特别提醒：这就是实验报告，请下载后直接在本报告中进行编辑，只可以要求的添加实验截图或数据，不能删除原来的内容！！！完成后需要将本实验报告提交回课程平台！！！

《数据库原理与应用A》实验9

专业班级：信科202003 姓名：刘梓淳 学号：202003919

1. 【实验目的】

* 理解数据库的安全保护
* 理解数据库的完整性保护

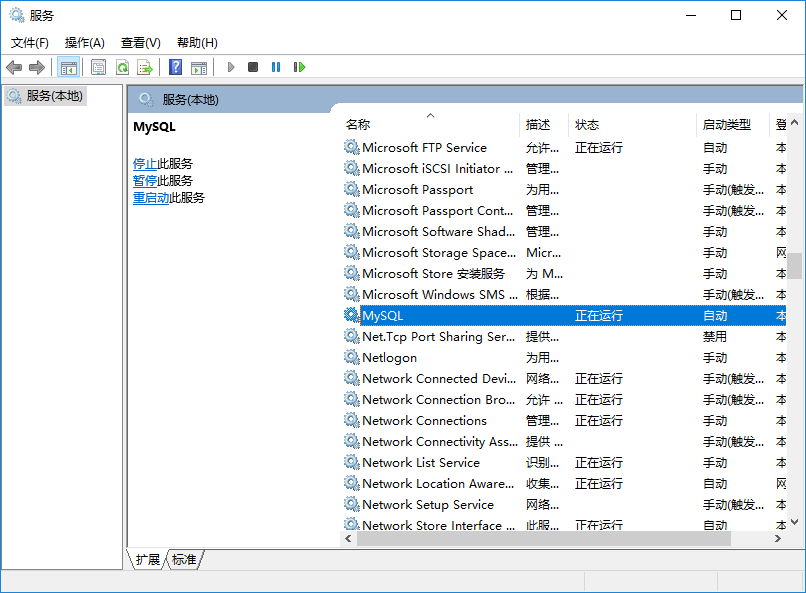
1. 【实验内容】
2. **实验准备**
3. 实验环境

操作系统：win7中文旗舰版

DBMS：MySQL 5.5或者SQLyog

1. 检查本机是否安装且启动MySQL服务

打开windows的服务（有N种方法可以打开它，最简单最快捷的方式就是按win键打开“开始”菜单后直接输入“服务”就找到了）：



找到MySQL服务实例名称（默认就是“MySQL”，除非你在安装时自己命名过，那就得用你自己命名的名称了），观察其状态，如不是“正在运行”态，则利用工具按钮或右键菜单**启动它**。

1. 启动SQLyog

注：也可以使用Navicat等其他图形化管理工具，不过相应的命令或操作请自行查找使用，这里不做指导。

1. 准备实验数据库

在SQLyog中创建以自己的中文姓名为名的数据库（**请将下面数据库名称中的"雷净兰"改成你自己的姓名**）



打开你自己的数据库（**下面命令要将"雷净兰"改成你自己的数据库名称**）：

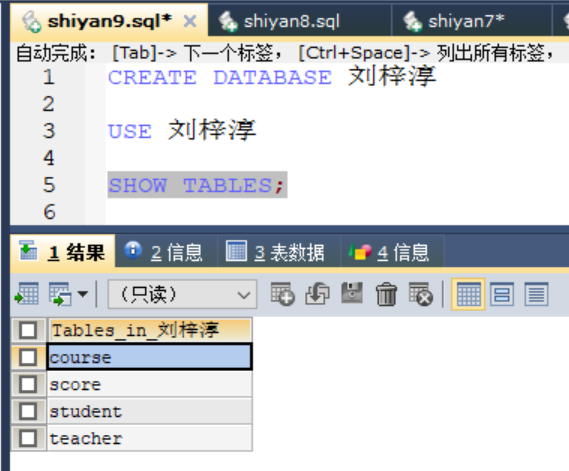


下载用于生成实验数据的[import\_xjgl\_data.sql](http://eol.sicau.edu.cn/Download?classid=300367434&teacherid=79077&resourceid=4691)脚本文件保存到本地D:\下。然后在SQLyog中导入实验数据，完成后再输入并执行如下命令查看导入的情况：



**将你show出来的结果截图放在下面：**

提示：你应该能看到有熟悉的4张表了



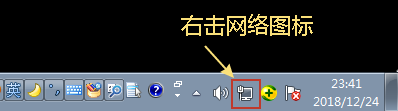
自行用select命令检查一下这4张表里是否有数据，如果没数据，说明前面导入数据的操作有错，请重新完成前面的装入数据操作。

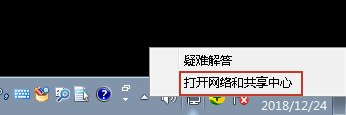
1. **用户鉴别实验**
2. 在实验室找一个搭档的同学（最好就近位置），本次实验需要两位同学搭档镜像操作，即双方要完成同样但互为对方的实验内容。

特别说明：如果确实找不到搭档，可以自已一人分担两个角色来完成所有的实验项目。

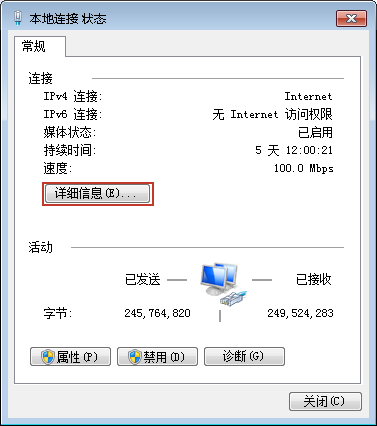
1. 查找出本机的IP地址。

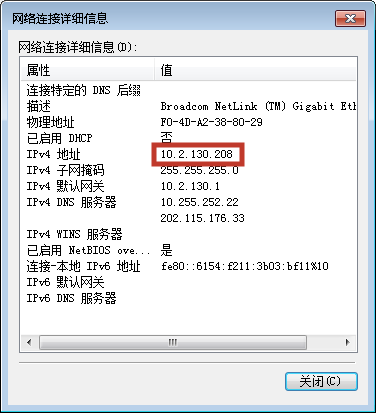
提示：会命令行的同学可以直接用命令查询出本机IP地址。











**将你IP地址的界面截图放在下面：**



1. 创建登录账户

在SQLyog（后面我们称为**管理器**）中为本机服务器新建一个名为“**你的同学的姓名全拼**”（例如雷雷的同学菁菁）的用户，登录密码为 "123"（你可以自行设置密码）。同时也请你的同学在他（她）的服务器上在管理器中用类似的方法为你新建登录用户。

注意：MySQL规定登录用户账户要指定其登录位置，“%”表示可以在任何地方登录，所以命令中的 ‘jingjing’@’%’ 表示允许 jingjing 用户在任何地方登录，即本地可以，远程任何机器也可以。



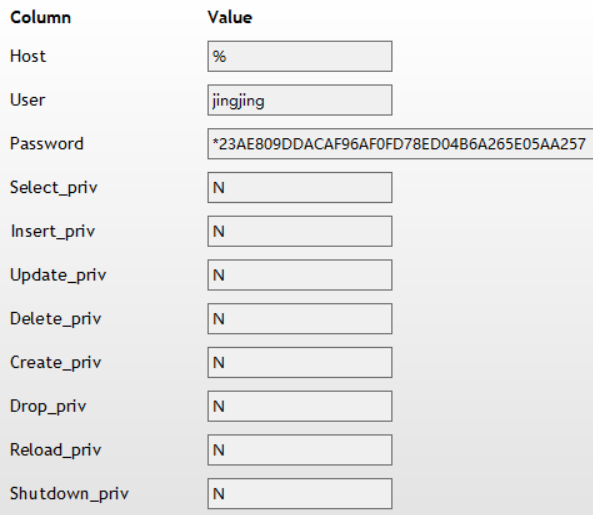
**将你成功创建了用户的操作结果截图放在下面：**



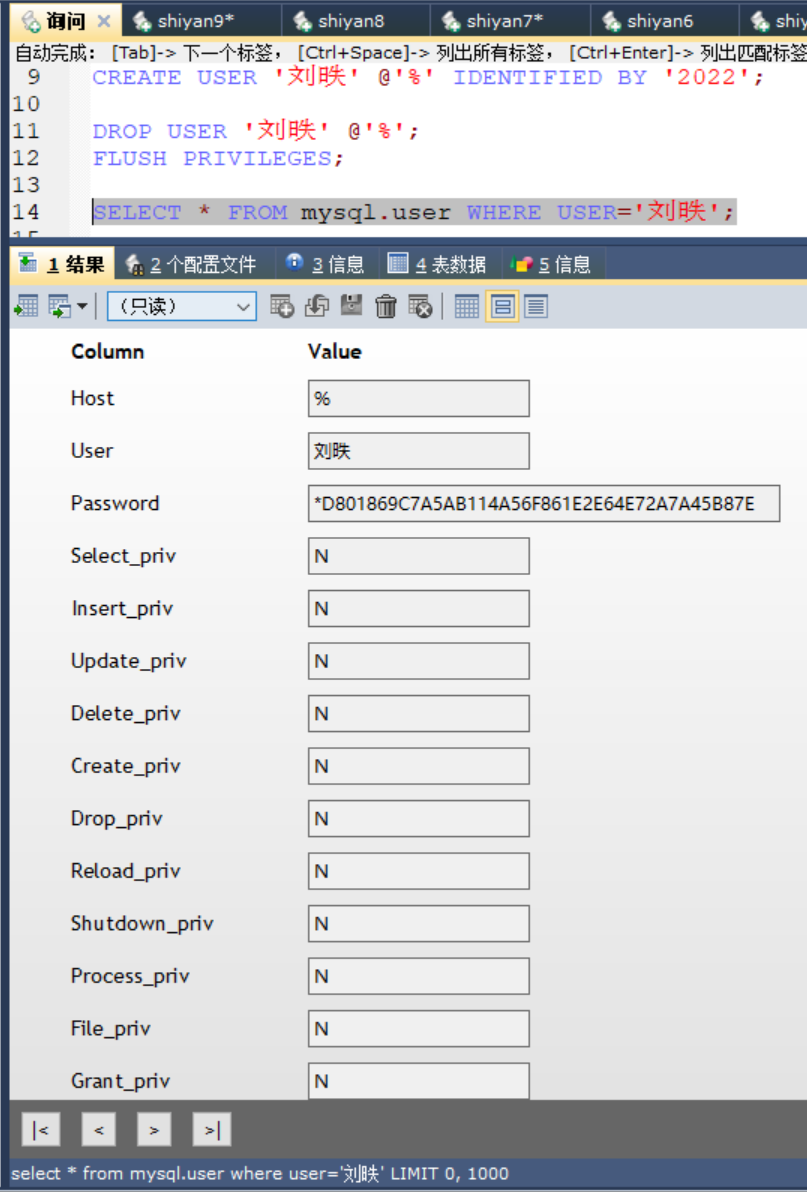
创建后的数据库用户信息被MySQL存放在了它的系统数据库的user表里。所以可以通过查询该user表来验证和查看指定登录用户的信息。

从下图中我们可以看到jingjing的Host为%，表示他能从任何地方登录，密码是加了密的，下面一堆是他的各种权限等。





**参考上图的命令将你创建的你的同学的登录用户的信息截图放在下面：**



上面这个查询mysql.user表的操作仅仅是让大家理解用户鉴别的原理，并不是每次创建用户都需要操作的。

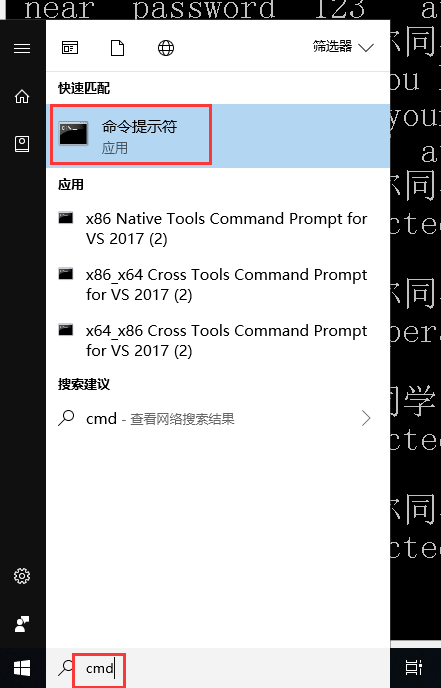
1. 互相登录服务器

请你同学登录到你的服务器来进入你名字的数据库；同时你也登录到你同学的服务器上，试试进入他（她）名字的数据库。

有两种方式，第一种是使用SQLyog，第二种使用下面的命令行方式。（**随便选择一种，如果命令行方法不行就用SQLyog**）

你去登录你同学的服务器的具体做法参考如下：(**以下是命令行方式，SQLyog方式实验课上我会示范**)

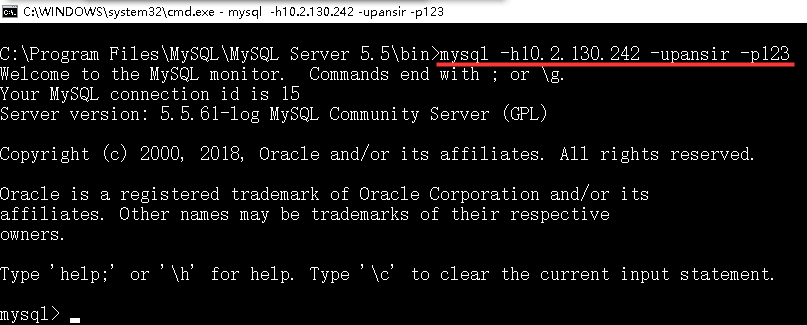
首先启动cmd命令行：



进入windows命令控制台后，输入以下命令进入MySQL的安装目标文件夹的bin文件夹下（默认是C:\Program Files\MySQL\MySQL Server 5.5\bin，如果是自己安装的请自行查找安装的位置）

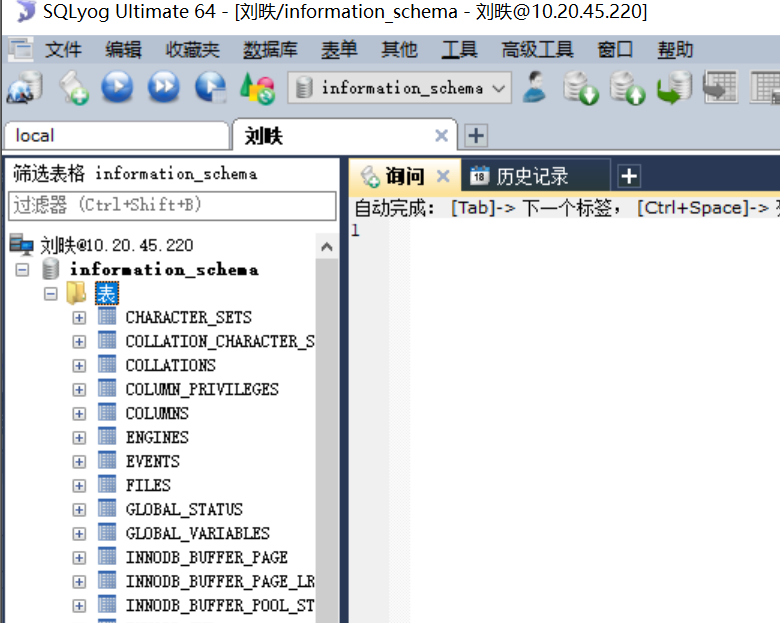


然后在命令行中输入如下命令行（注意-h后换成你同学的IP，-u后换成你自己的姓名全拼，-p后换成你同学给你设定的密码）:



**不管你用命令行还是SQLyog，请将你连接成功到你同学的服务器的操作结果截图放在下面：**





1. **权限实验**

在当前mysql命令窗口（后面我们称为**测试窗口**）中尝试打开他（她）服务器上的数据库：



**将你操作结果截图放在下面，并写出为什么是这样的结果：**

注意：正确的错误信息应该是：Access denied for user …





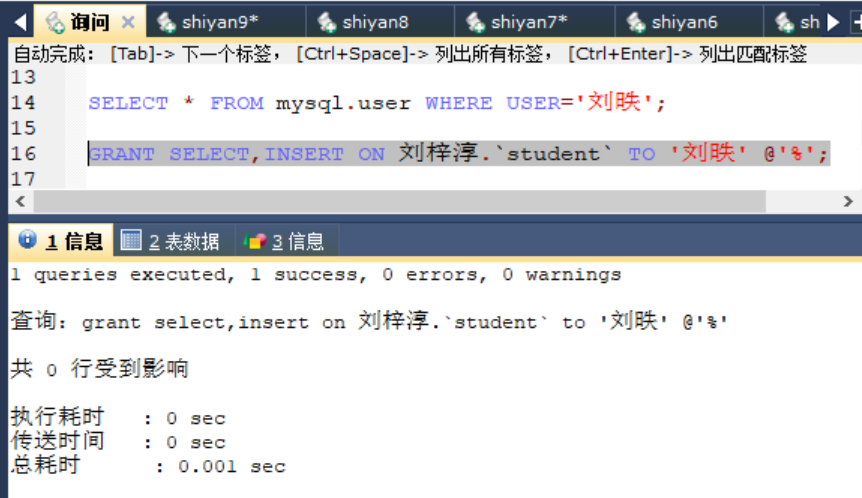
* 1. 授权

互相在本地的**管理窗口**中为对方开通student表的查询权和插入权。

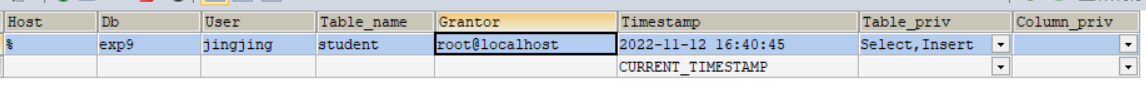
下面的示例为雷雷在本机为登录用户jingjing授权：



**将你成功为同学开通权限的命令及结果的截图放在下面：**

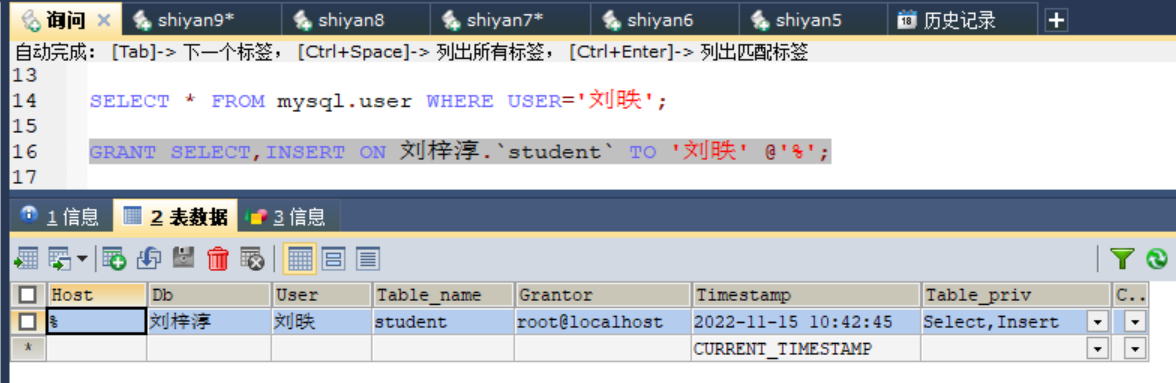


MySQL的系统数据库Mysql中有一张专用于存储数据表权限的表tables\_priv，哪位用户能对哪个数据库的哪些表进行哪些操作，也就是权限就记录于此表中进行管控，所以我们可以查查看：



图中显示出jingjing可以查询（select）雷雷的数据库中的student表，也可对student表插入新的元组（insert）。

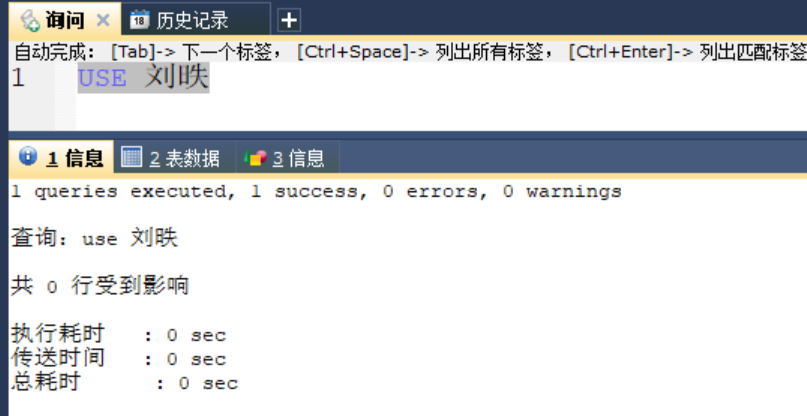
**参考上图的命令将你的服务器的tables\_priv权限表信息截图放在下面：**



在**测试窗口**中再次use你同学的数据库。



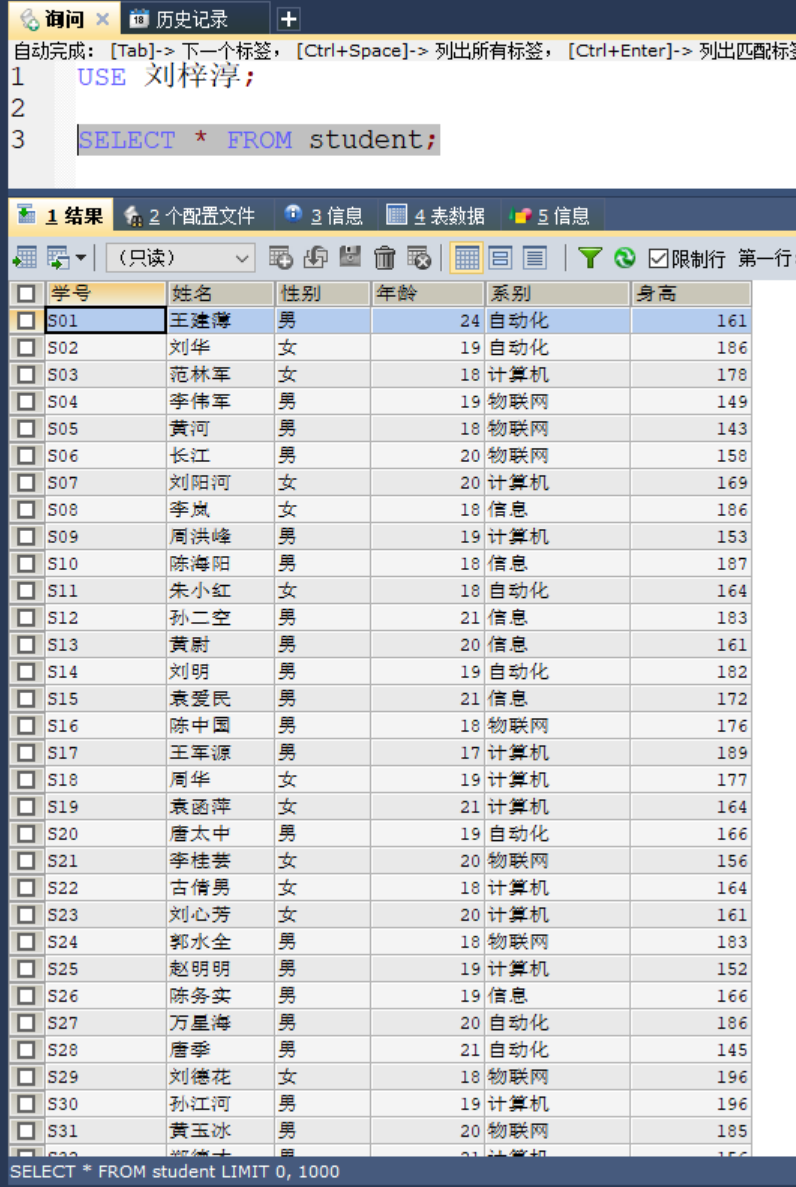
**将你成功打开了你同学的数据库的操作结果截图放在下面：**



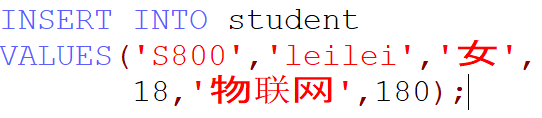
在**测试窗口**中测试自己在对方数据库中对student的查询权。



**将你成功查询了student表的操作结果截图放在下面：**



在**测试窗口**中测试自己在对方数据库中对student的插入权，即将自己的信息（学号只能有4位字符）插入到对方数据库的student表中（注意下面命令是只是示意，并非正确的语法，请根据以前学的知识写出正确的insert into命令）。



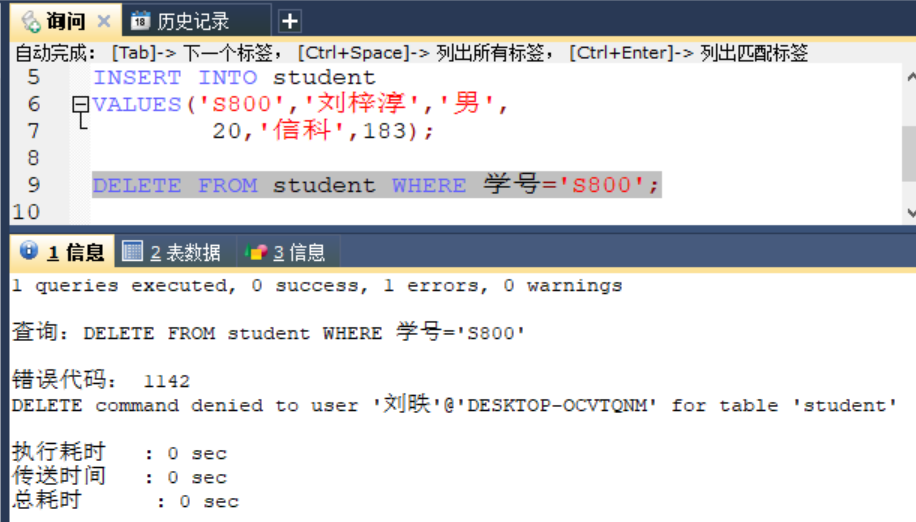
**将你成功在对方数据库的student表中插入了自己信息的操作结果截图放在下面：**



在测试窗口中测试自己在对方数据库中对student的删除权

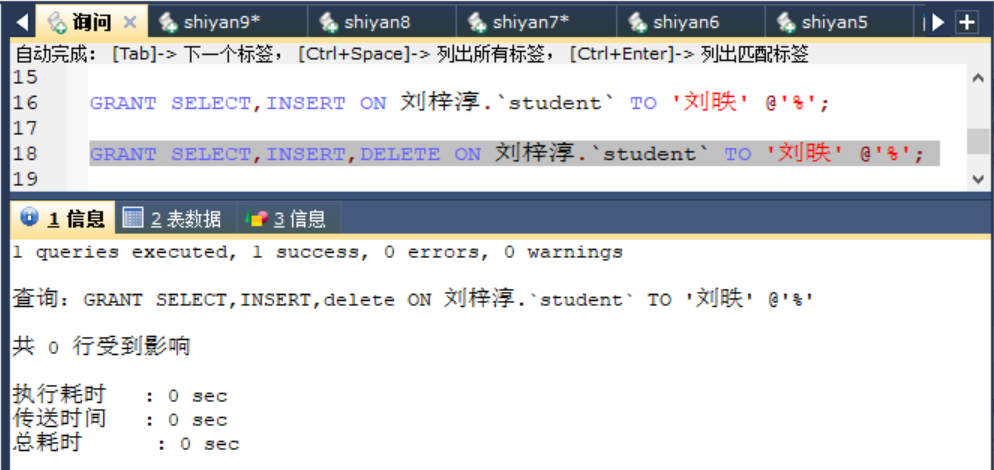
**在对方数据库的student表中delete你刚才insert into进去的那条个人记录的操作截图：**

（注意：肯定删不掉，想想为什么）

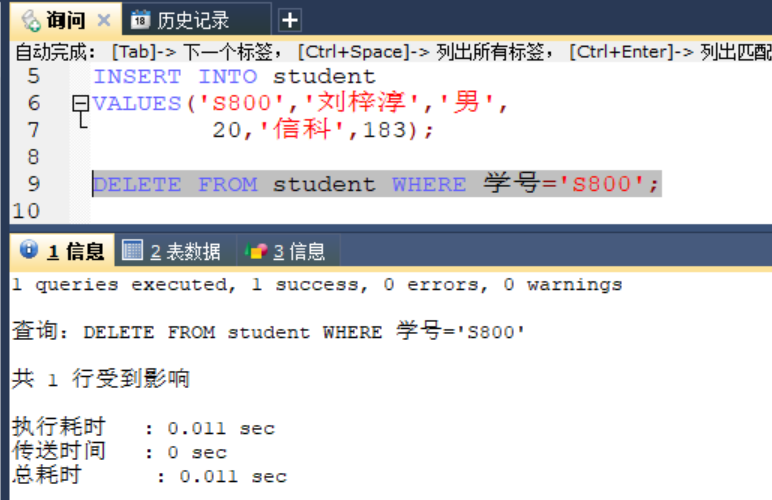


参照前面的做法，你和你的同学各自在自己的**管理窗口**中增加对方对student的删除权并验证

**将你成功为同学开通了student权限的命令及结果截图放在下面：**



**测试在对方数据库的student表中删除了自己记录的截图放在下面：**



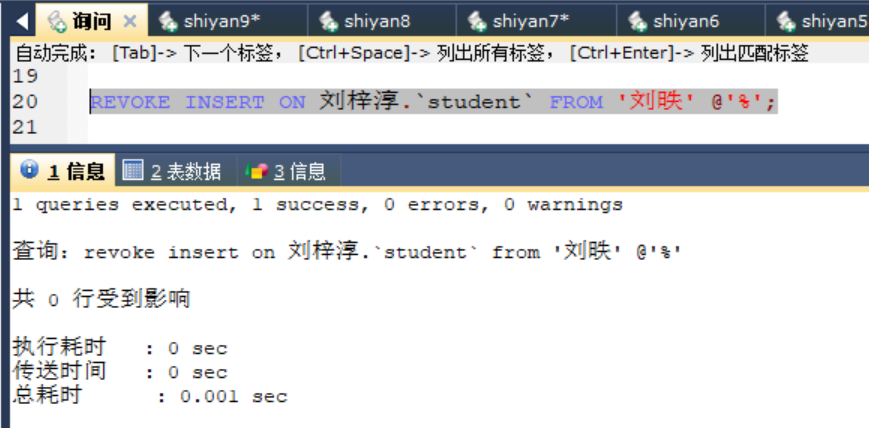
* 1. 回收权限

你和你的同学各自在**管理窗口**中回收对方对student的插入权并验证

回收权限的命令参考：



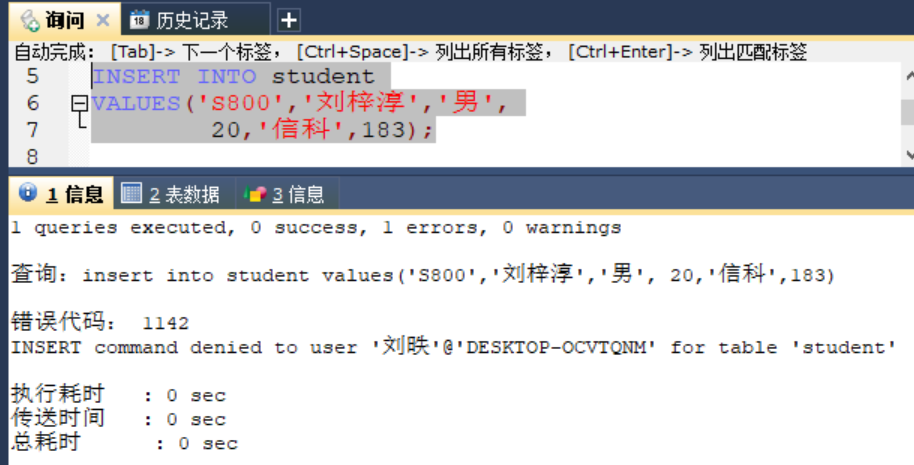
**将你成功为同学回收了权限的截图放在下面：**



在测试窗口中测试自己在对方数据库中对student的插入权

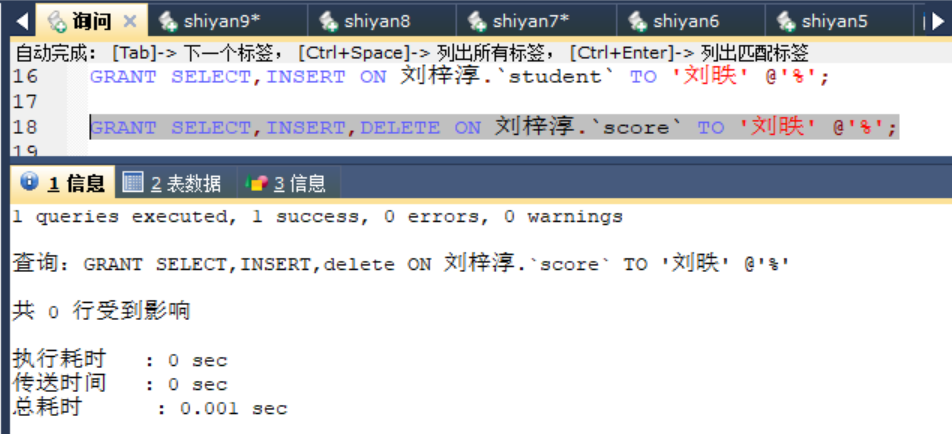
**在对方数据库的student表中再次插入你的个人记录的操作截图：**

（注意：肯定不能插入了，想想为什么）

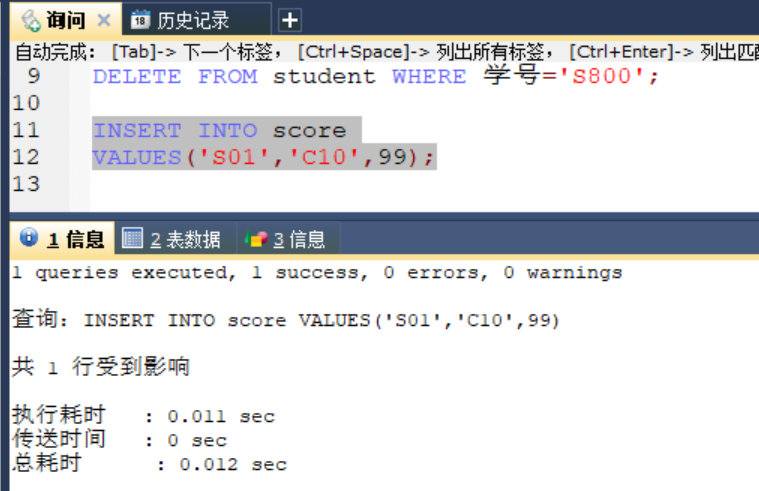


参照前面的做法，你和你的同学互相协商并实验对库中score表开通查询和更新的权限并测试效果。

**将你成功为同学开通了score权限的命令及结果截图放在下面：**



**测试在对方数据库的score表中成功修改了数据的截图放在下面：**

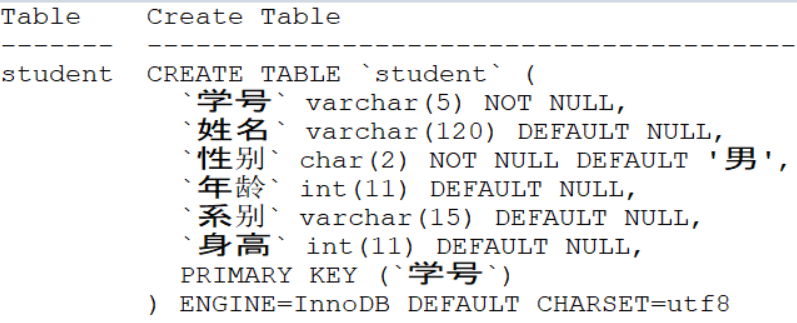


1. **数据库完整性保护**

可以关闭测试窗口了，只留下管理窗口进行后面的实验。

1. 实体完整性

实体完整性即主码的属性不能为空。而主码就可保证元组是不重复的，即主码值是不能重复的。



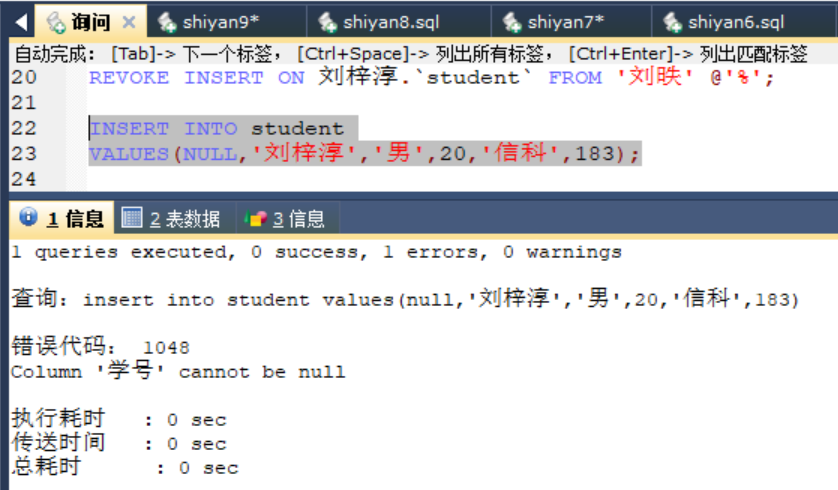
可见学号是student的关键码。

a）用自己的姓名、性别、年龄、系别、身高等信息为数据，写一条insert命令，测试能新增加你自己为一名学生，但学号为空（NULL）吗？

(提示：本实验也顺便复习一下insert命令)

**你的测试用例及测试情况截图：**

注意：正确的错误信息应该是：Column '学号' cannot be null

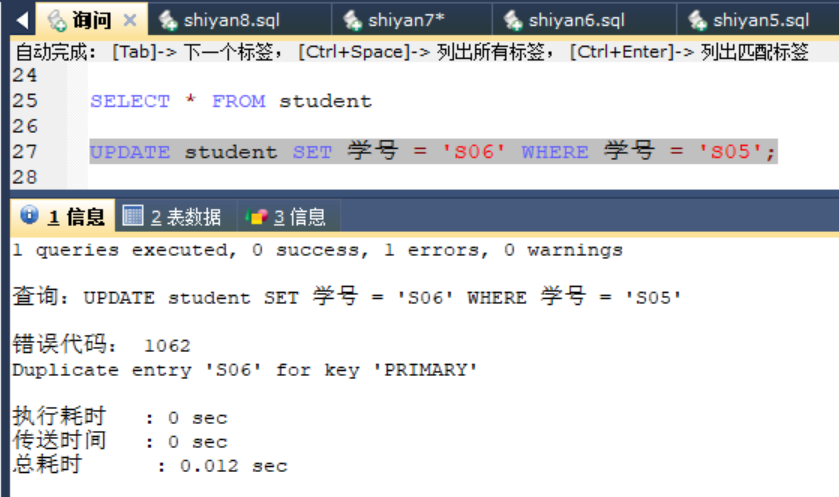


b）自编update命令测试能把两个以上的学生的学号改为相同的吗？比如能把长江同学和黄河同学的学号改相同吗？

(提示：本实验也顺便复习一下update命令)

**你的测试用例及测试情况截图：**

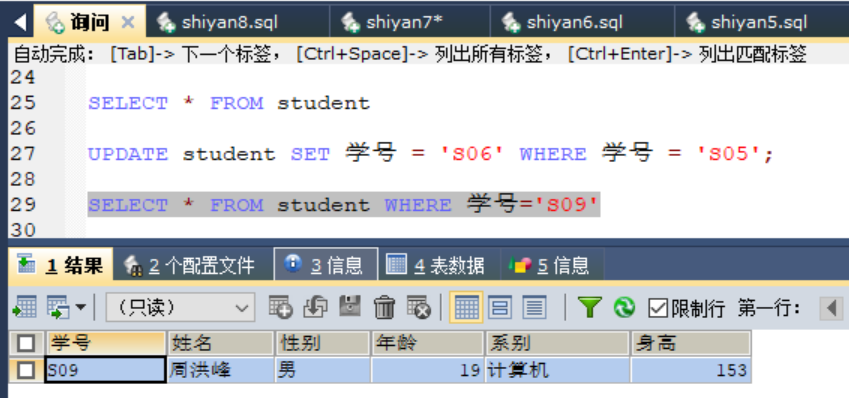
注意：正确的错误信息应该是：Duplicate entry ……. for key 'PRIMARY'



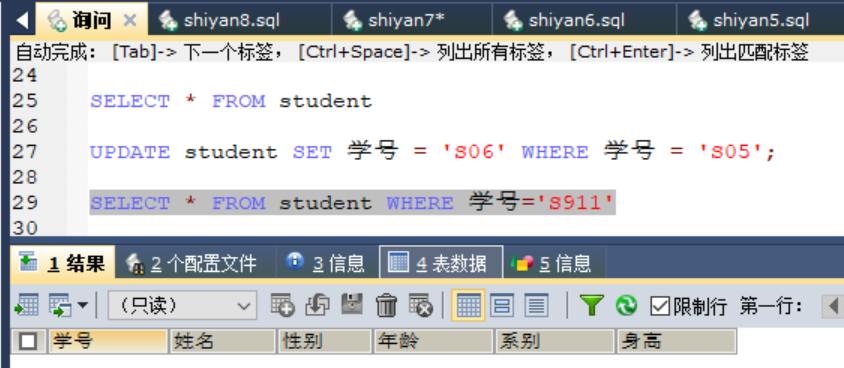
c）系统怎么知道学号是否会重复的呢？

答案就是：查询。查询它的数量，或者是用exists量词查询！

**请用select命令证明新增学号’S09’会重复，你的命令及结果截图：**



**请用select命令证明新增学号’S911’不会重复，你的命令及结果截图：**



1. 参照完整性

参照完整性保证外码的值要么是被参照关系中的主码值，要么取空值。

输入show create table 命令查看course表的结构：

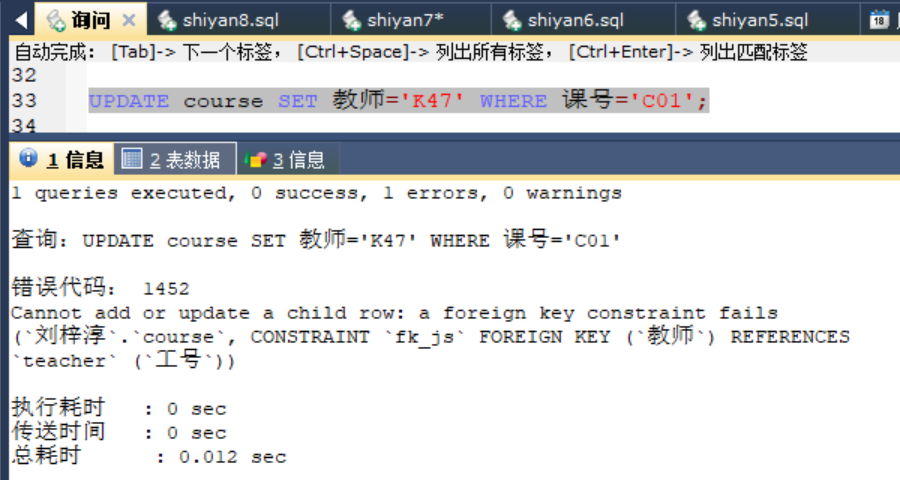


可见course中的任课教师编号要参照引用teacher表中的工号。

a）用update命令测试能将C01高等数学的任课教师编号改成K47吗？

**你的修改命令及其运行情况截图：**

注意：正确的错误信息应该是：Cannot add or update a child row: a foreign ……



**为什么是这样的结果呢？**

外键约束失败，添加的外键列与另一个表的唯一索引列（一般是主键）的数据类型不同

b）测试能用update命令将T01老师的工号改成T99吗？

**你的修改命令及其运行情况截图：**

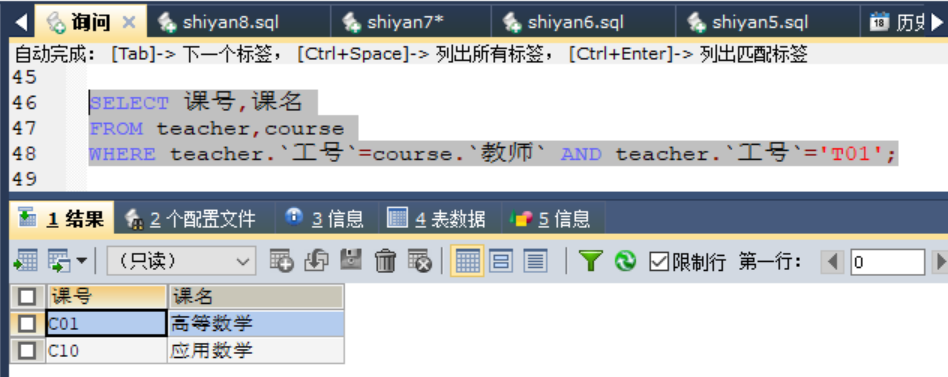
注意：正确的错误信息应该是：Cannot delete or update a parent row: ……



**为什么是这样的结果呢？**

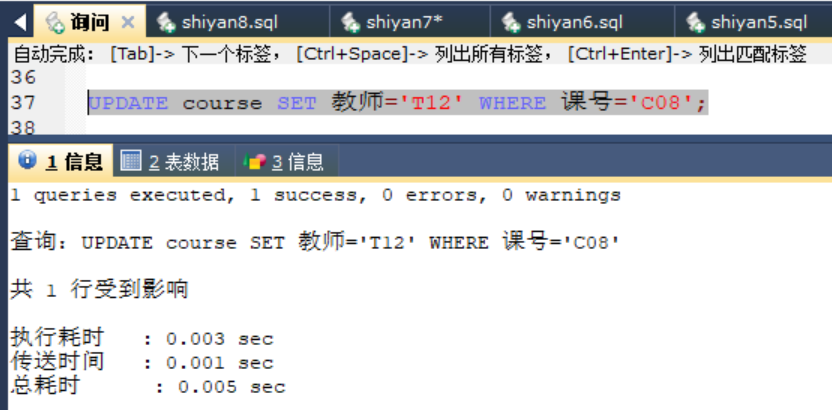
无法删除或更新父记录：外键约束失败；设置的外键与另一个表中的唯一索引列（一般是主键）中的值不匹配

**系统又是怎么知道course表中有没有工号为T01的老师上的课的呢？请你设计一条SQL命令证明一下。**



c）用update命令测试能将C08软件工程的任课教师编号改成T12吗？

**你的修改命令及其运行情况截图：**

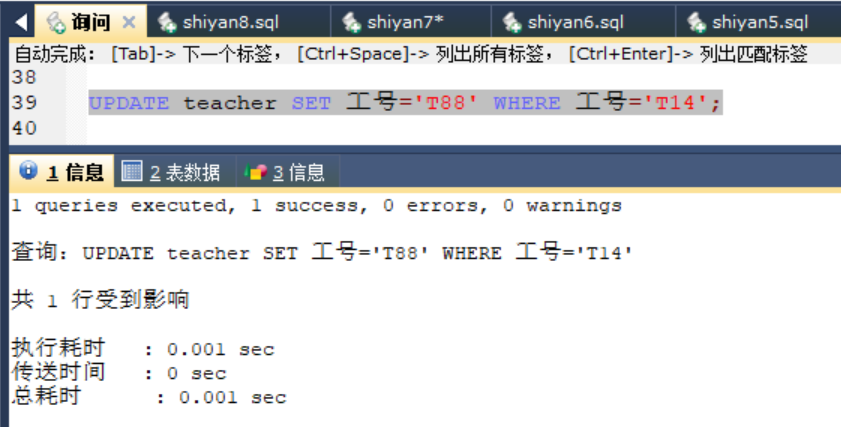


**为什么可以成功修改呢？**

外键约束成功，添加的外键列与另一个表的唯一索引列（一般是主键）的数据类型相同

d）测试能用update命令将T14老师的工号改成T88吗？

**你的修改命令及其运行情况截图：**



**为什么可以成功修改呢？**

设置的外键与另一个表中的唯一索引列（一般是主键）中的值不存在匹配关系，course表没有教室T14的课，所以不存在此类约束，可以进行更改

e）如果 我们确实要用update命令将有课的T01老师的工号改成T99而保持他所上的课不变呢？很显然 ，如果系统能自动将引用T01的所有外键的值自动改成T99，那T01老师的工号改成T99就不会违反参照完整性。幸运的是，数据库管理系统有这样一种管理外键的方式。

据库管理系统有这样一种管理外键的方式。

数据库系统的外键值更新模式一般有3种（MYSQL其实有5种，大家可以HELP CREATE TABLE后查找reference\_option查看）：

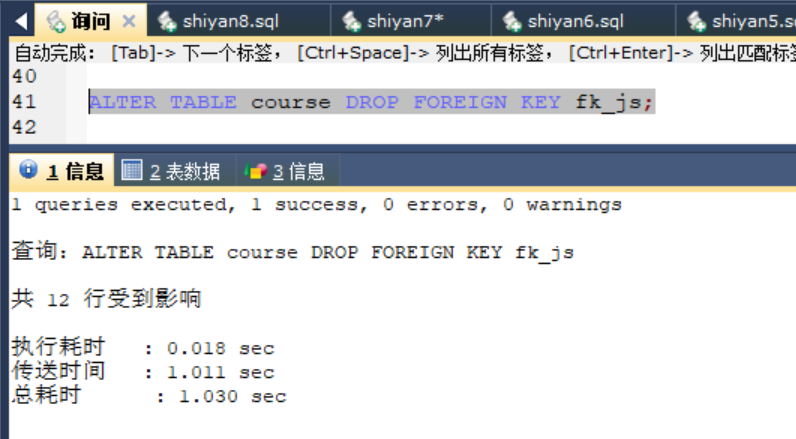
* 默认的外键管理是Restict，即限制方式。前面实验我们已经见识了这种方式的作用。即外键值要么是空值（相当于没引用），要么一下是被参照关系中的某个主键值（就是所谓的引用），不能有其它的取值。
* CASCADE，即级联方式，可以理解为株连九族。CASCADE只能用于UPDATE或者DELETE。如果是用于UPDATE，则主键改变后，引用它的外键值自动改变成新主键值来保证参照完整性；若用于DELETE，当主键所在的元组被删除时，外键所在的关系会自动删除所有引用了该主键的元组来保证参照完整性。
* SET NULL，置空，只能用于UPDATE或者DELETE。它的作用是当主键值改变后，引用它的外键值自动改为NULL来保证参照完整性。

所以，这里可以用CASCADE模式来实现我们改T01的工号而保持他上的课不变这种需求。具体做法如下：

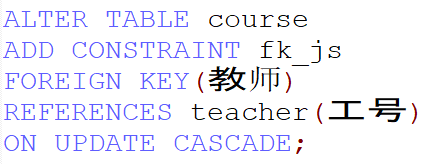
首先，由于前面已经设置了fk\_js外键为Restrict（默认的模式），所以需要先删除该外键约束：



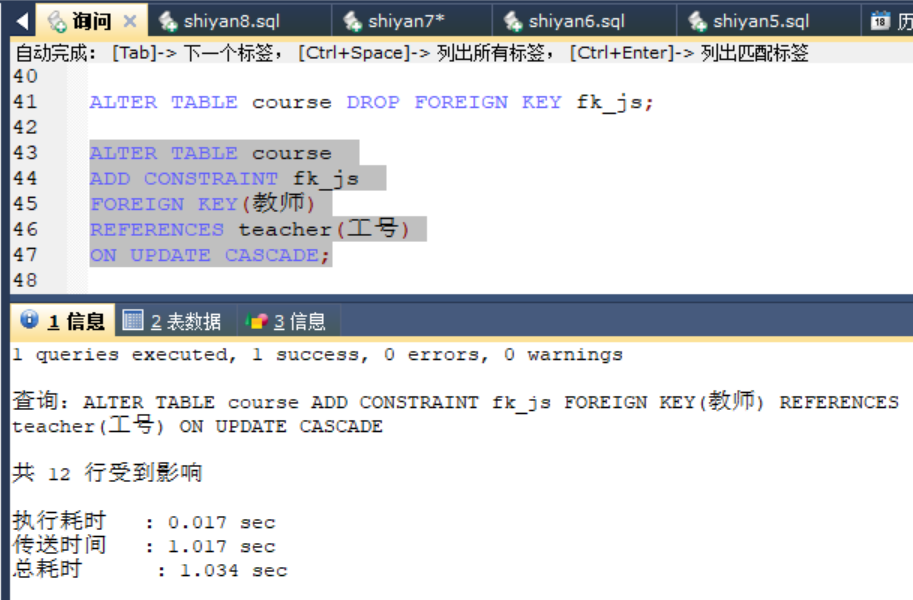
**你的命令及其运行情况截图：**



然后重新设置可级联更新的外键约束：

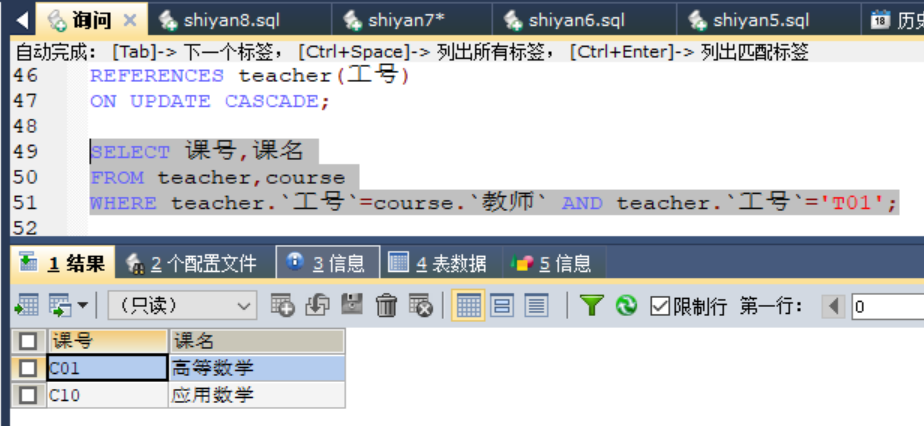


**你的命令及其运行情况截图：**



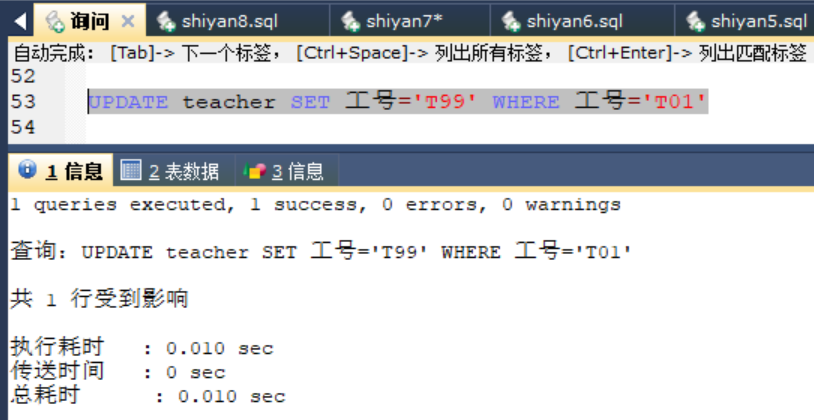
先查询一下course表中T01老师上了哪几门课

**你的查询命令及其运行情况截图：**



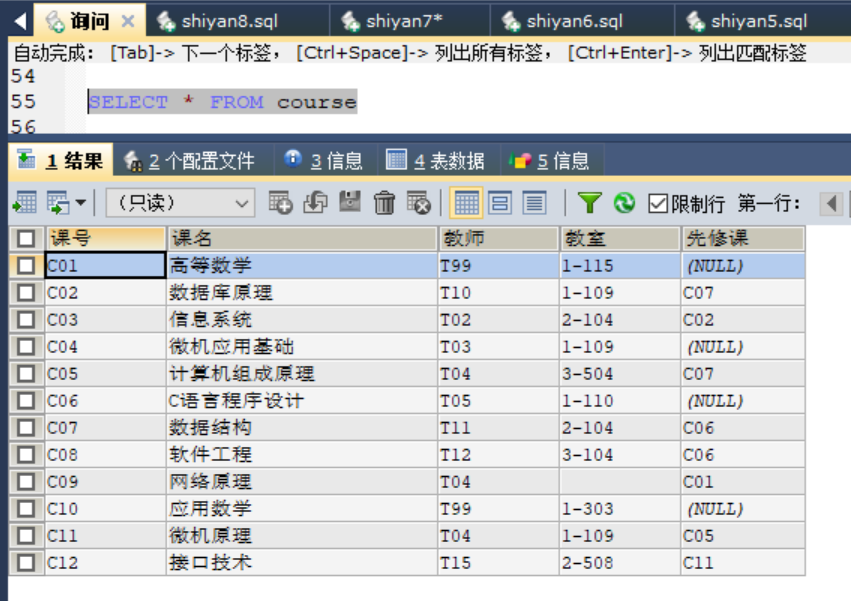
现在试试看用update命令将T01老师的工号改成T99

**你的修改命令及其运行情况截图：**



成功了吧。再次查询一下course表，看看刚才T01老师上的课，现在的工号是什么。

**你的查询命令及其运行情况截图：**

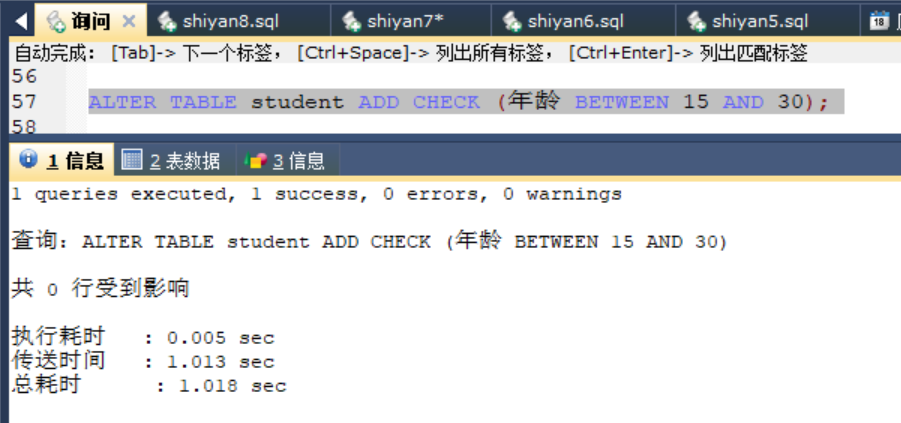


说明：具体使用哪一种外键更新模式，要根据业务中的实际情况来进行选择，没有绝对。甚至可以不设计外键，自己从业务逻辑层来管理和控制。

1. 用户定义完整性

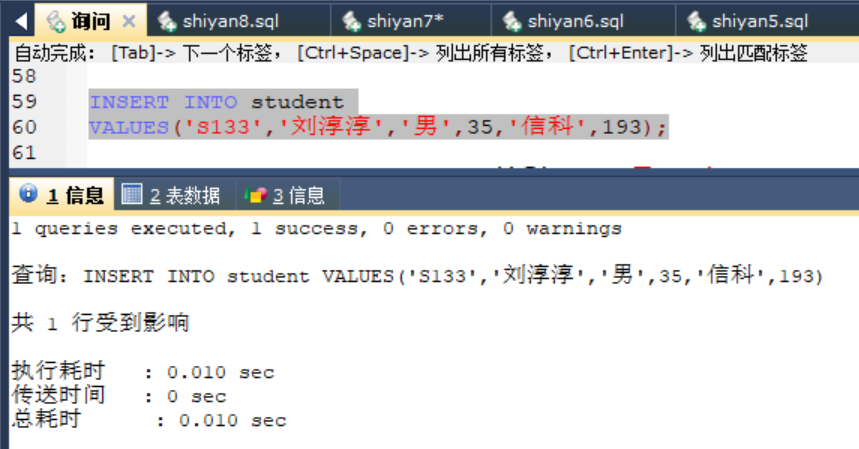
给student表添加一个check约束，用于检查插入的年龄不能大于30或者小于15。

**你的修改命令及其运行情况截图：**



往student插入一条年龄为35岁的学生信息，

**你的插入命令及其运行情况截图：**

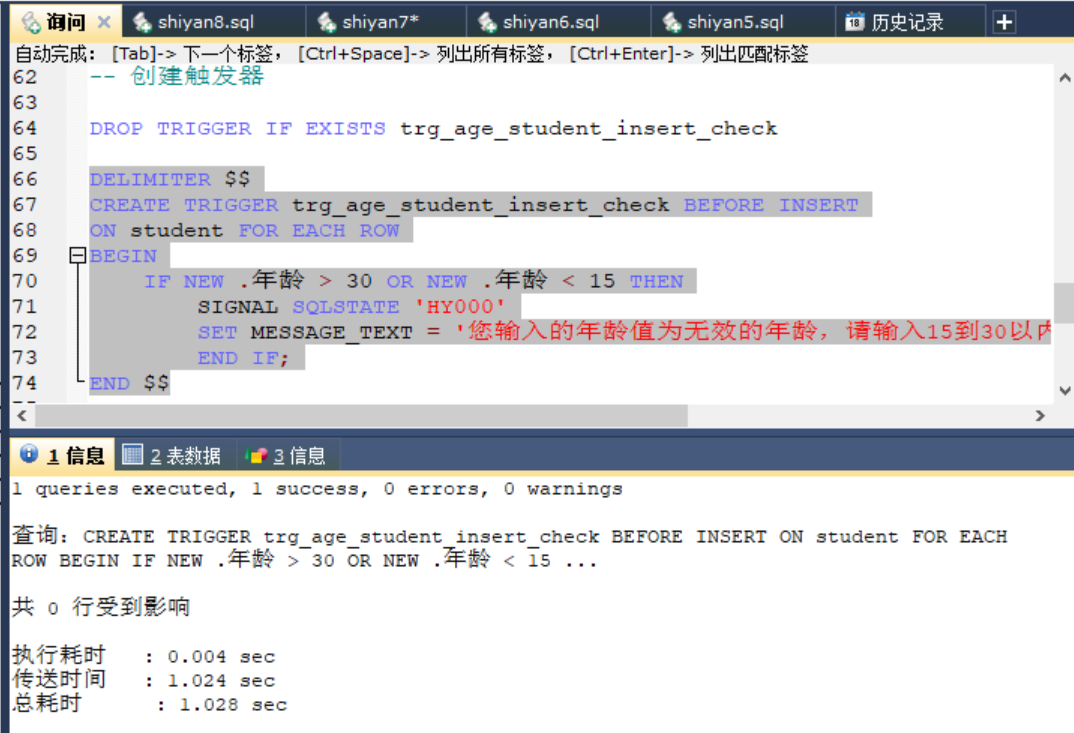


**为什么能插入成功呢？**

Check只是一段可调用但无意义的子句，虽然会被分析但是也会被忽略，因此可插入成功

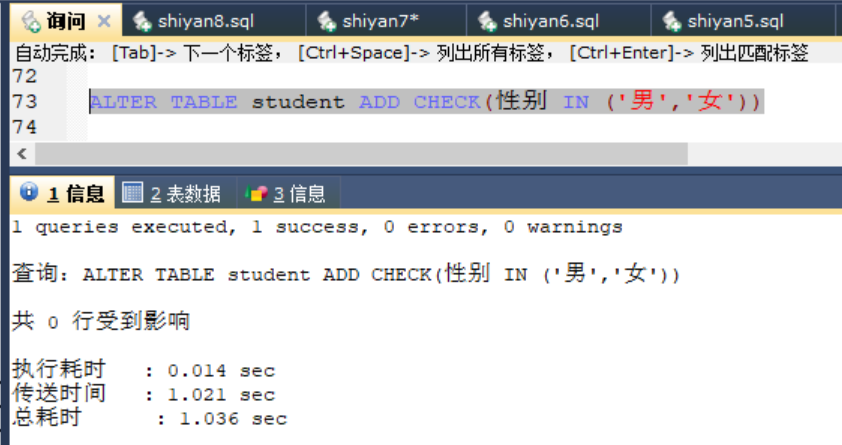
接下来设计一个触发器，使用触发器来代替约束插入数据的有效性，用于检查插入的年龄不能大于30或者小于15。（可以参考实验8中的触发器设计方法）

**你的触发器及其运行情况截图：**



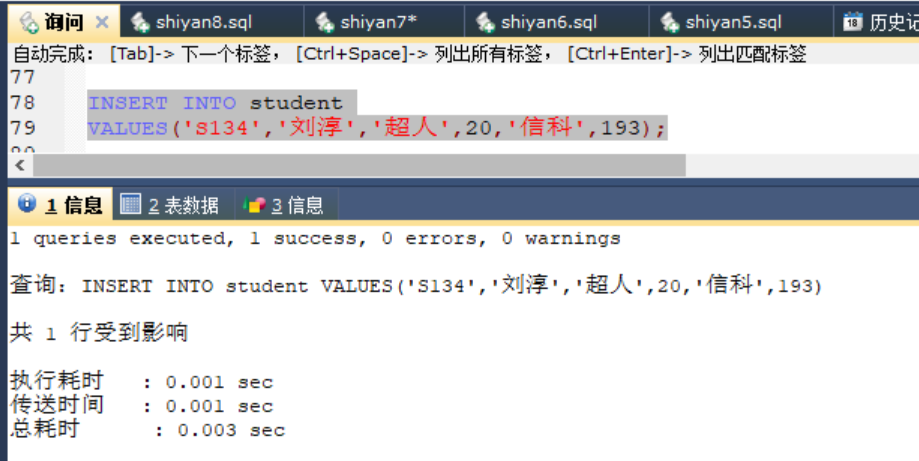
给student表添加一个check约束，用于检查插入的元组性别只能为男或女。

**你的修改命令及其运行情况截图：**



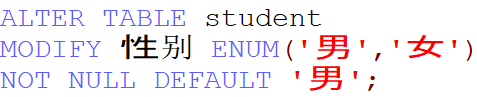
往student插入一条学生信息，性别的取值为男女以外的值。

**你的插入命令及其运行情况截图：**



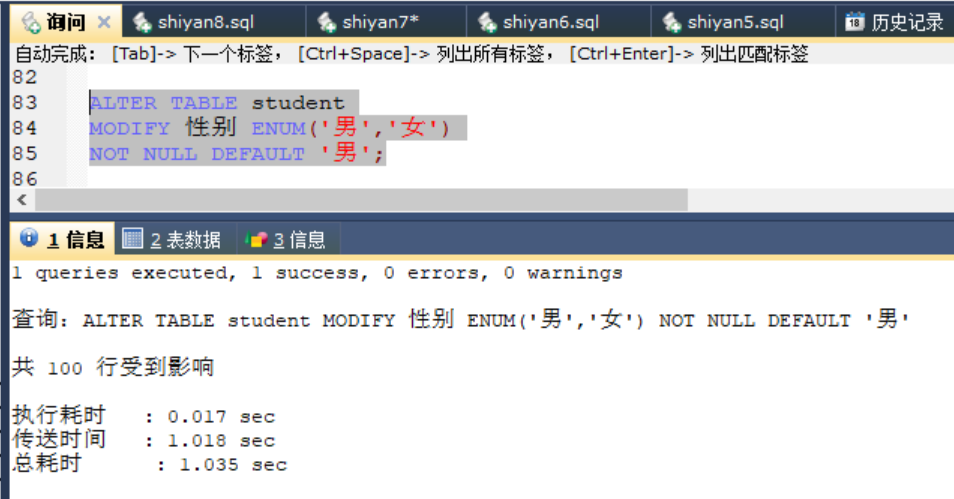
好像check没起作用对吧？尝试用下面的方法解决：

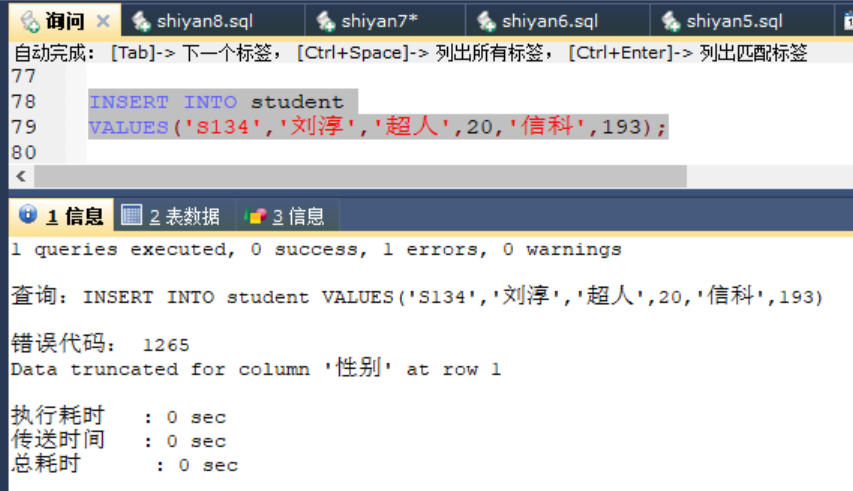
如果要设置CHECK约束的字段范围小，并且比较容易列举全部的值，就可以考虑将该字段的类型设置为枚举类型 enum()或集合类型set()。比如性别字段可以这样设置，插入枚举值以外值的操作将不被允许。用如下的方法修改student表：



修改后再尝试往里面插入一条学生信息，性别的取值为男女以外的值。

**你的修改命令及其运行情况截图：**





【实验结果】

**请将完成后的本实验报告提交到课程平台。**