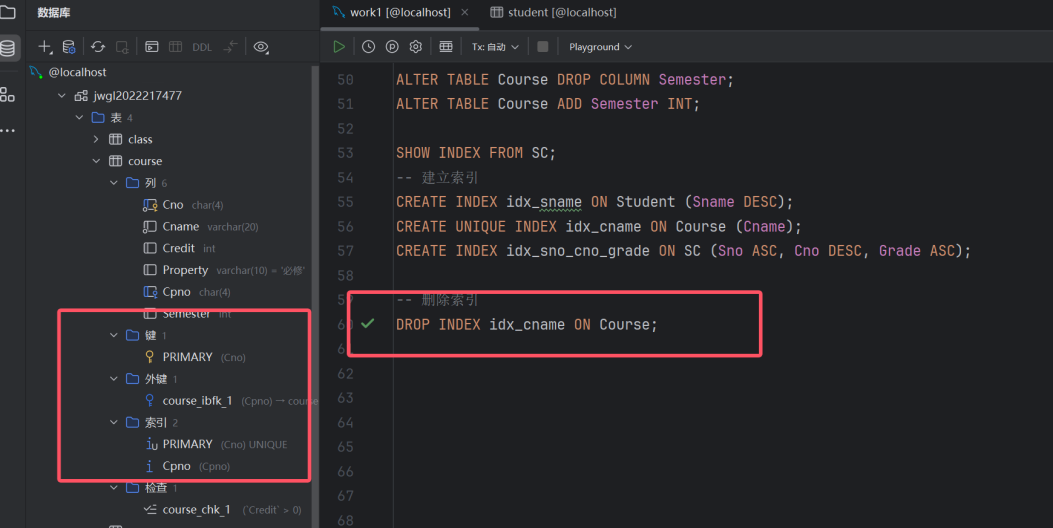
(3) 在sc表的sno列上建立聚集索引。

(4) 在sc表的sno(升序)、 cno (降序)和grade(升序)三列上建立一个普通索引。

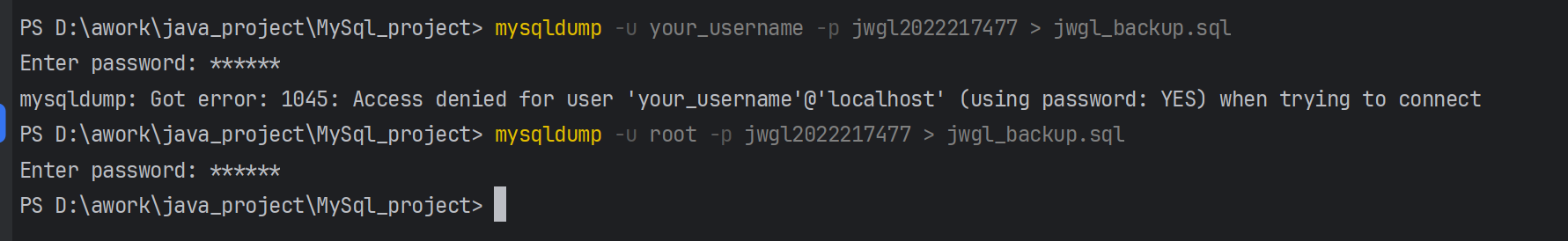


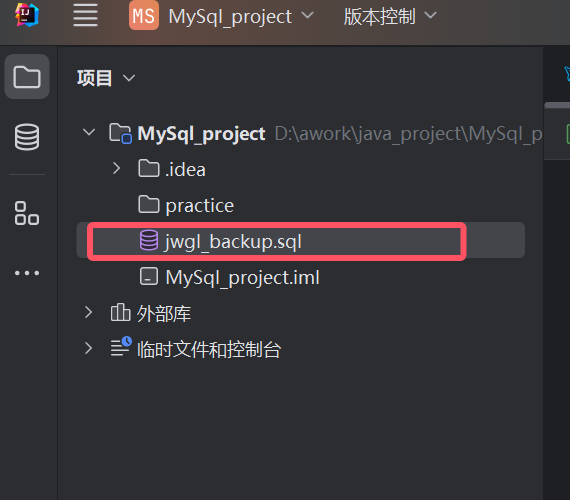
**5.删除索引**

将course表的cname列的唯一索引删掉。



**6.备份和恢复数据库jwgl。**



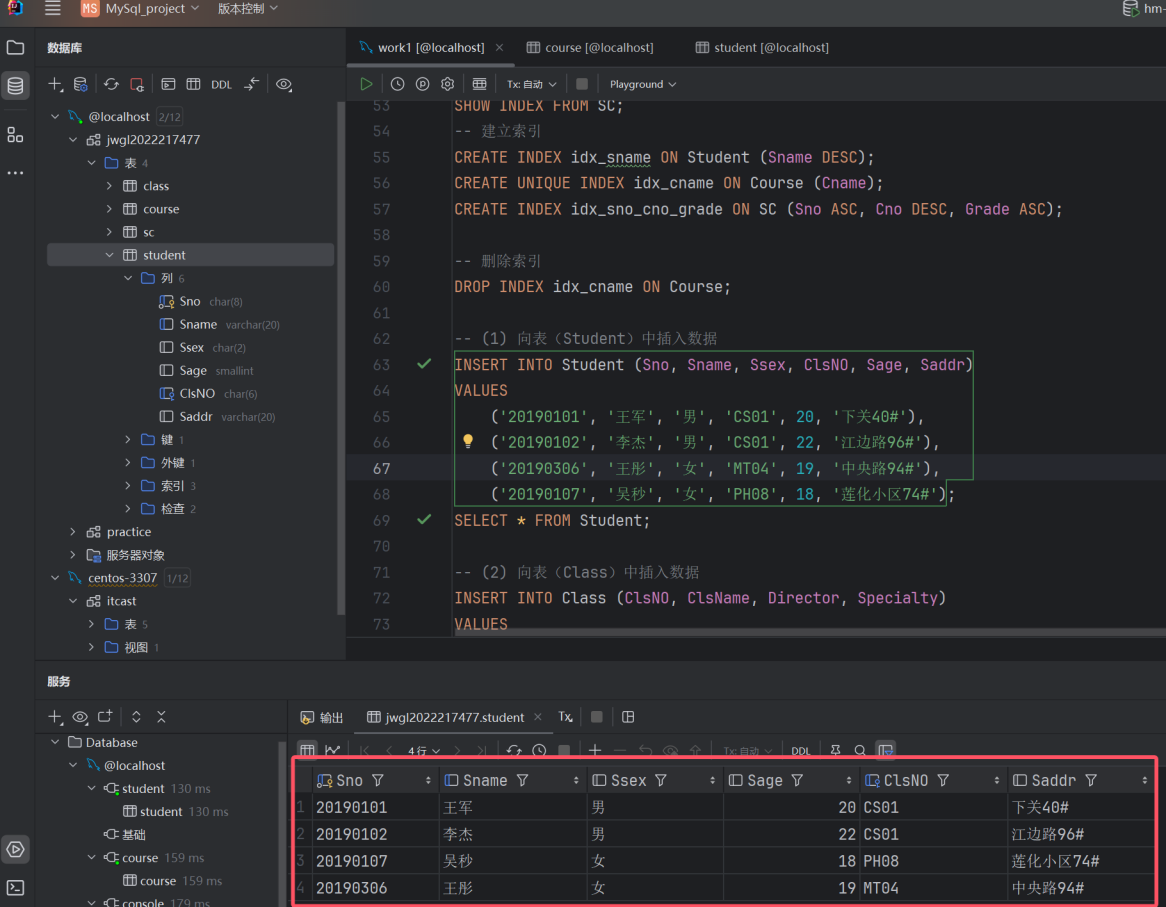


**7.向Class、student、course、sc数据表中输入一些数据。**

(1) 向表（Student）中插入数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sno** | **Sname** | **Ssex** | **ClsNO** | **Sage** | Saddr |
| 20190101 | 王军 | 男 | CS01 | 20 | 下关40# |
| 20190102 | 李杰 | 男 | CS01 | 22 | 江边路96# |
| 20190306 | 王彤 | 女 | MT04 | 19 | 中央路94# |
| 20190107 | 吴杪 | 女 | PH08 | 18 | 莲化小区74# |

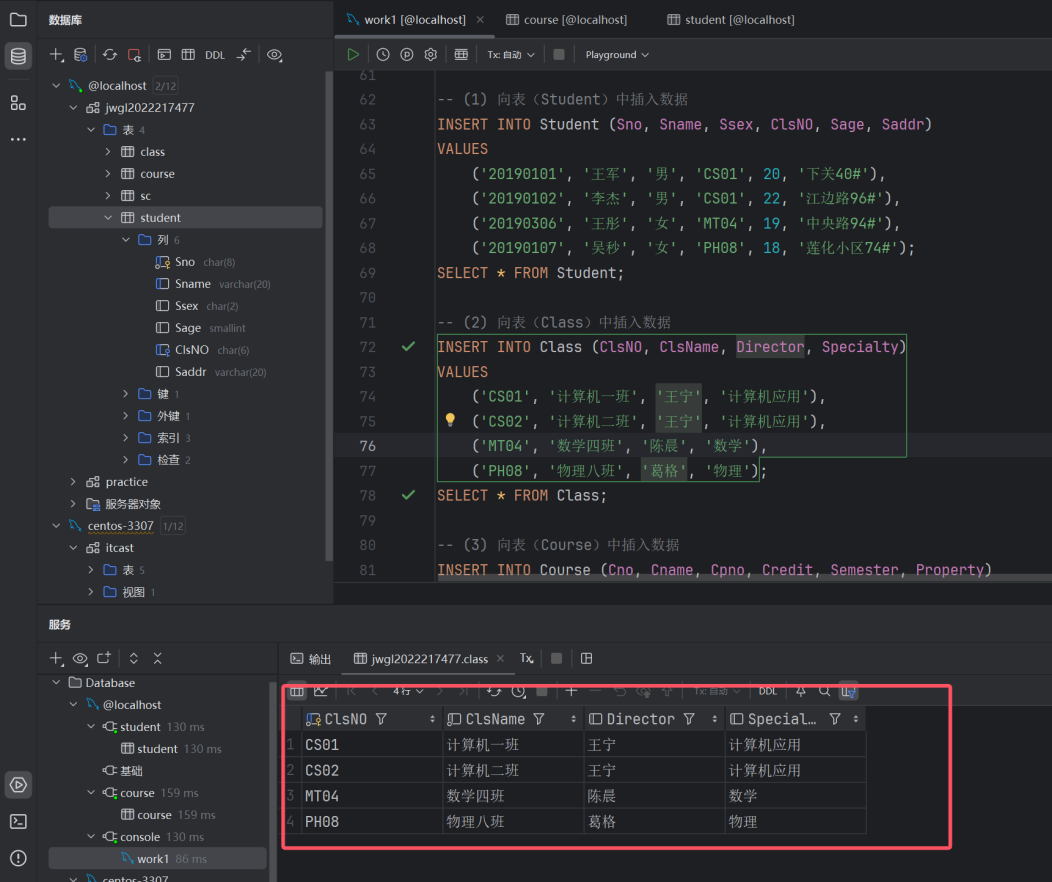
插入数据之后使用命令：Select \* from Student; 检查插入数据的正确性



（2）向表（Class）中插入数据

|  |  |  |  |
| --- | --- | --- | --- |
| **ClsNO** | **ClsName** | **Director** | **Specialty** |
| CS01 | 计算机一班 | 王宁 | 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 |

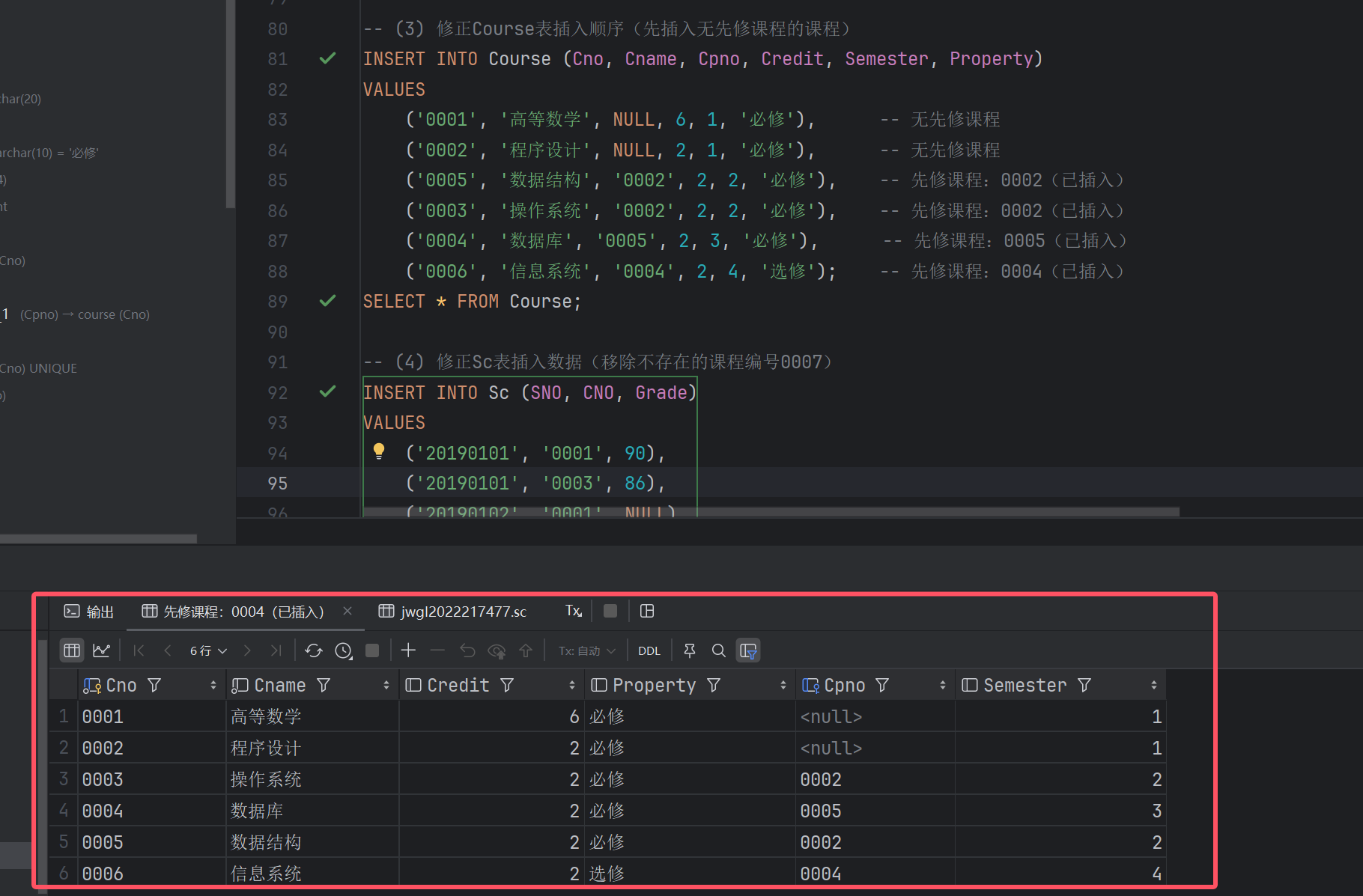
插入数据之后使用命令：Select \* from Class; 检查插入数据的正确性



（3）向表（Course ）中插入数据

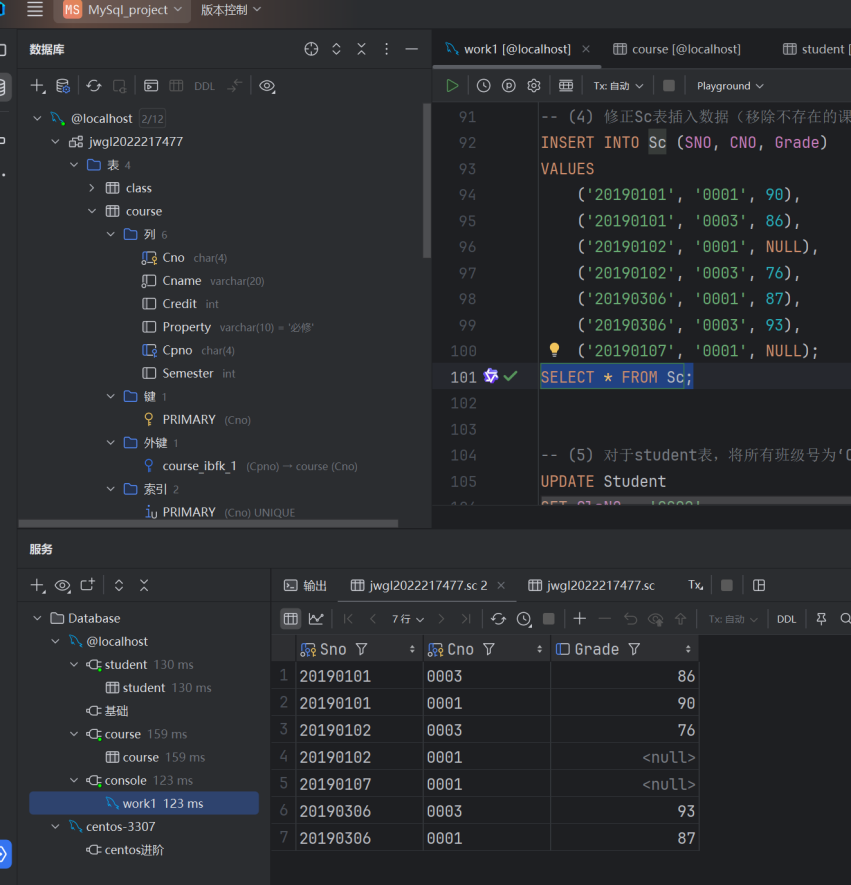
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cno** | **Cname** | **Cpno** | **Credit** | **Semester** | **Property** |
| 0001 | 高等数学 | Null | 6 | 1 | 必修 |
| 0002 | 程序设计 | NULL | 2 | 1 | 必修 |
| 0003 | 操作系统 | 0002 | 2 | 2 | 必修 |
| 0004 | 数据库 | 0005 | 2 | 3 | 必修 |
| 0005 | 数据结构 | 0002 | 2 | 2 | 必修 |
| 0006 | 信息系统 | 0004 | 2 | 4 | 选修 |

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



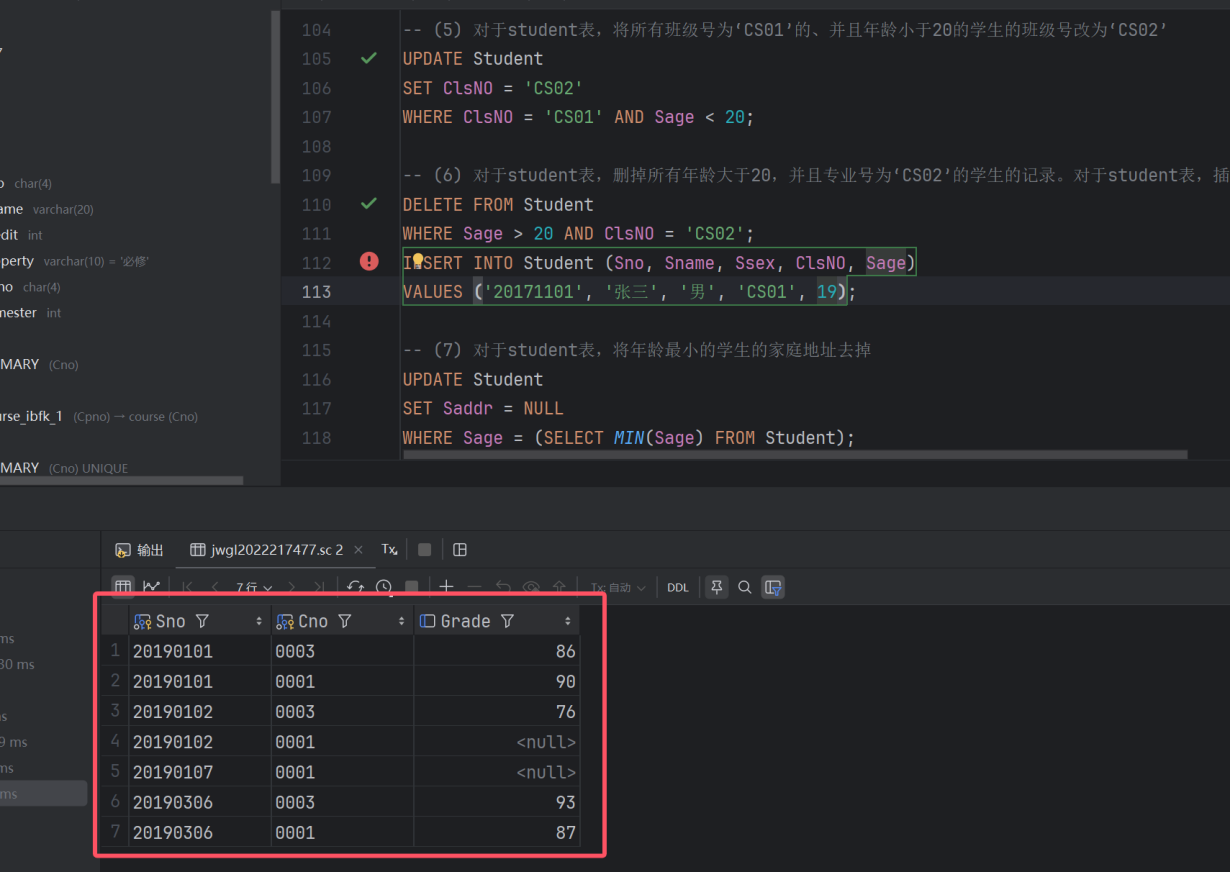
（4）向表（Sc ）中插入数据

|  |  |  |
| --- | --- | --- |
| **SNO** | **CNO** | **Grade** |
| 20190101 | 0001 | 90 |
| 20190101 | 0003 | 86 |
| 20190101 | 0007 | 87 |
| 20190102 | 0001 |  |
| 20190102 | 0003 | 76 |
| 20190306 | 0001 | 87 |
| 20190306 | 0003 | 93 |
| 20190107 | 0007 | 85 |
| 20190107 | 0001 | NULL |



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

对于student表，插入一条新记录，它的具体信息为，学号：20171101、姓名：张三、性别：男、年龄：19、班级编号：‘CS01’。



1. 对于student表，将年龄最小的学生的家庭地址去掉。



1. 对于student表，将平均年龄最小的一个班级编号改为‘GL01’

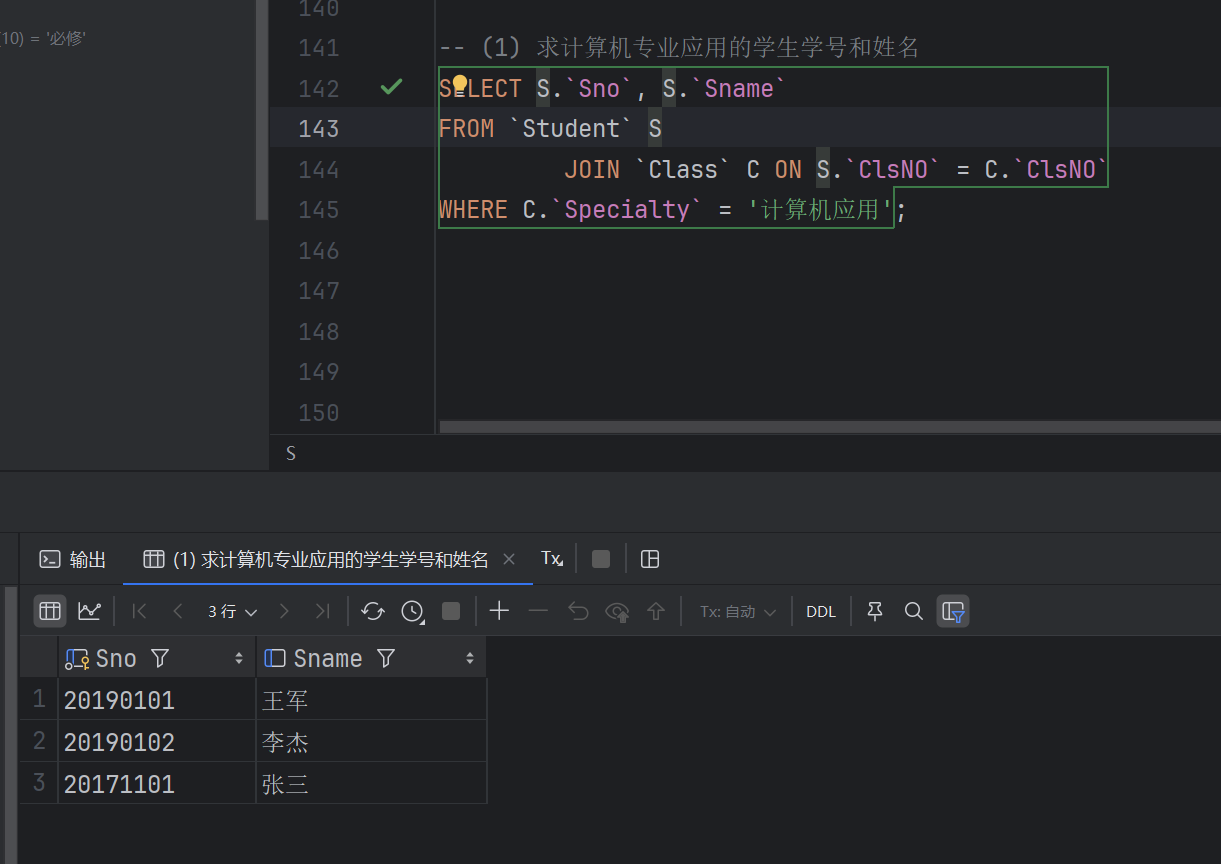


**8.简单查询操作**

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jwgl数据库实现以下查询：

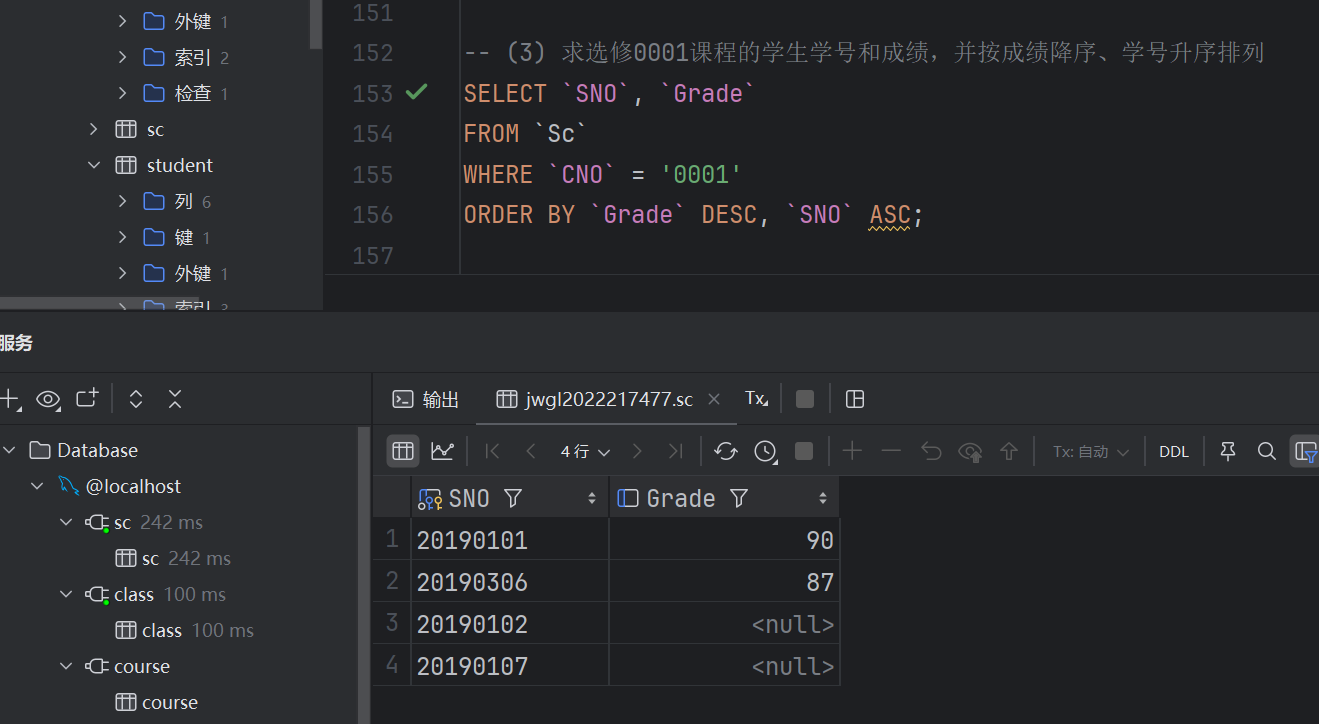
1. 求计算机专业应用的学生学号和姓名；



1. 求选修了课程的学生学号；



1. 求选修0001 课程的学生学号和成绩，并要求对查询结果按成绩的降序排列，如果成绩相同则按学号的升序排列；



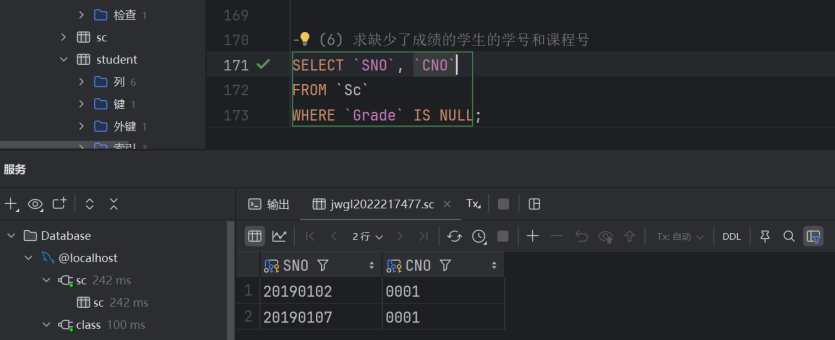
1. 求选修课程0001 且成绩在80－90 之间的学生学号和成绩，并将成绩乘以系数0.75 输出；



1. 求计算机应用和数学专业的姓张的学生的信息；



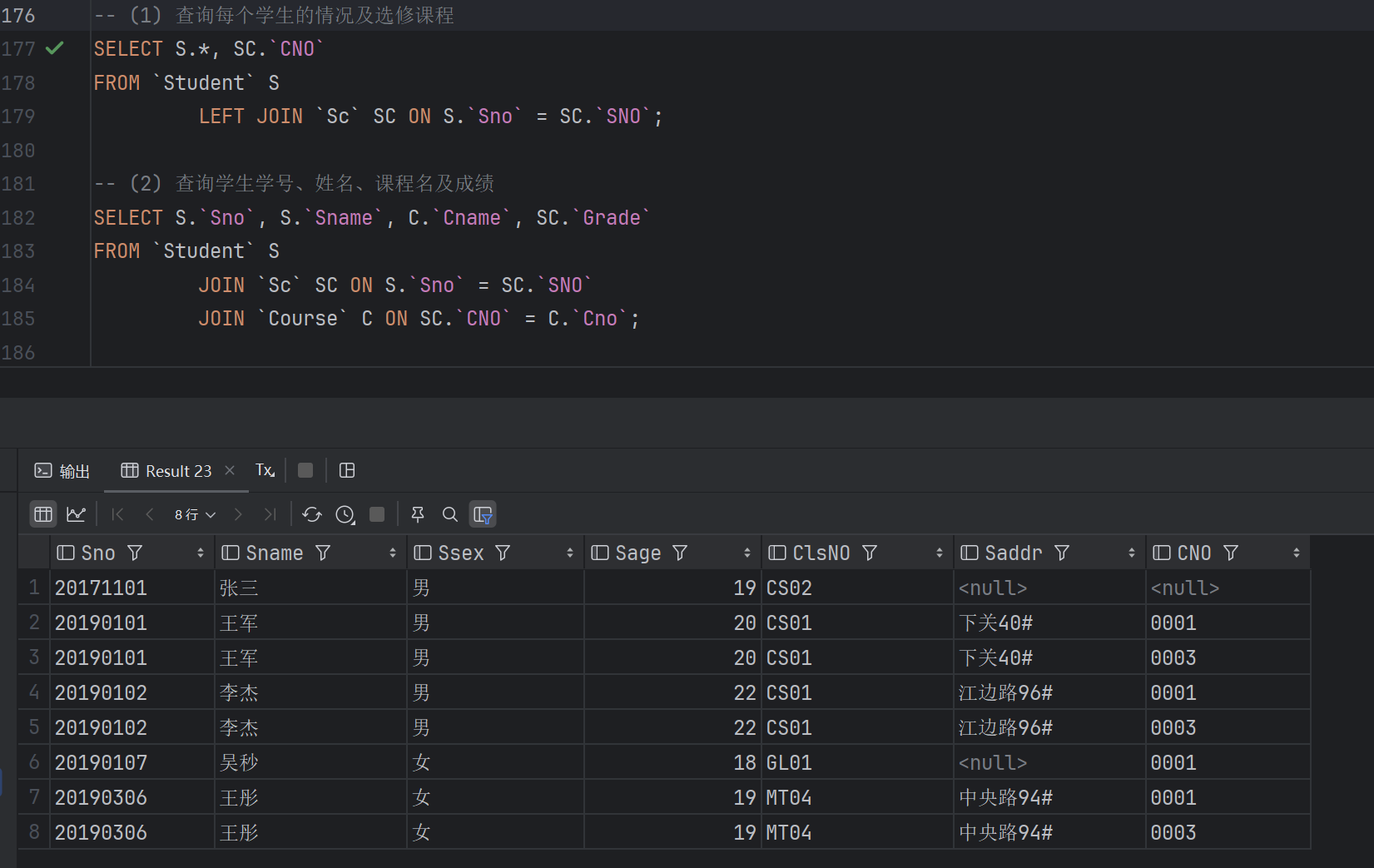
1. 求缺少了成绩的学生的学号和课程号。



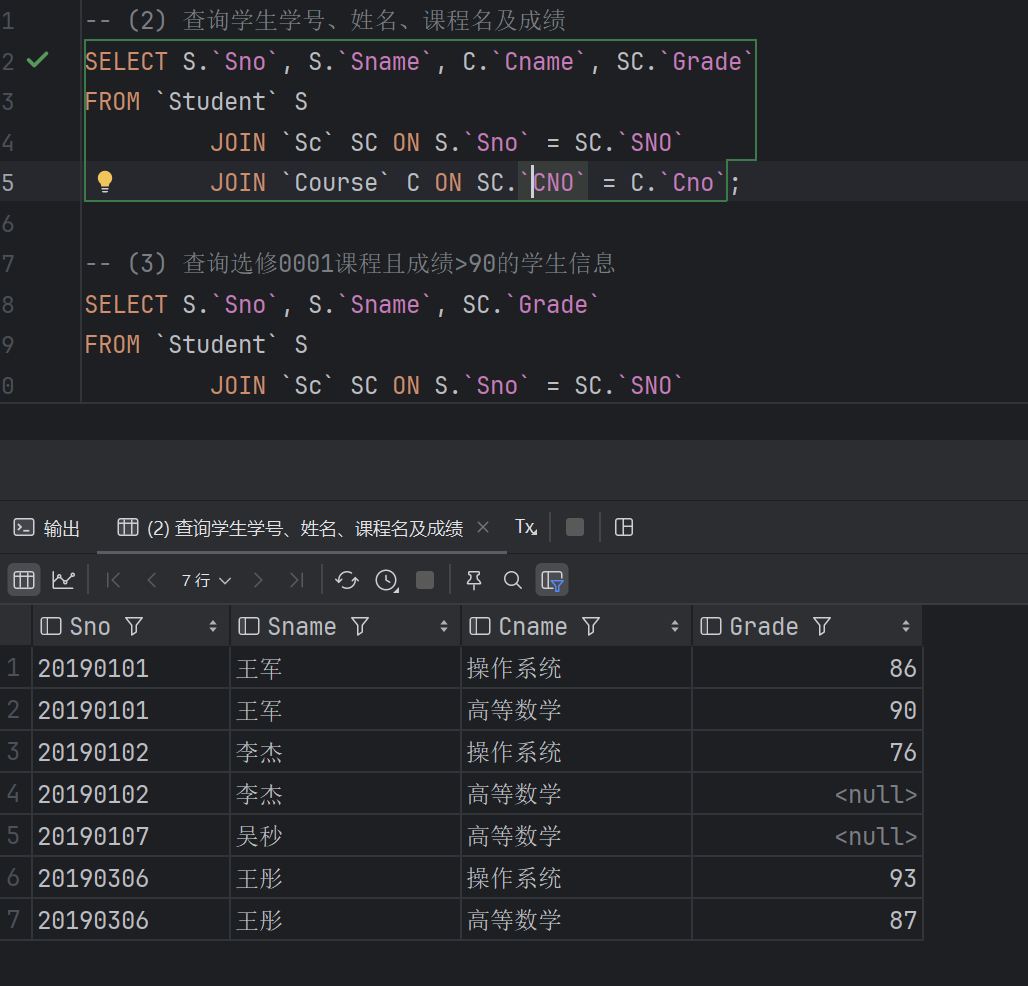
**9.连接查询操作**

对jwgl数据库实现以下查询：

1. 查询每个学生的情况以及他（她）所选修的课程；



1. 求学生的学号、姓名、选修的课程名及成绩；



1. 求选修0001 课程且成绩在90 分以上的学生学号、姓名及成绩；

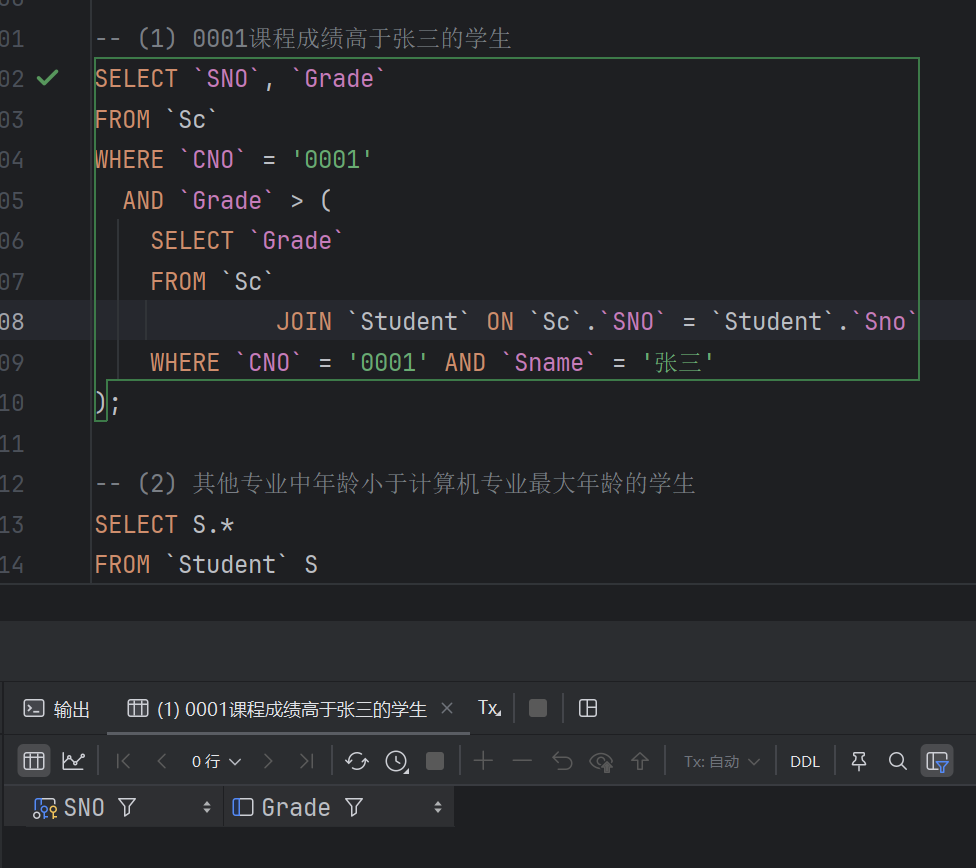


1. 查询每一门课的间接先行课。

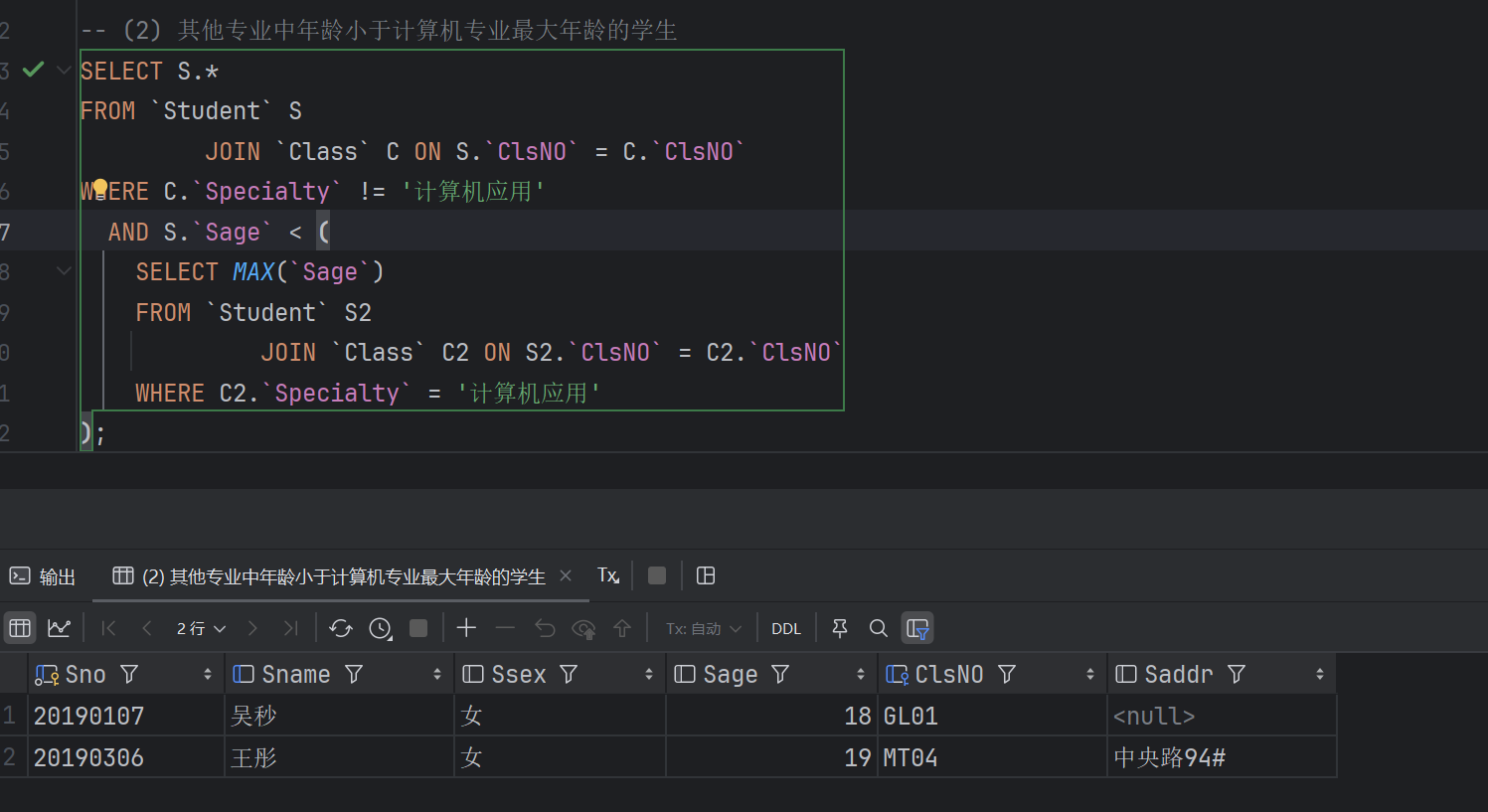


**10.嵌套和集合查询**

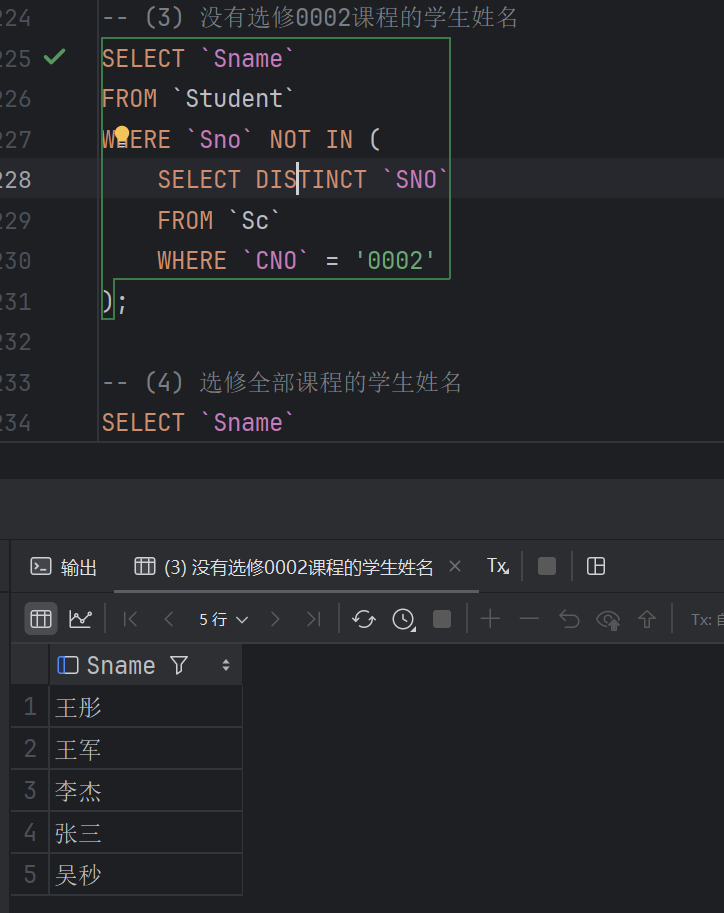
1. 求0001 课程的成绩高于张三的学生学号和成绩；



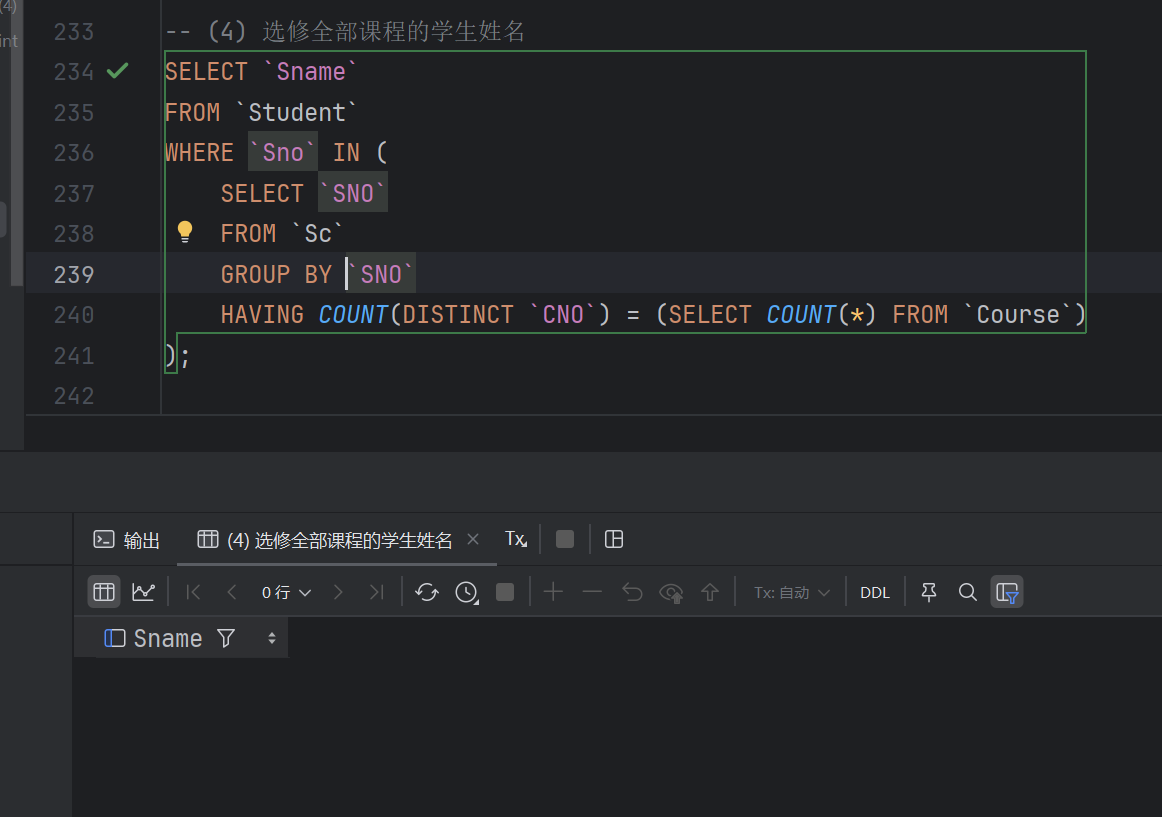
1. 求其他专业中比计算机应用专业某一学生年龄小的学生信息（即求其它专业中年龄小于计算机应用专业年龄最大者的学生）；



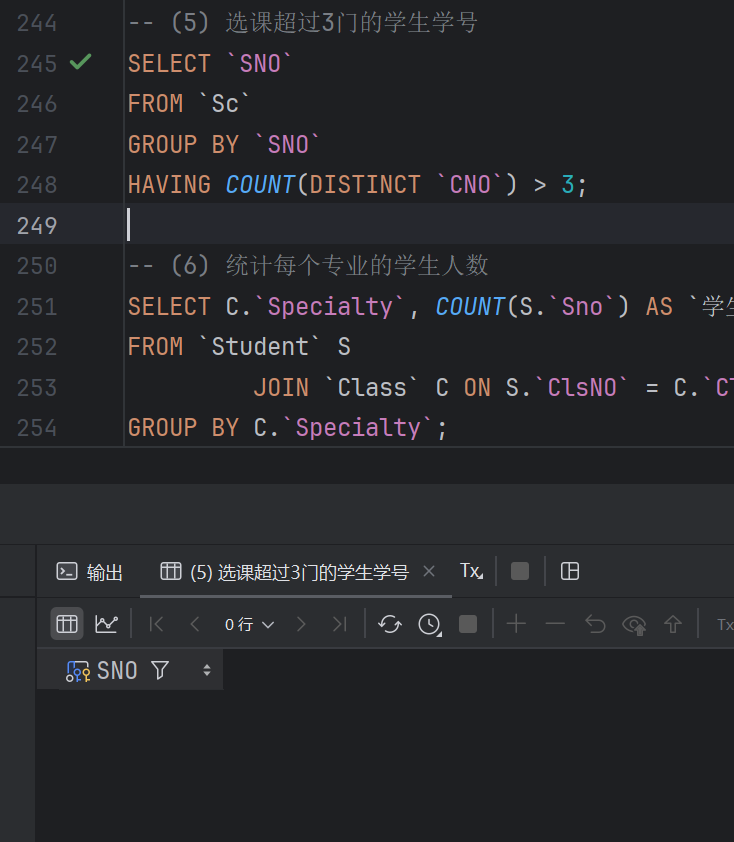
1. 求没有选修0002 课程的学生姓名；



1. 查询选修了全部课程的学生的姓名；



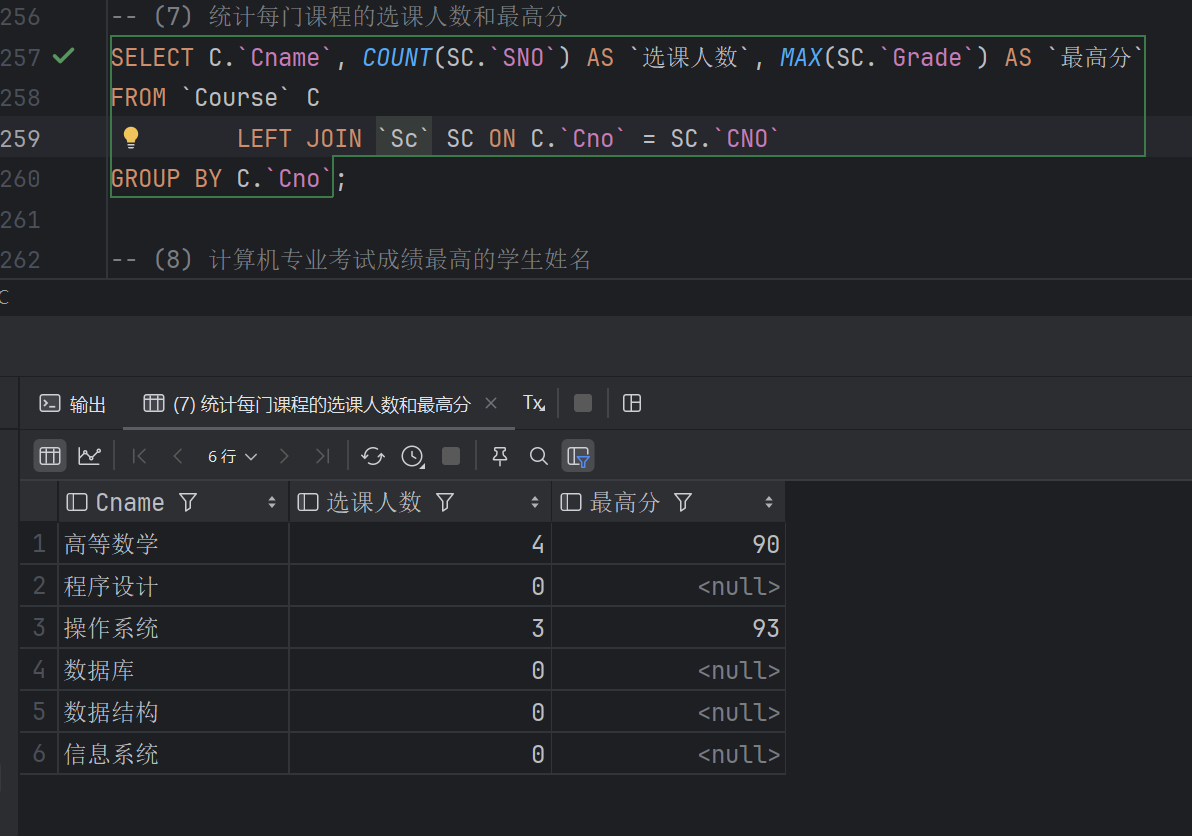
1. 求选修课超过3 门课的学生学号



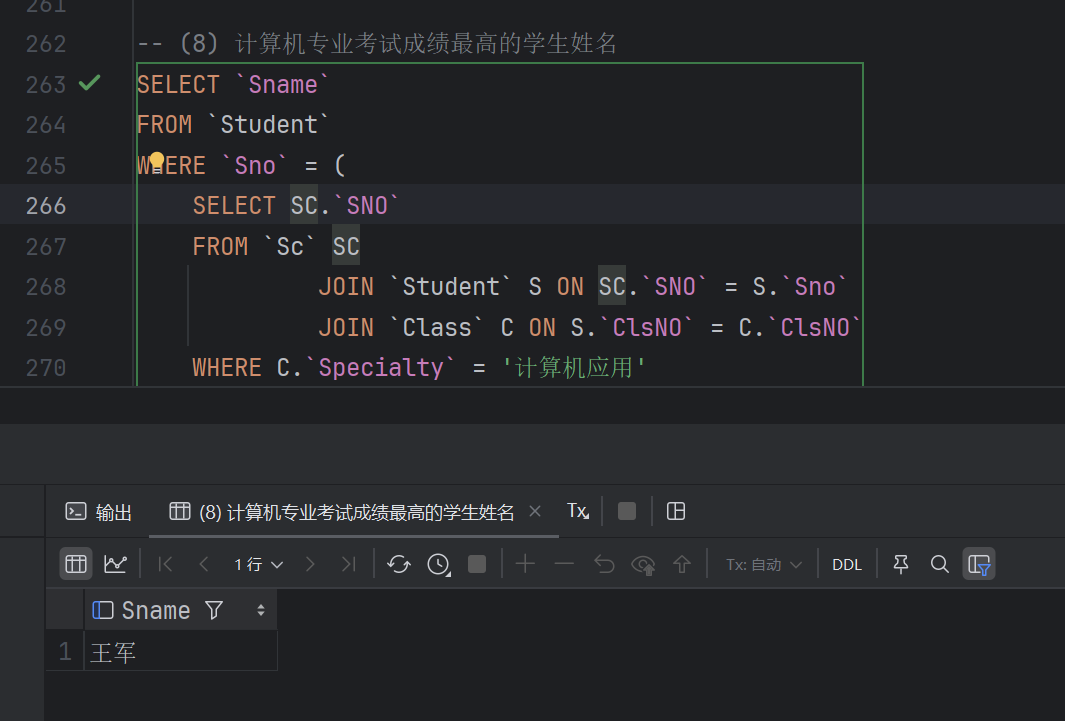
1. 统计每个专业的学生人数；



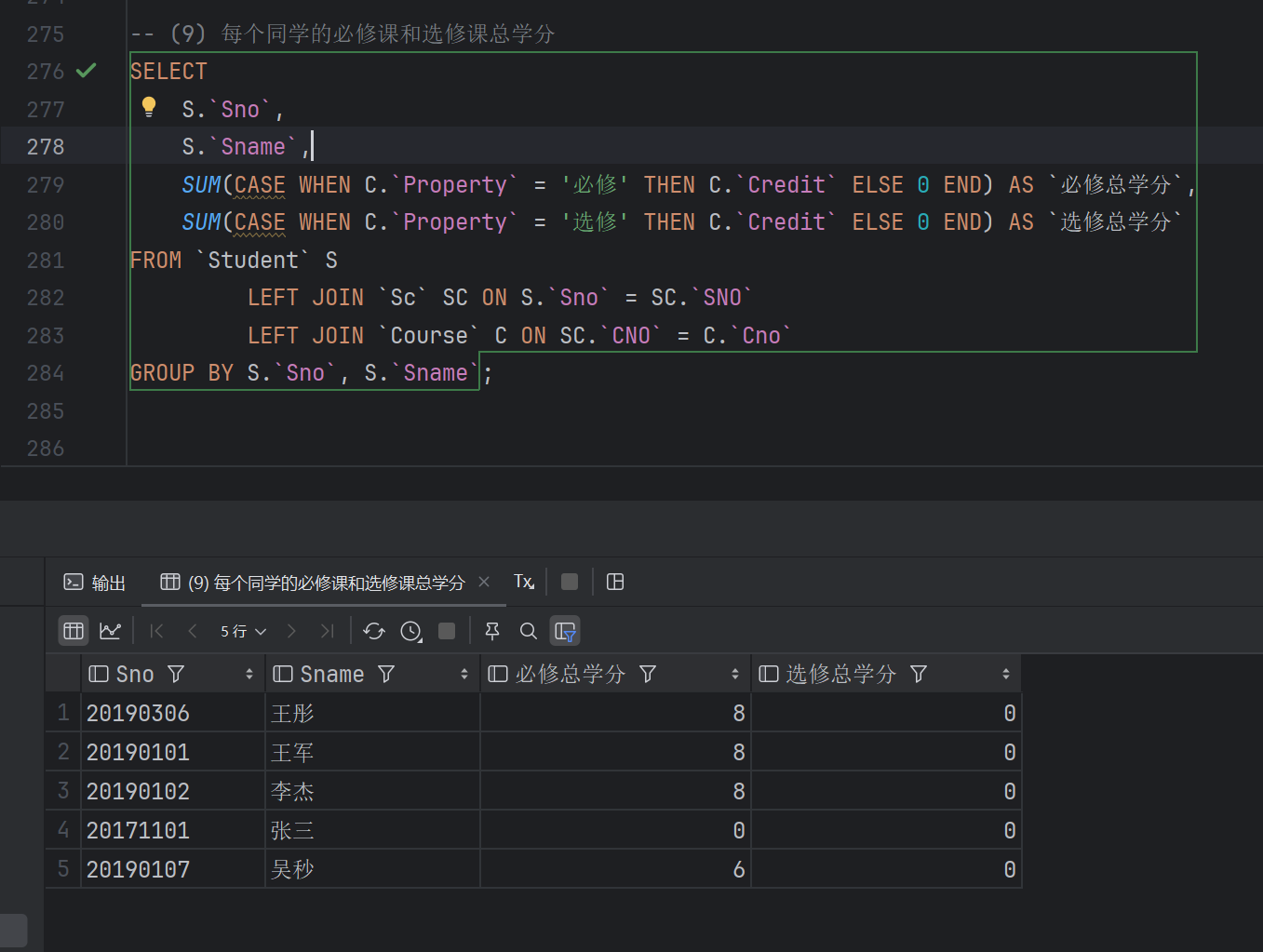
1. 统计每门课程的选课人数和考试最高分；



1. 查询计算机专业考试成绩最高的学生的姓名；

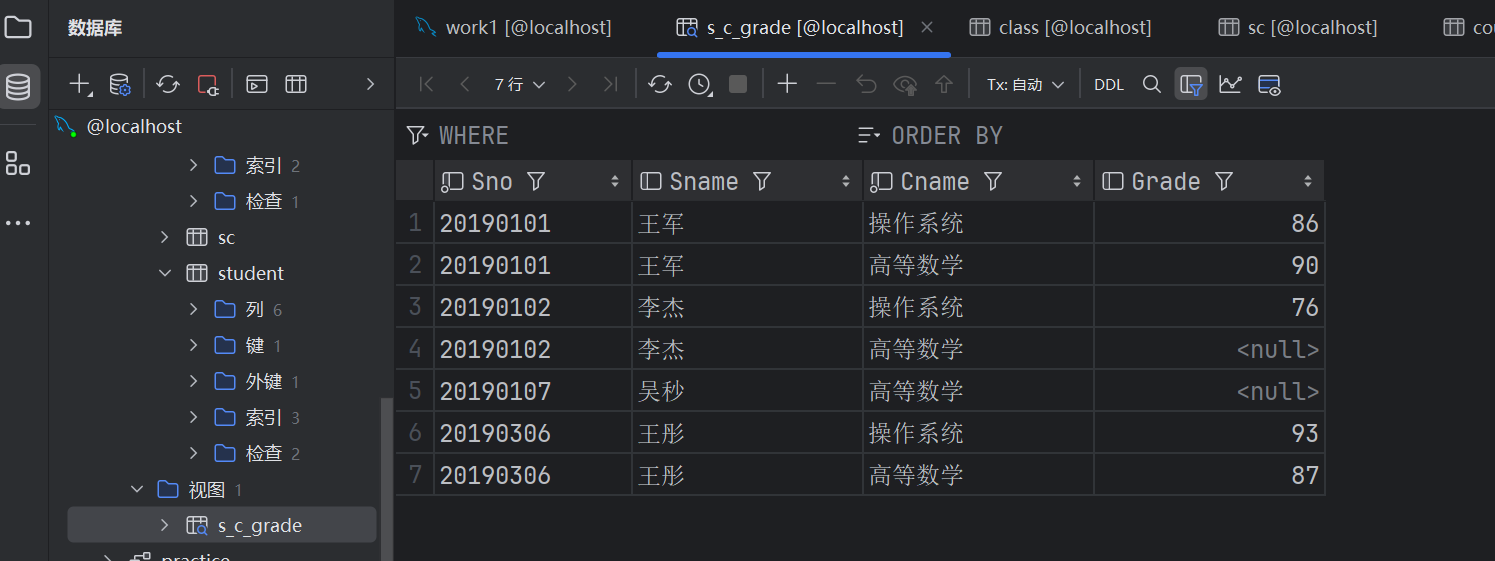


1. 查询每个同学的必修课总学分和选修课总学分；

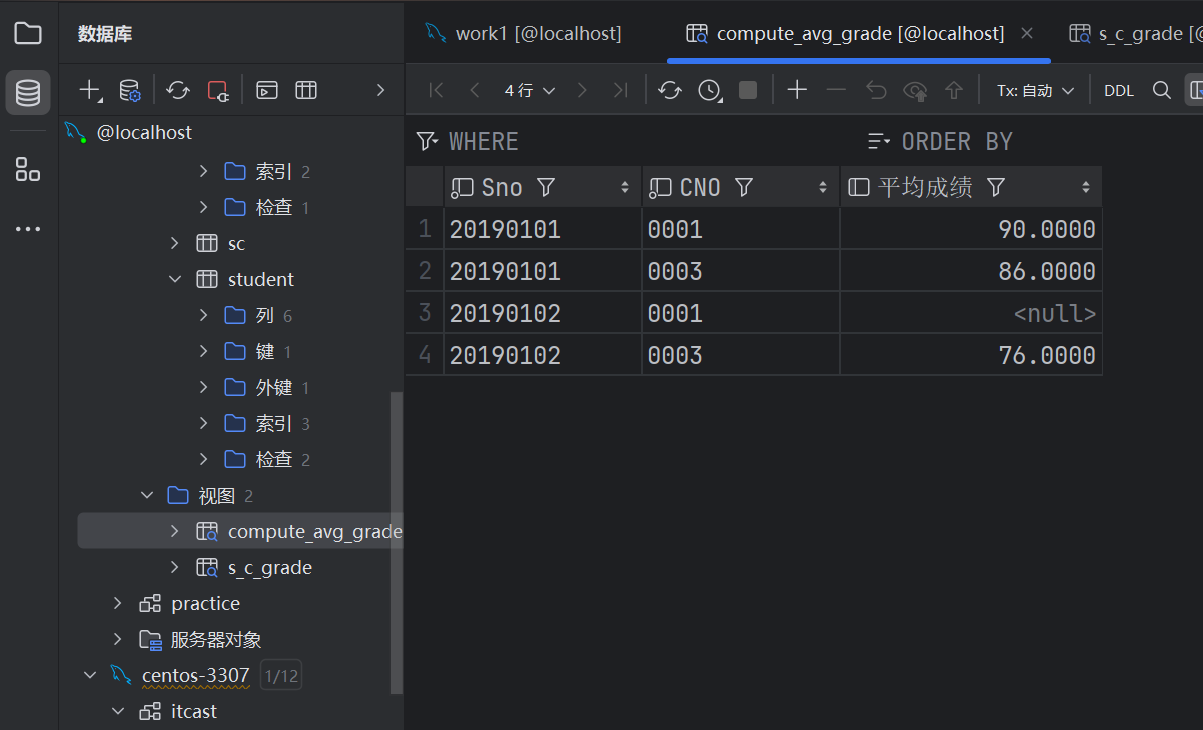


**11.视图**

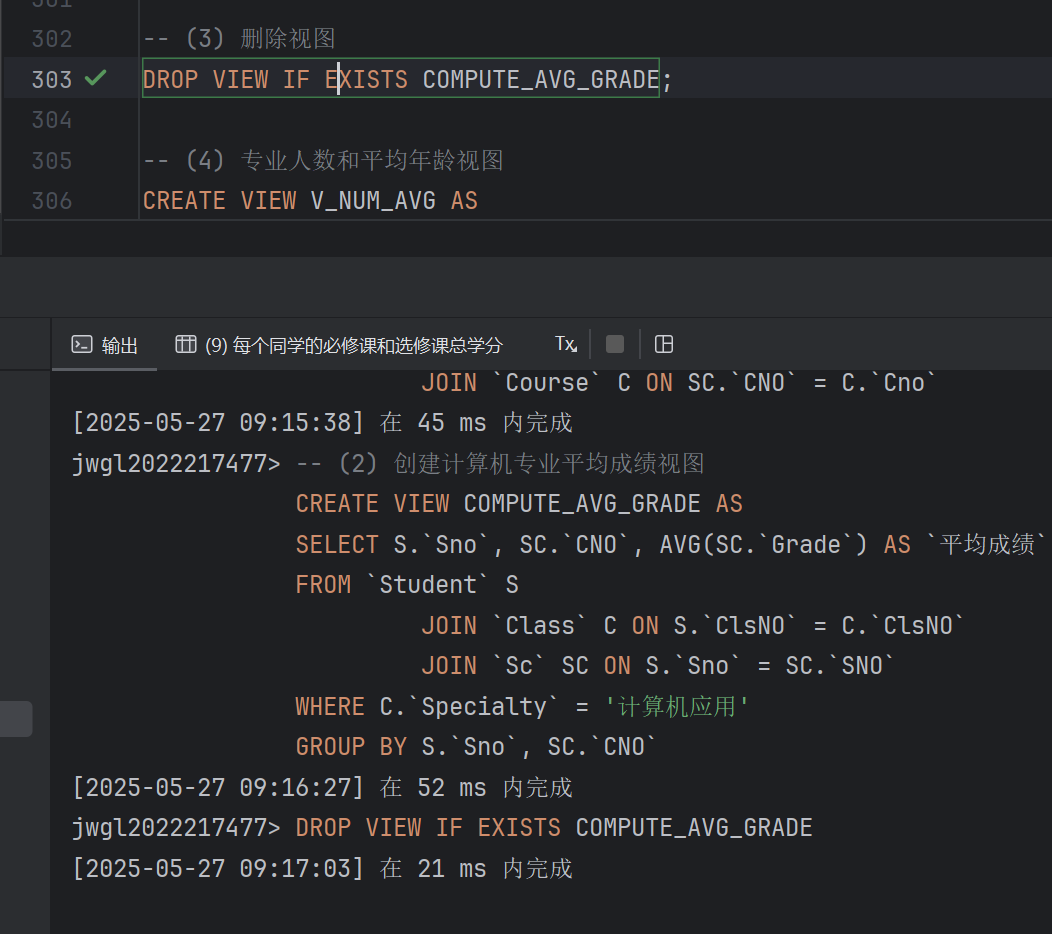
1. 使用SQL语句建立一个每个学生的学号、姓名、选修的课名及成绩的视图S\_C\_GRADE；



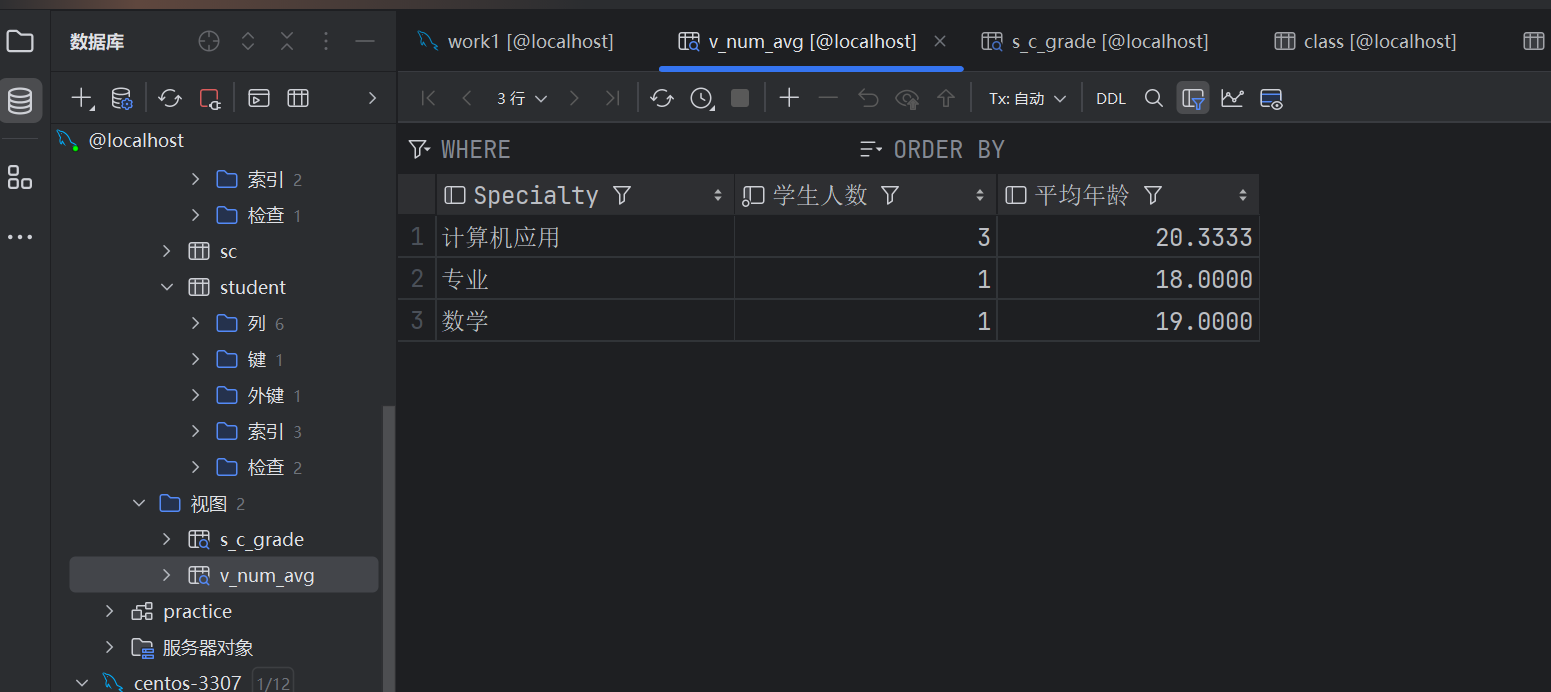
1. 建立一个所有计算机专业学生的学号、选修课程号以及平均成绩的视图COMPUTE\_AVG\_GRADE;



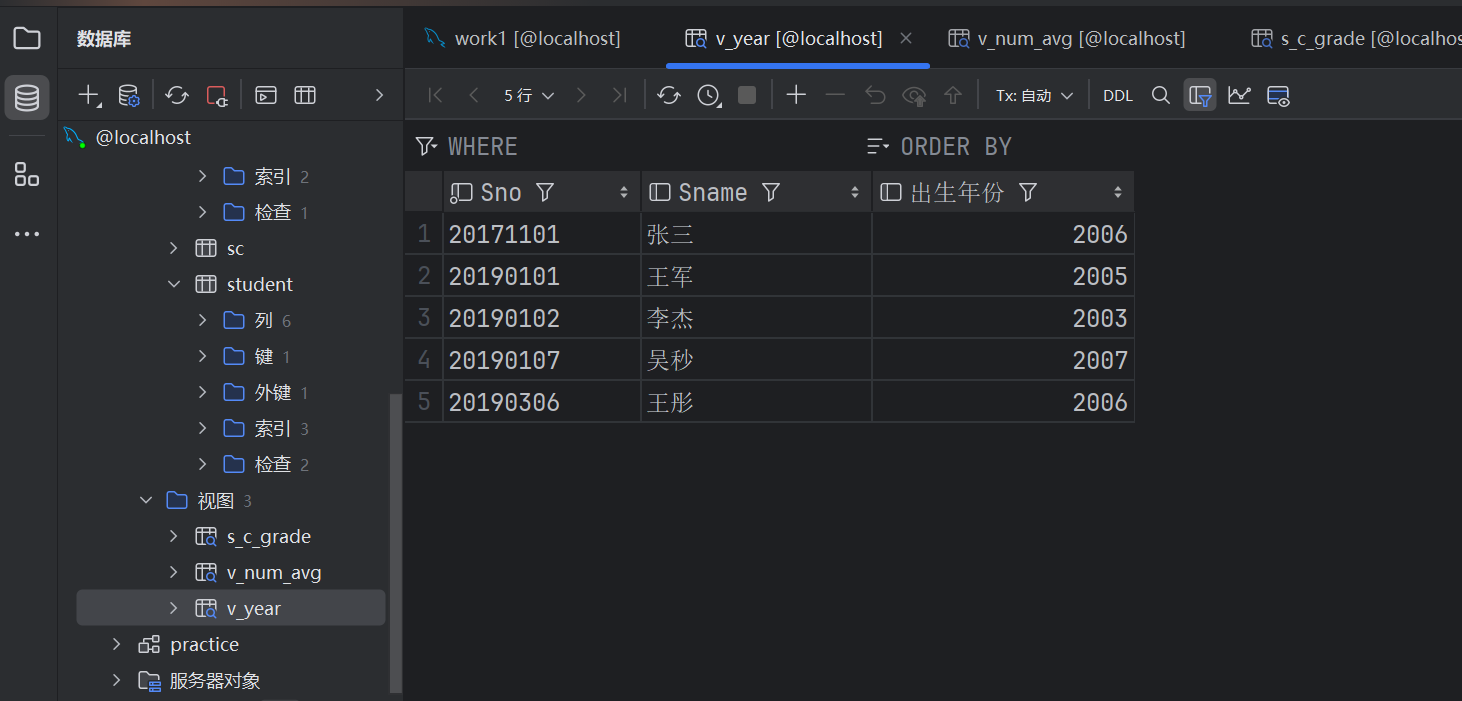
1. 用SQL语句删除视图COMPUTE\_AVG\_GRADE;



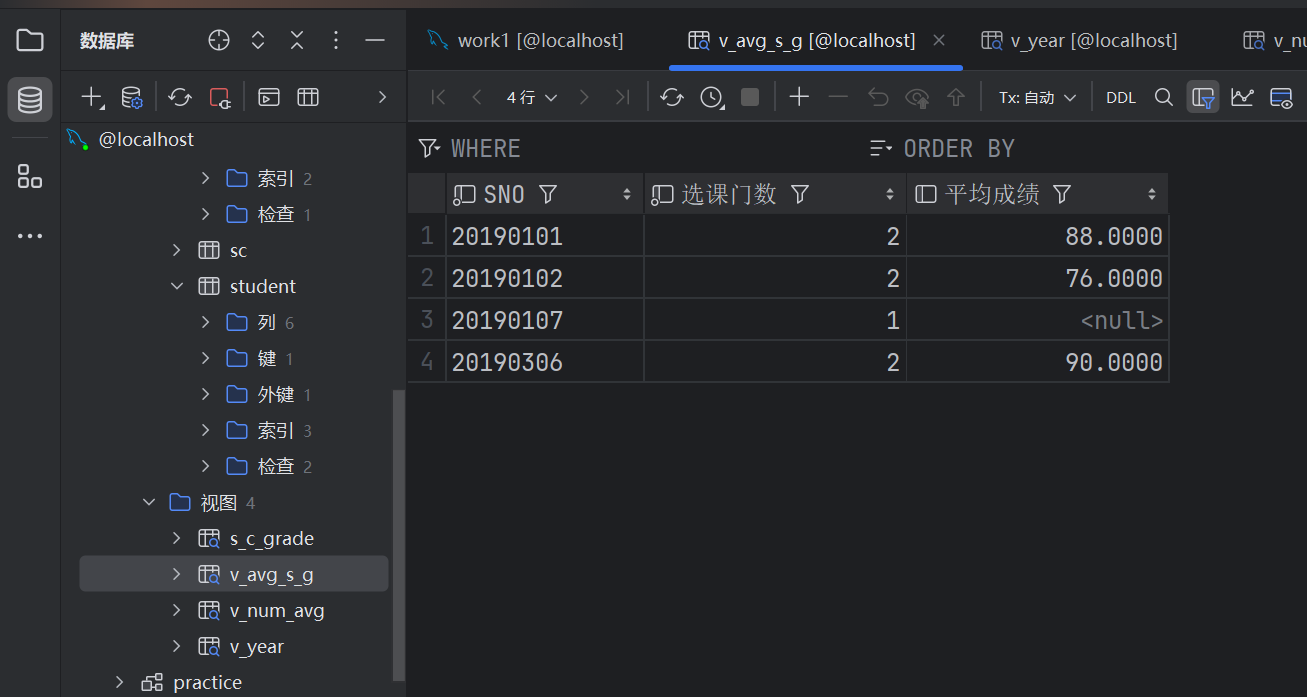
1. 将各专业学生人数，平均年龄定义为视图V\_NUM\_AVG



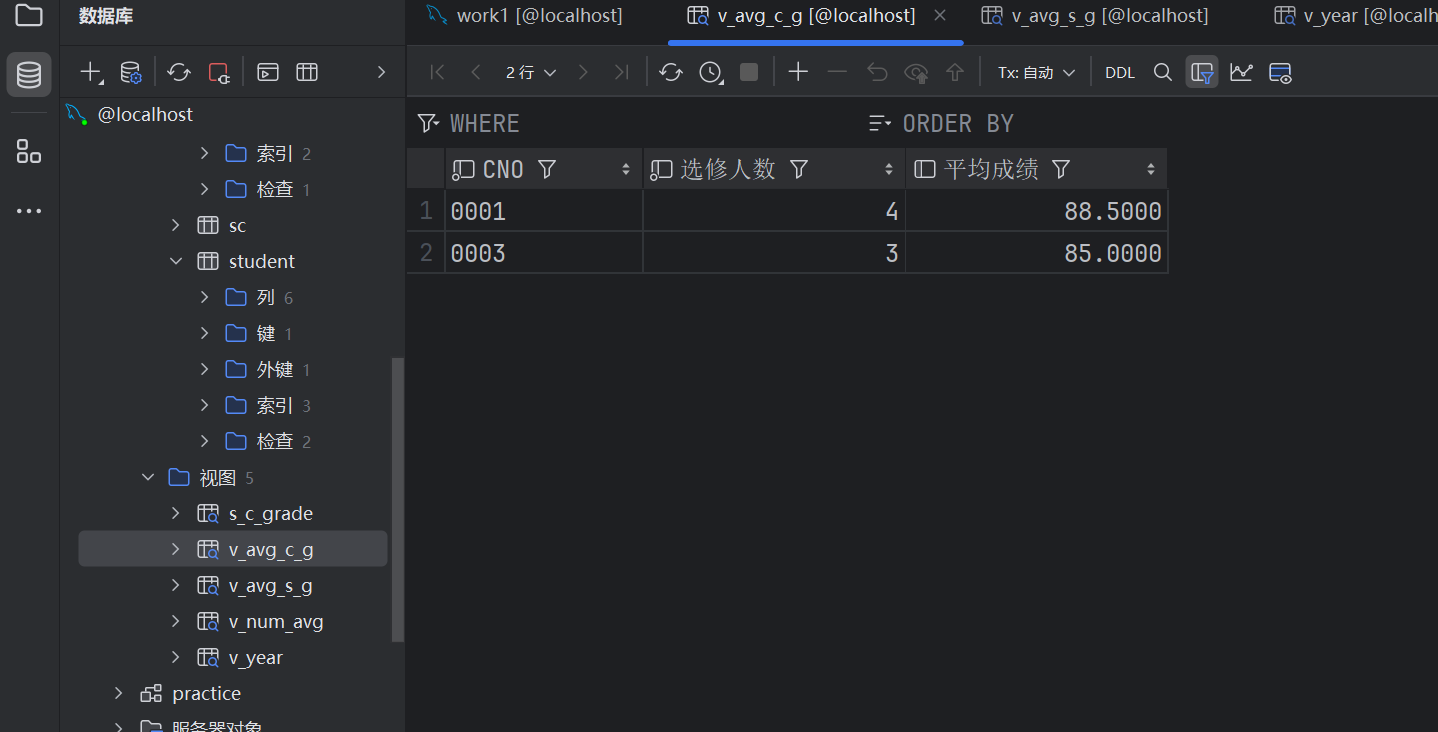
1. 定义一个反映学生出生年份的视图V\_YEAR



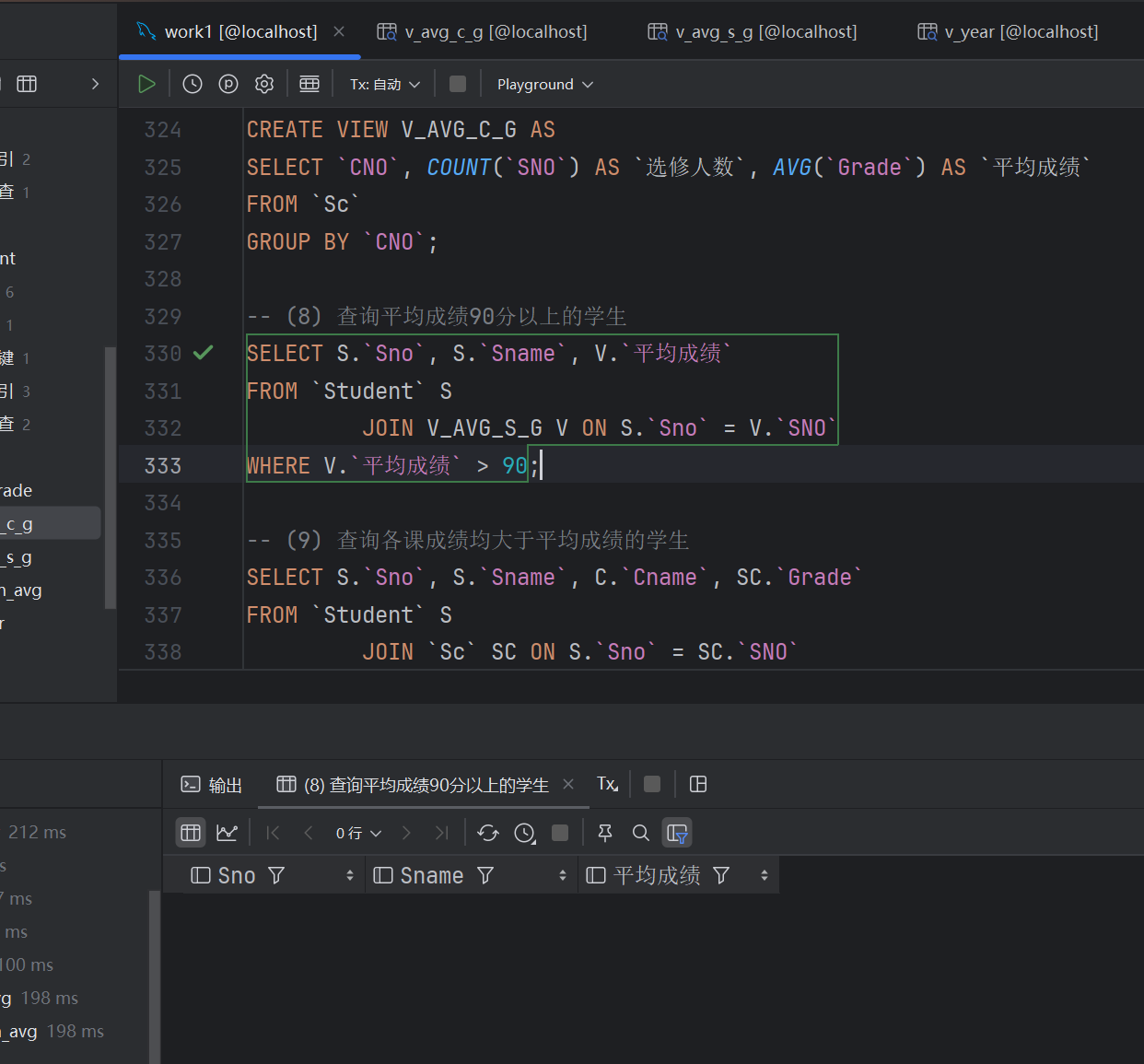
1. 将各位学生选修课程的门数及平均成绩定义为视图V\_AVG\_S\_G



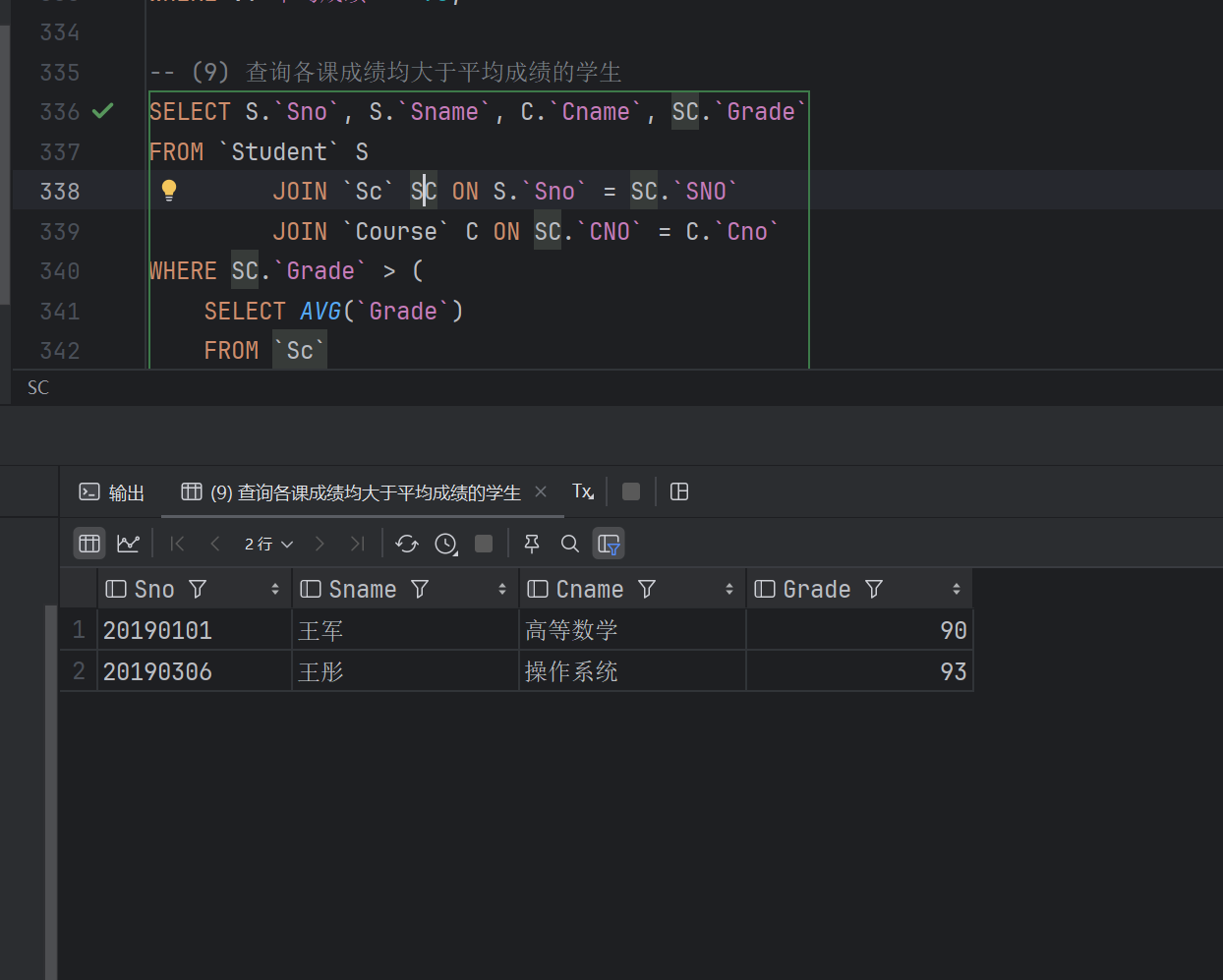
1. 将各门课程的选修人数及平均成绩定义为视图V\_AVG\_C\_G



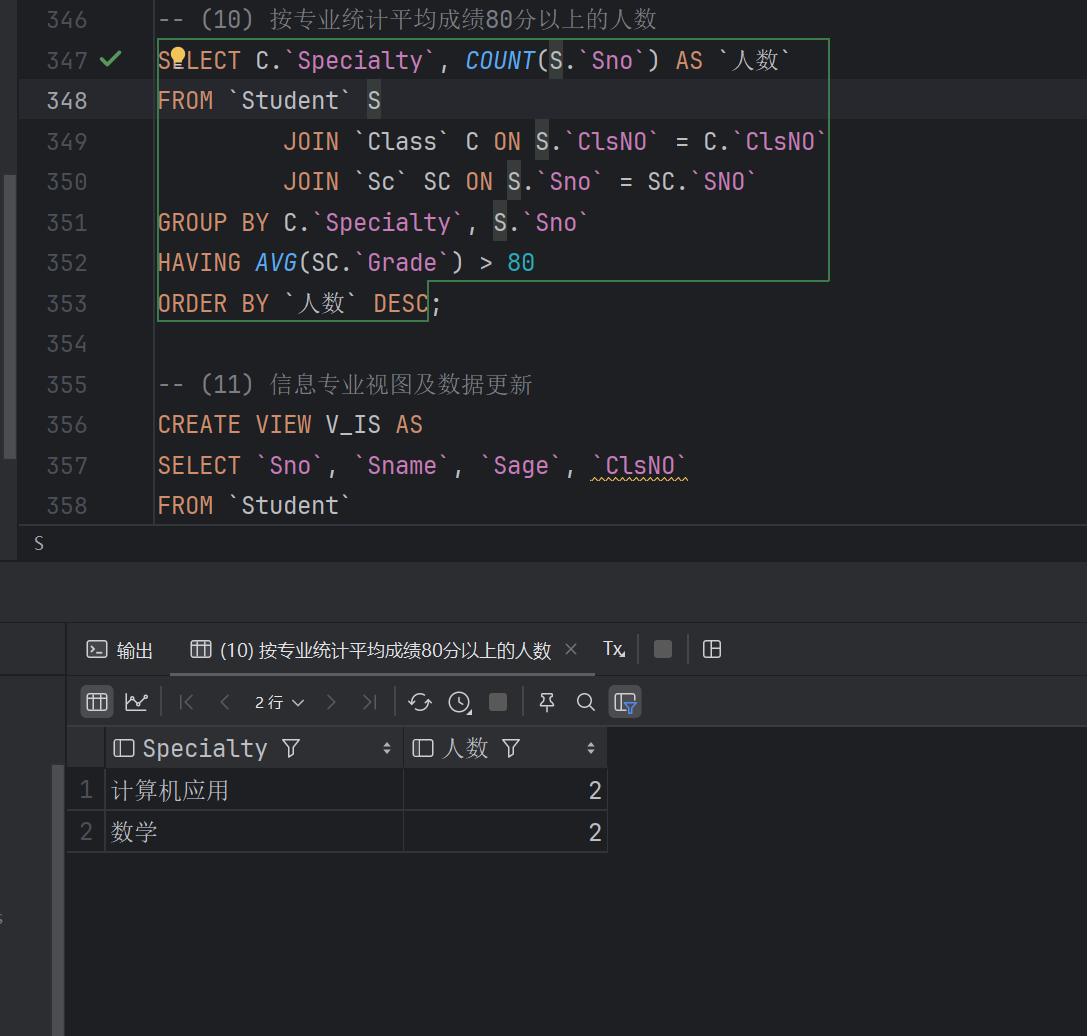
1. 查询平均成绩为90分以上的学生学号、姓名和成绩；



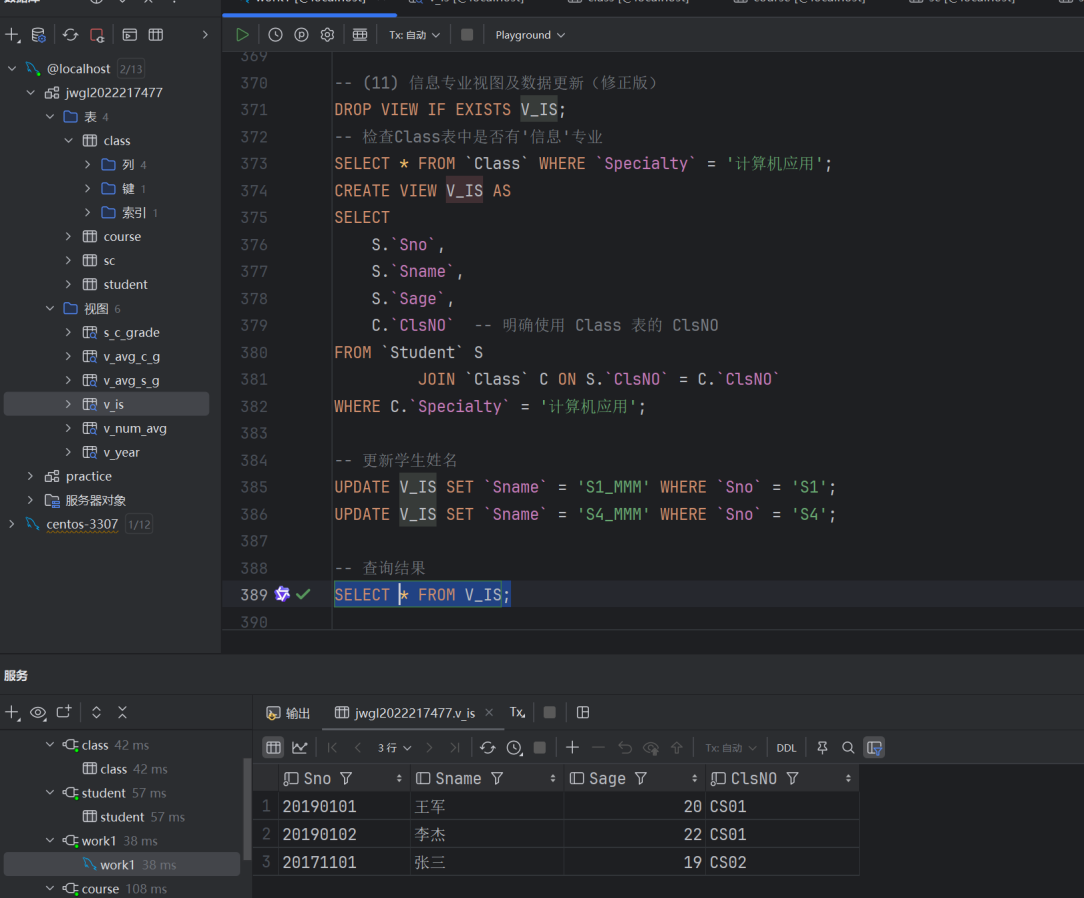
1. 查询各课成绩均大于平均成绩的学生学号、姓名、课程和成绩；



1. 按专业统计各专业平均成绩在80分以上的人数，结果按降序排列；、



1. 定义数学专业学生基本情况视图V\_IS。通过视图V\_IS，分别将学号为“S1”和“S4”的学生姓名更改为“S1\_MMM”,”S4\_MMM” 并查询结果;



1. 通过视图V\_IS，新增加一个学生记录 ('S12','YAN XI',19,'IS')，并查询结果

你所创建的视图 V\_IS 是通过连接 Student 表和 Class 表得到的。

尝试向这个视图插入数据时，数据库需要把插入操作映射到对应的基表上。不过，插入的数据里包含了来自两个不同基表的列：

Sno、Sname、Sage属于Student 表

ClsNO属于Class 表

大多数数据库不支持通过连接视图一次性向多个基表插入数据，所以会报错。

1. 要通过视图V\_AVG\_S\_G，将学号为“S1”的平均成绩改为90分，是否可以实现？并说明原因

不可以实现，原因视图包含聚合函数和GROUP BY，属于不可更新视图。

# 实验二 触发器、存储过程

## 一、实验目的

1.掌握 SQL Server中的触发器的使用方法；

2.掌握存储过程的操作方法。

## 二、实验准备

1、了解触发器

2、了解存储过程

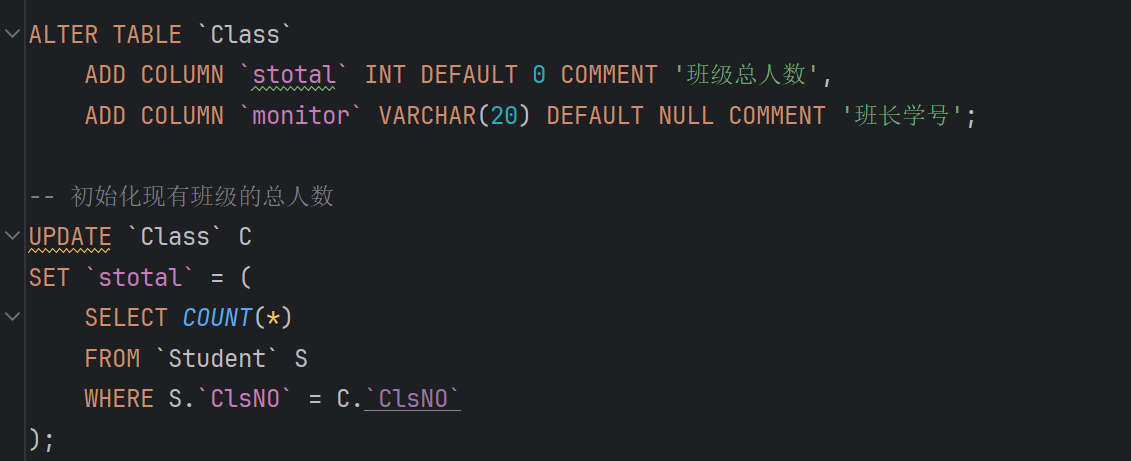
## 三、实验要求

1. 在实验之前做好准备

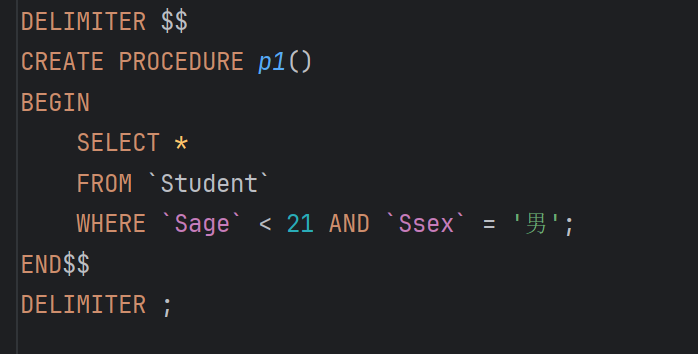
2. 试验之后提交实验报告，并验收实验结果

## 四、实验内容

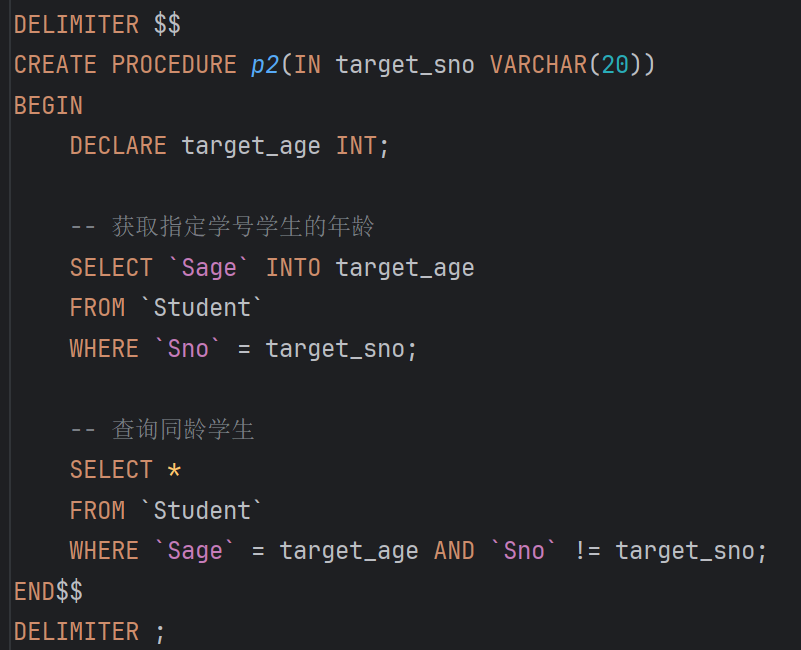
1.在班级表class中增加总人数（stotal）和班长两个字段；



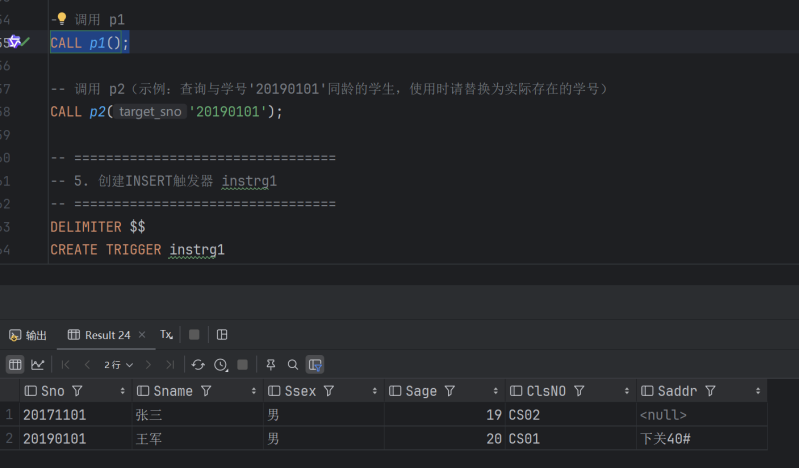
2.建立不带参数的存储过程p1（选择所有年龄<21岁的男同学）；



3.建立带有参数的存储过程p2（根据用户传递的参数只显示与指定学号的学生同龄的所有同学）；



4.使用Transact-SQL分别调用p1、p2，并将结果显示出来；

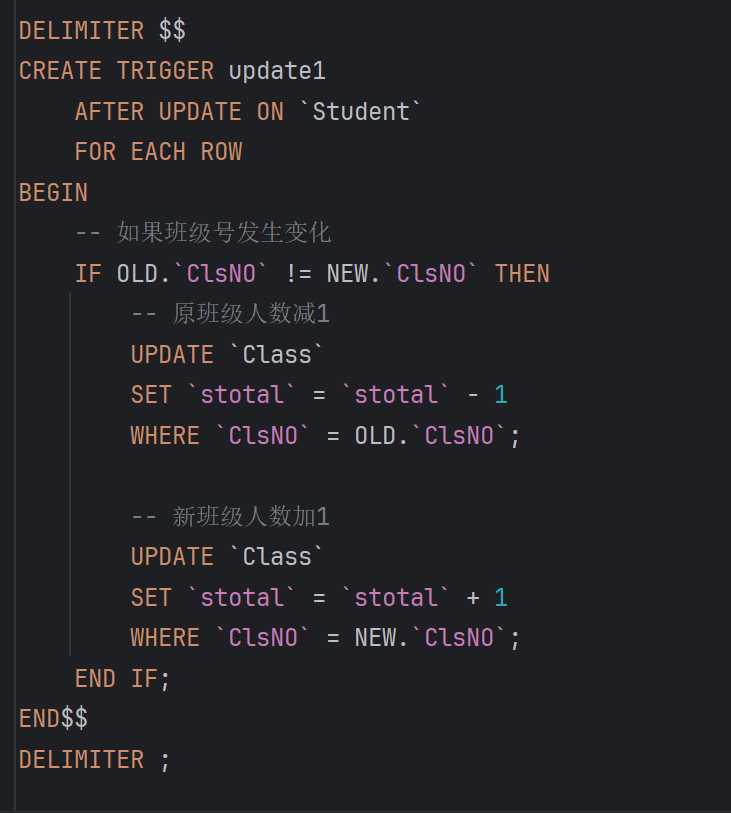


1. 创建INSERT触发器instrg1：在学生表中每新增一名学生，将在班级表class表的总人数自动增加1。



6.创建UPDATE触发器update1:当某学生所在班号发生变化时（即调到另一班级后），

将其原先所在班级总人数减1，将新调入的班级表class表的总人数增加1。



## 五、实验总结

触发器作为数据库自动化机制，通过事件驱动实现数据一致性维护。本次实践中，INSERT触发器自动更新新增学生的班级人数，UPDATE触发器处理转班时的跨班级人数同步，体现了其在业务逻辑自动化中的核心价值。在数据变更时即时执行规则，避免手动操作遗漏（如转班时自动调整两班人数）。将规则嵌入数据库层，无论数据通过何种渠道变更，均可强制执行（如班级人数联动）。通过引用变更前后的数据，精准定位操作对象（如比较新旧班级号判断是否调整人数）。

触发器实现了 “数据变更 - 规则执行” 的无缝衔接，简化应用层逻辑的同时保障数据准确性。但需注意：复杂触发器可能影响性能，需平衡自动化与效率；其隐式执行特性要求在设计时预留调试空间，确保系统可控性。

# 实验三 数据库的安全性、备份与恢复

## 一、实验目的

1.掌握 Navicat for mysql 的安全性机制；

2.掌握Navicat for mysql安全性、备份和恢复的操作方法。

## 二、实验准备

了解数据库的安全性、备份与恢复

## 三、实验要求

1. 在实验之前做好准备

2.可视化设置数据库用户及操作权限；进行数据库的备份和恢复；

3.利用T-SQL语句进行备份和恢复操作。

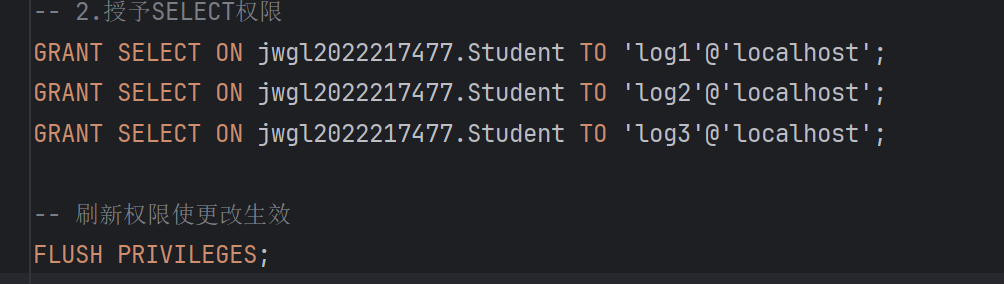
4. 试验之后提交实验报告。

## 四、实验内容

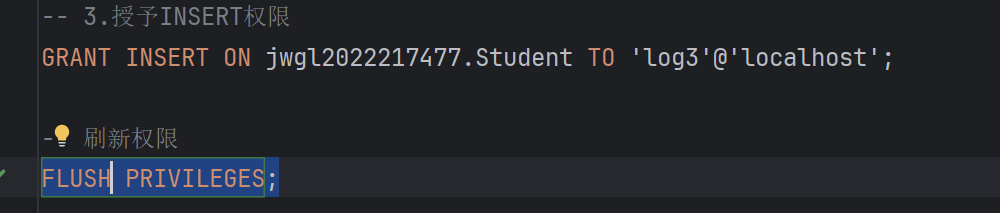
1. 建立登陆帐户log1、log2、log3,并将它们映射为jwgl中的用户



1. 为log1、log2、log3授予对Student表的查询权



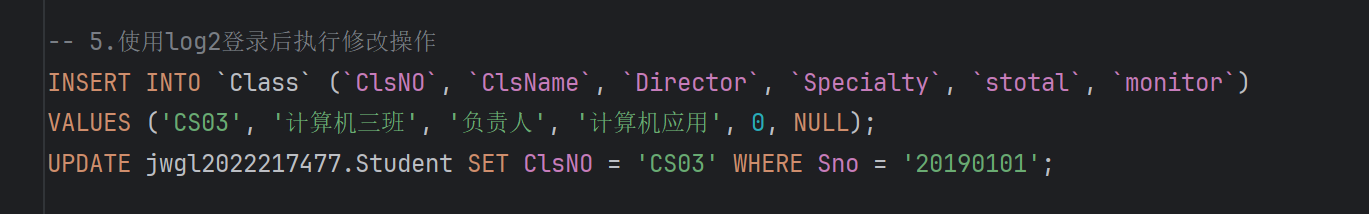
1. 在查询编辑器中使用SQL语句为log3授予对student表的插入权

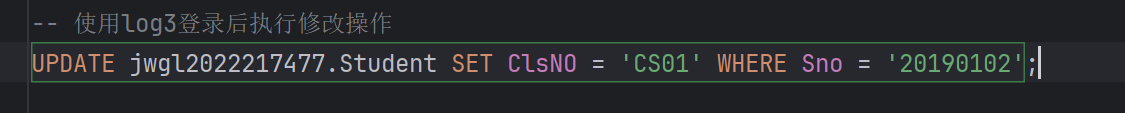


1. 建立jwgl数据库的用户角色ROLE1, 并将log1、log2、添加到此角色中，为此角色授予对student表的修改权



1. 用log2登录，并修改student中某个学生的”所在专业“的值。如果用log3登录，执行相同的操作，结果如何？



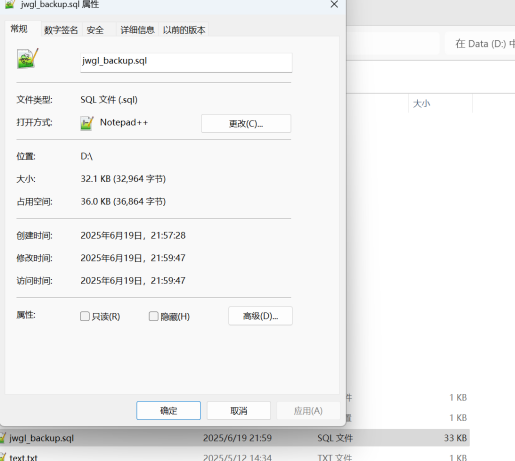


结果：更新为新值

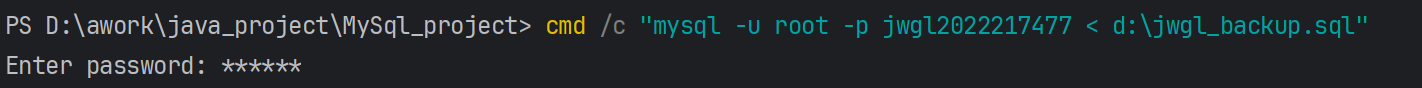
1. 将jwgl数据库完整备份到某个路径下。

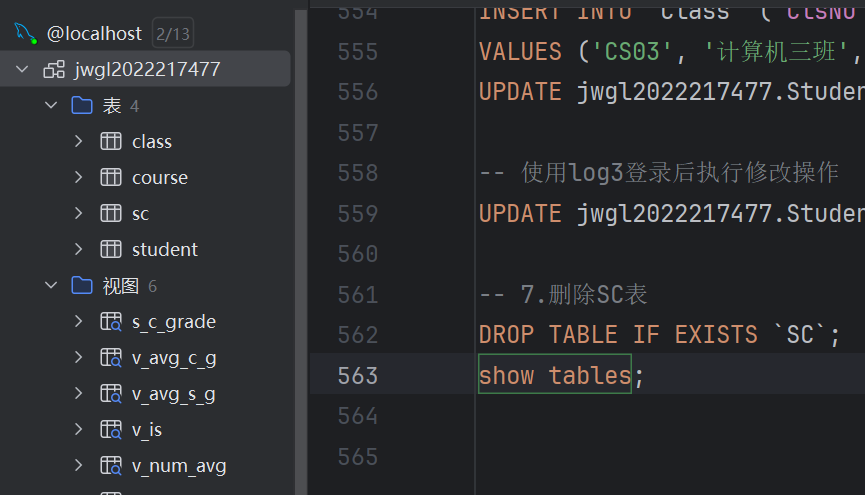
**注（举例）**：mysqldump -u root -p database\_name d:\db.bak

恢复: mysql -u root -p database\_name d:\db.bak

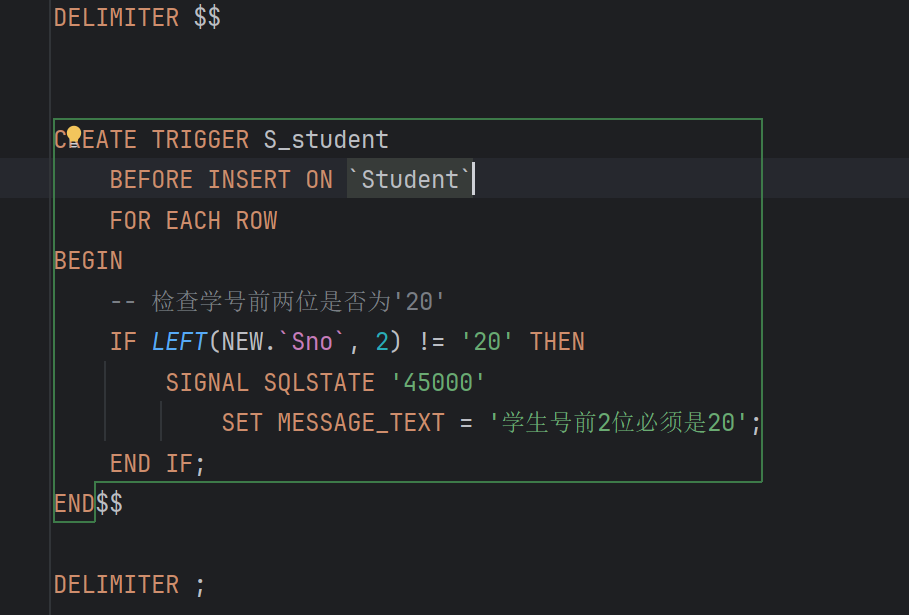


1. 删除SC表, 利用备份文件进行恢复

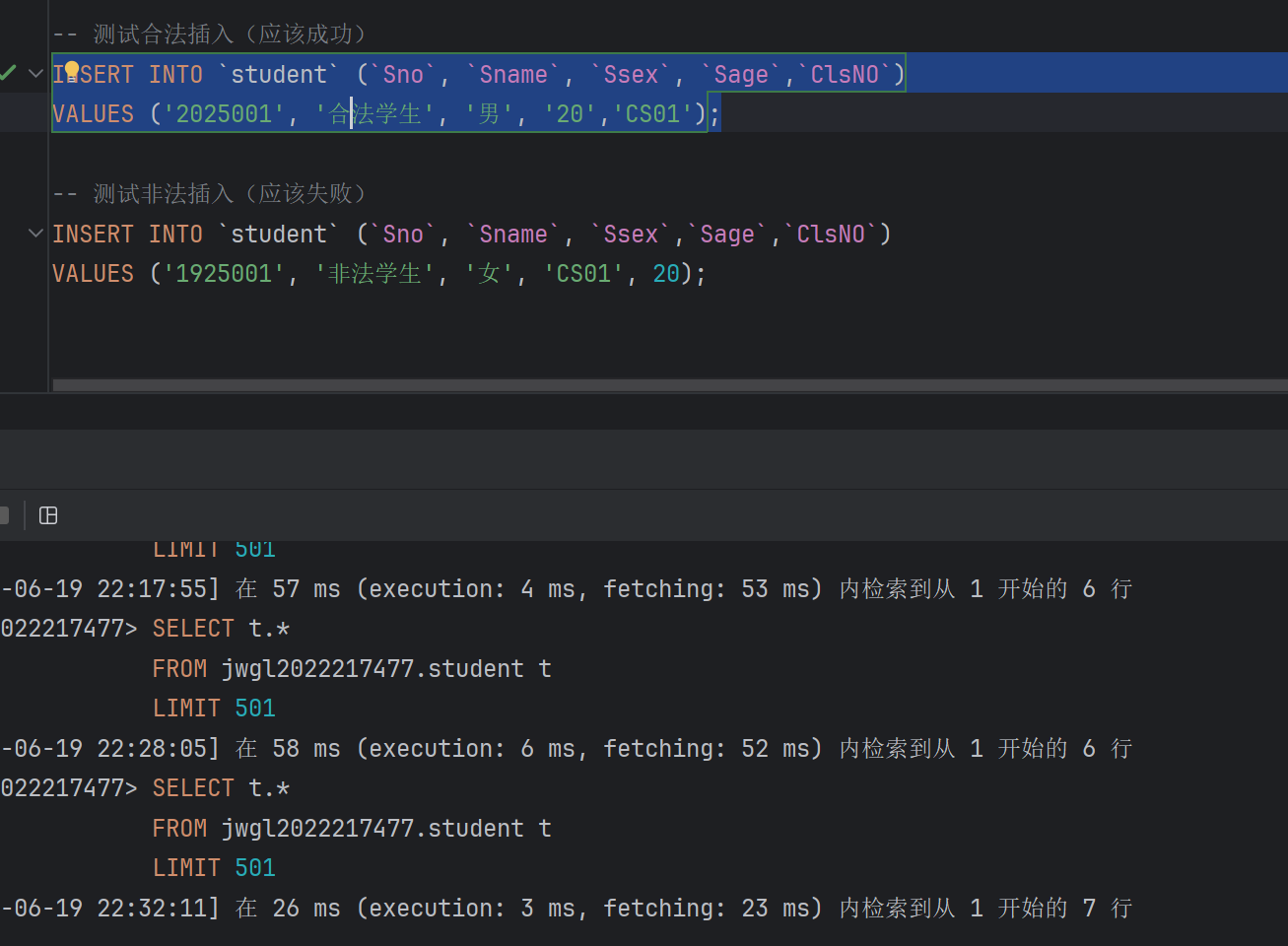


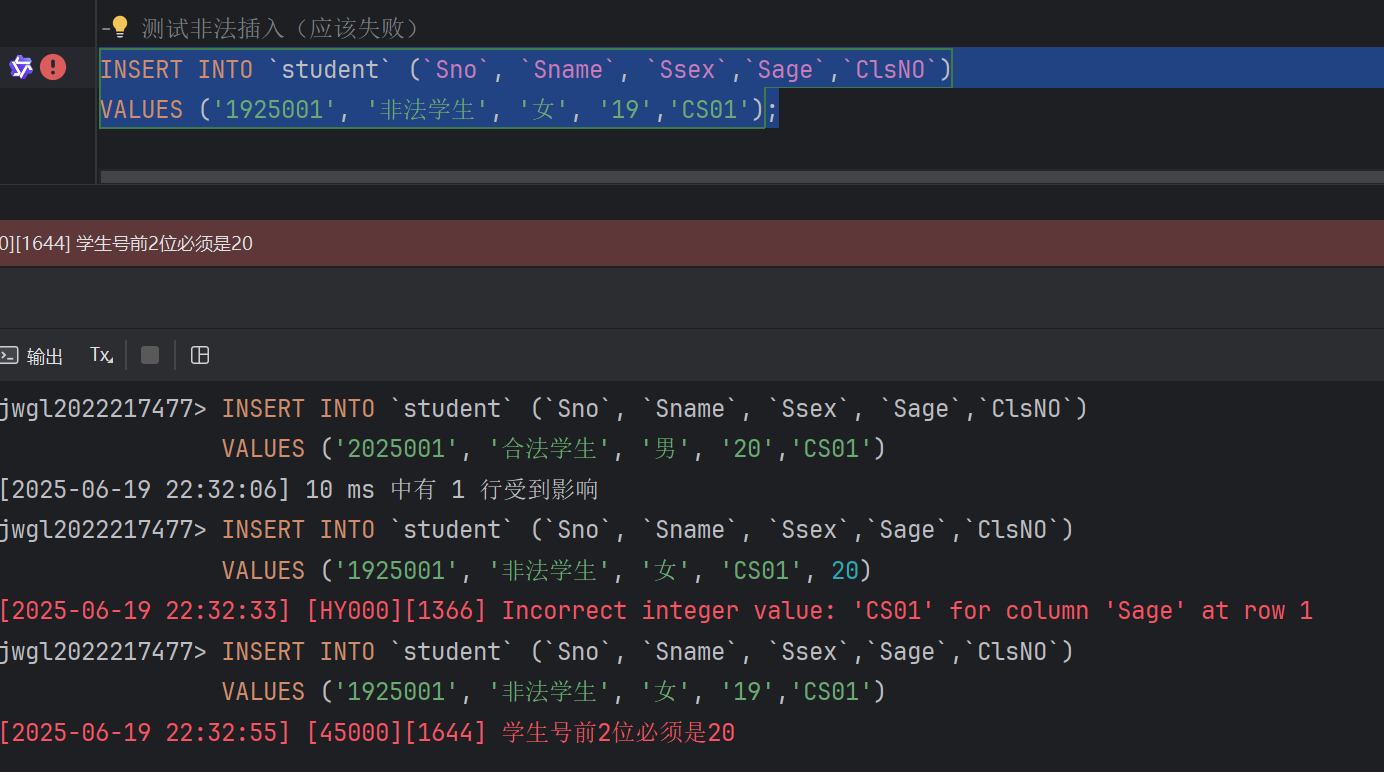


1. 创建一个作用在Stuent表上的的触发器 S\_student，确保向Student表中插入一记录时，如果学生号的前2位不等于“20”，则显示信息“学生号前2位必须是20”，并不允许插入。



1. 测试触发器是否能成功执行。





## 五、实验总结

数据库实验总结与感悟 本次实验围绕MySQL数据库操作展开，涵盖表结构设计、数据增删改查、多表连接、视图创建等核心内容。通过完成表结构修改、数据插入更新、复杂查询等任务，深入理解了关系型数据库的完整性约束与SQL编程逻辑。

实验中多次遇到外键约束失败、自引用更新错误等问题，例如插入课程表时因先修课程编号不存在导致的外键冲突，以及更新操作中因子查询直接引用目标表触发的MySQL限制。通过调整数据插入顺序、使用临时变量存储中间结果等方式，逐步解决了各类异常，深刻体会到数据库操作中数据一致性与语法规则的重要性。

在视图操作部分，明确了视图作为查询封装的特性：简单视图支持数据更新，而包含聚合函数或分组操作的视图仅能用于查询。这一实践强化了对数据库视图机制的理解，也认识到视图在简化复杂查询、提升代码复用性方面的价值。

触发器作为数据库自动化机制，通过事件驱动实现数据一致性维护。本次实践中，INSERT触发器自动更新新增学生的班级人数，UPDATE触发器处理转班时的跨班级人数同步，体现了其在业务逻辑自动化中的核心价值。在数据变更时即时执行规则，避免手动操作遗漏（如转班时自动调整两班人数）。将规则嵌入数据库层，无论数据通过何种渠道变更，均可强制执行（如班级人数联动）。通过引用变更前后的数据，精准定位操作对象（如比较新旧班级号判断是否调整人数）。

看似简单的mysqldump命令，在 PowerShell 环境中因重定向符号不兼容而报错。这启示我：技术文档需结合实际环境（如操作系统、数据库版本），且任何自动化操作（如备份脚本）都应定期演练，避免在紧急恢复时手忙脚乱。

本次实验不仅深化了对数据库原理的理解，更培养了严谨的技术思维。通过解决实际问题，我认识到，一个字段名的拼写错误或权限配置的细微差异，都可能导致系统崩溃。数据库技术（如分区表、GTID 复制）不断演进，需保持学习热情。