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11

专业班级：信科202003 姓名：刘梓淳 学号：202003919

1. 【实验目的】

* 掌握数据库应用开发的一般方法

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

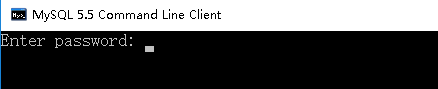
1. 启动MySQL服务

利用windows的“服务”，检查本机的MySQL服务器是否“正在进行/已启动”中，如果不是，则请启动它。当然，如果你确定服务器已经启动了，则可以直接跳过这一步。

1. 启动MySQL命令控制台

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

从开始菜单启动“MySQL 5.5 Command Line Client”,以默认的root身份连接到本地服务器实例，此时要求输入密码



MySQL服务器默认用户为root，无密码，直接回车即可(注意：机房的密码是123456)。如果是使用自己的机器，安装时设置了密码，请使用自己设置的密码后回车。

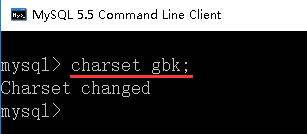
回车后如果闪退，说明连接失败了，请检查密码是否正确、服务器是否正在运行等，然后重新启动“MySQL 5.5 Command Line Client”重新连接。

连接成功后，会看到“welcome to the….”等多行信息，然后看到明显的



命令行标志。

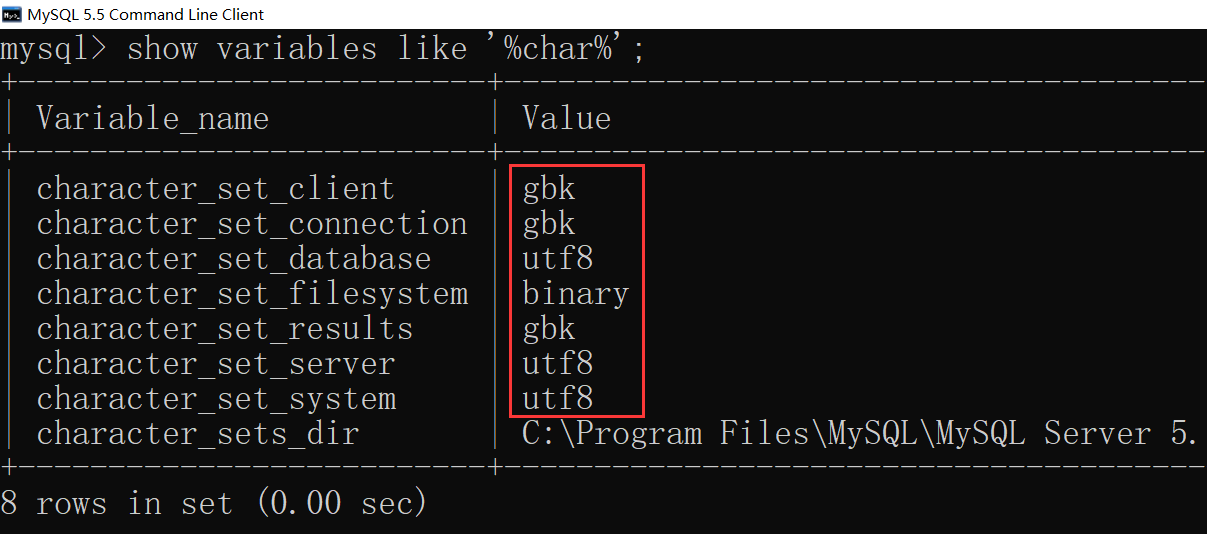
由于我们后面的实验内容中，需要用到中文字符，所以需要使用以下命令设置客户端的字符集为gbk：



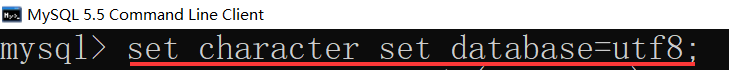
再输入以下命令检查服务器端的字符编码系统是否支持中文：



如果显示结果如下图所示，没有出现“latin1”，说明系统目前完成适用中文：

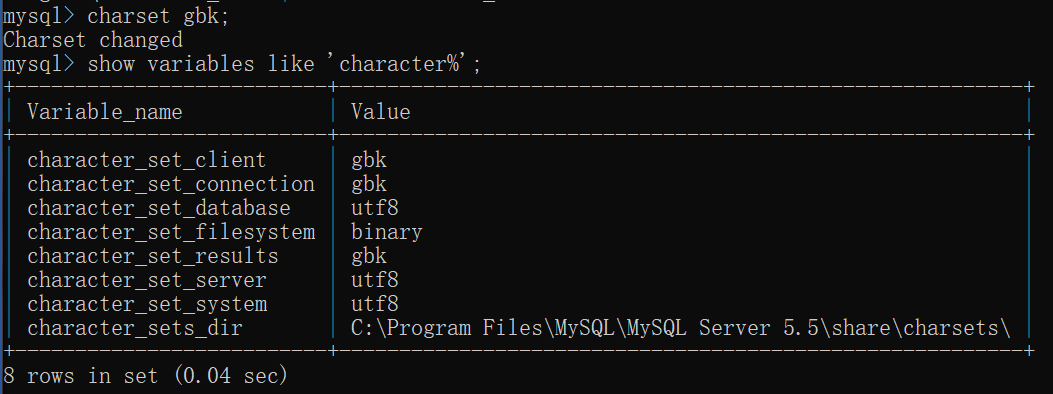


如果显示结果中出现有“latin1”，则可按实验1的中文配置办法先修改配置文件my.ini ，再重启MySQL服务来解决，或者用set命令将相应的字符集进行临时配置。比如“character\_set\_database”的value为“latin1”，则可用:



将其value改为“utf8”。其它以此类推。

你的配置好字符集后的编码变量显示的截图（不能有latin1）：

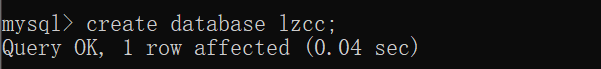


1. 准备实验数据库

输入以下SQL的DDL命令在本机服务器上建立一个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"潘sir"改成你自己的姓名**）



**请参照上面的示例图，将你成功创建库的DDL命令及结果的截图放在下面：**



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



**将你命令及其执行结果截图放在下面：**



下载用于生成实验数据的[import\_xjgl\_data.sql](http://eol.sicau.edu.cn/Courses/Down?courseid=6162398&rid=4253) 脚本文件保存到本地D:\下。然后在“MySQL 5.5 Command Line Client”中输入并执行如下命令导入实验数据：

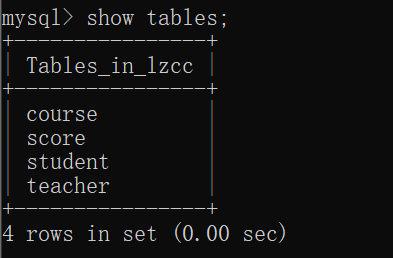


完成后再输入并执行如下命令查看导入的情况：



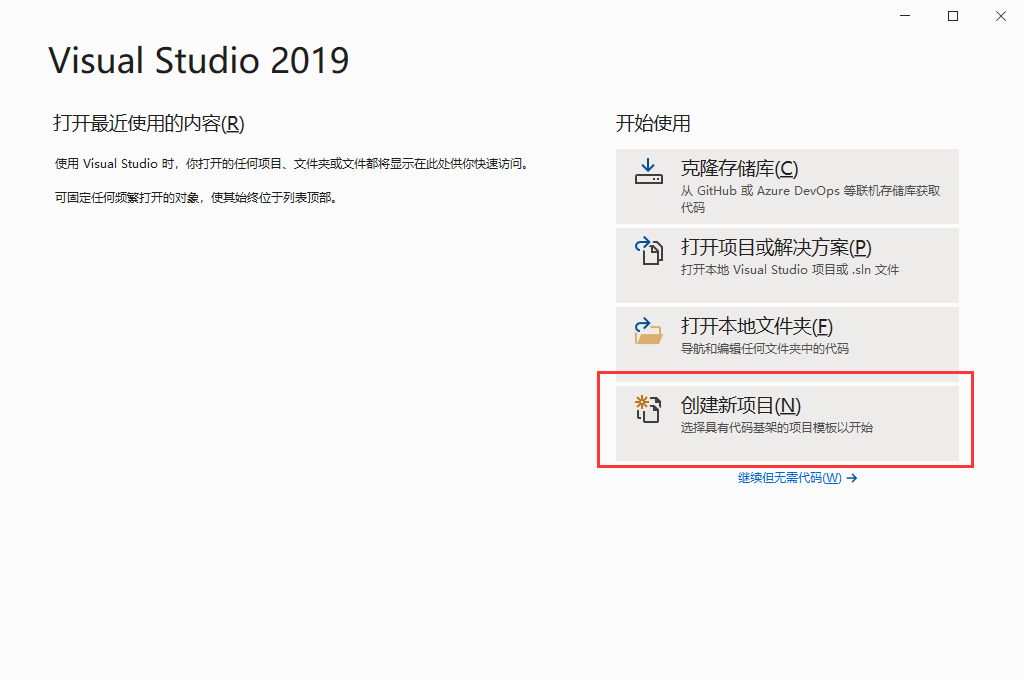
提示：你应该能看到有熟悉的4张表了

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。

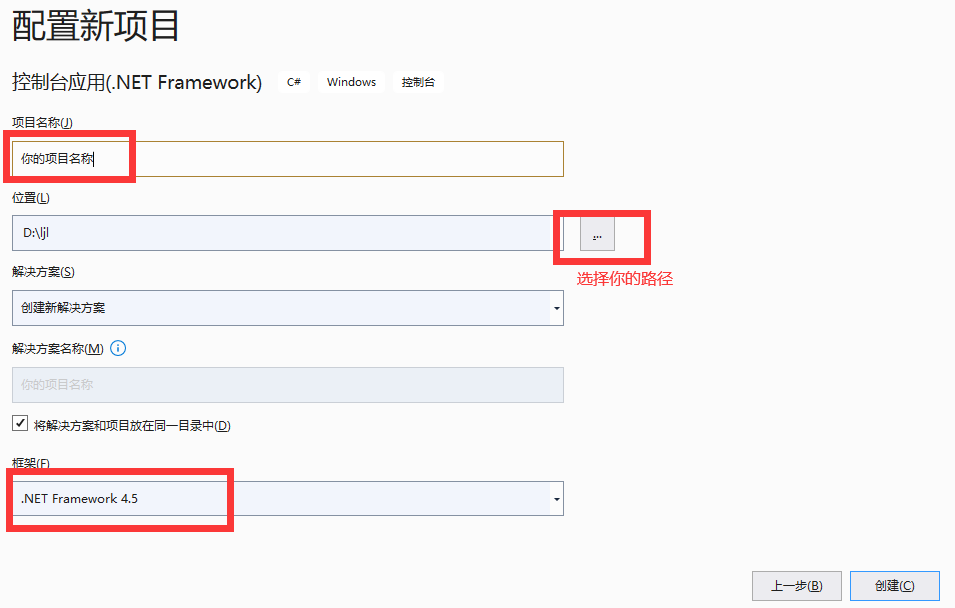


1. **数据库应用软件设计**
2. 启动开发环境，建立工程项目

启动VS 2019：

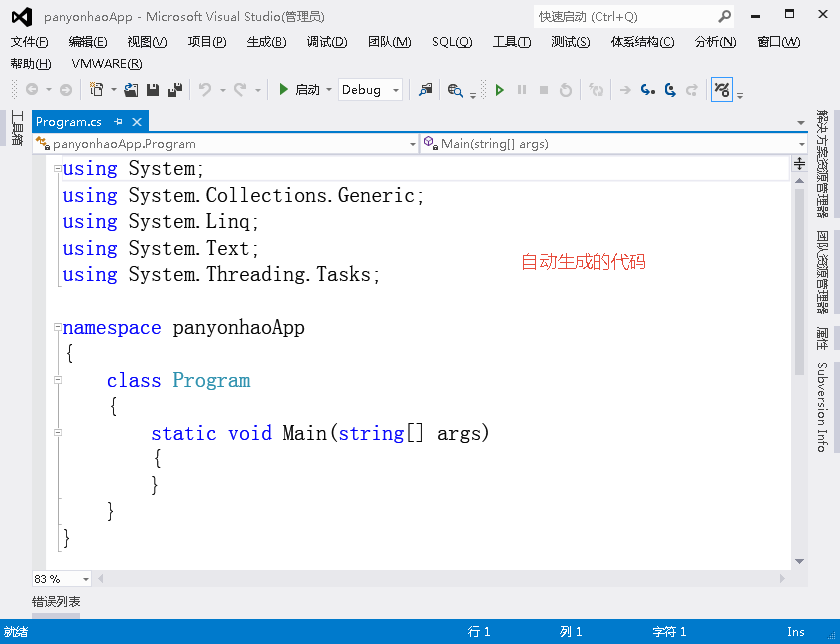






**注意部分机房如果界面中没有名称和位置等的输入，可以在确定后马上保存项目，就会要填写名称和位置等信息了。**

确定后：

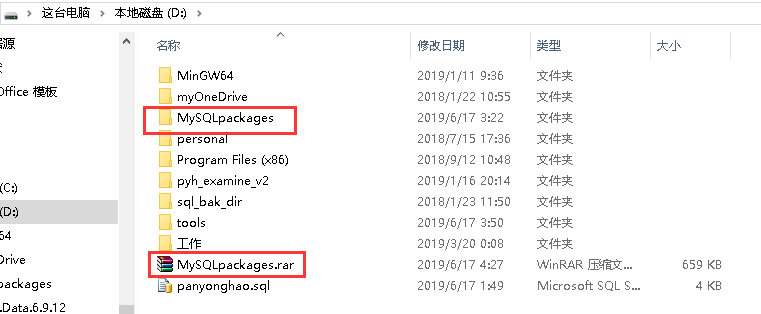


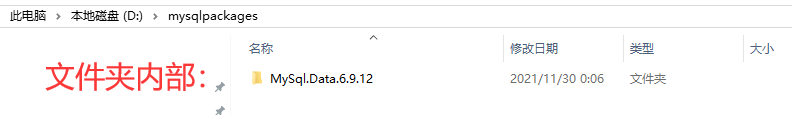
开发MySQL应用需要MySQL组件包，本来可以很简单的直接在线安装，但需要互联网支持，所以我们用如下的离线安装方式解决。

下载MySQL开发组件包：

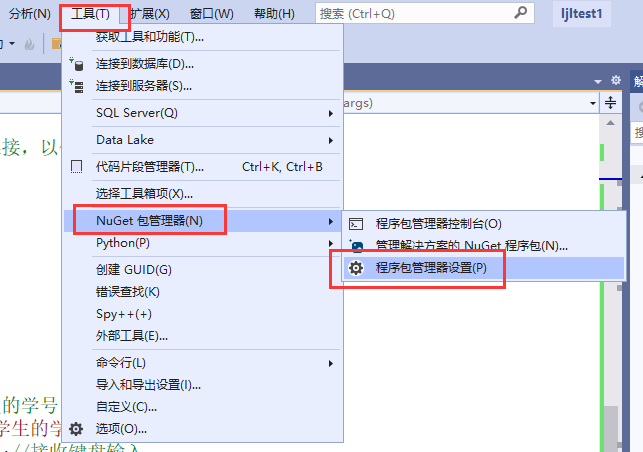
<ftp://10.2.132.71/数据库/MySQLpackages.rar>

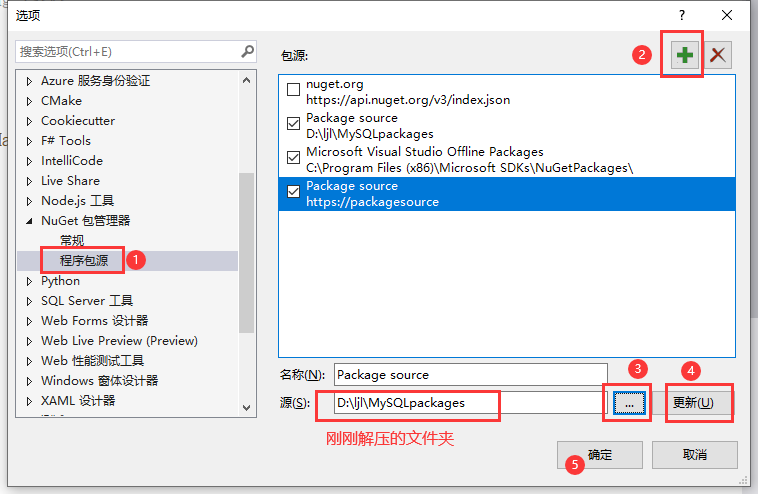
将其下载解压到D盘根下：

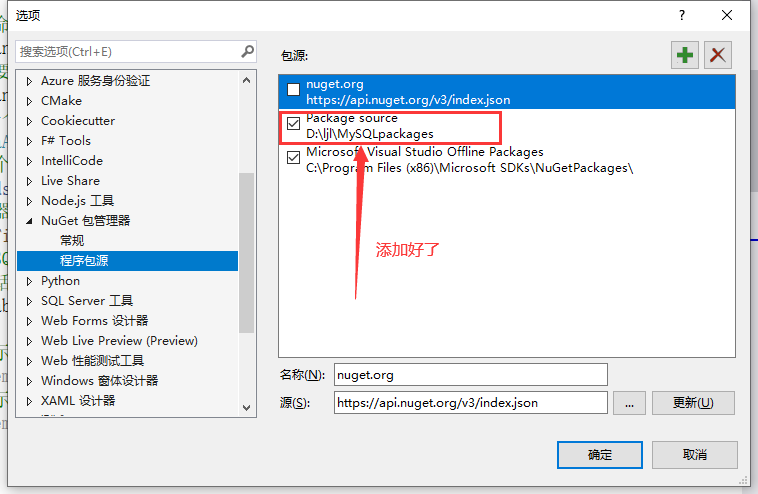




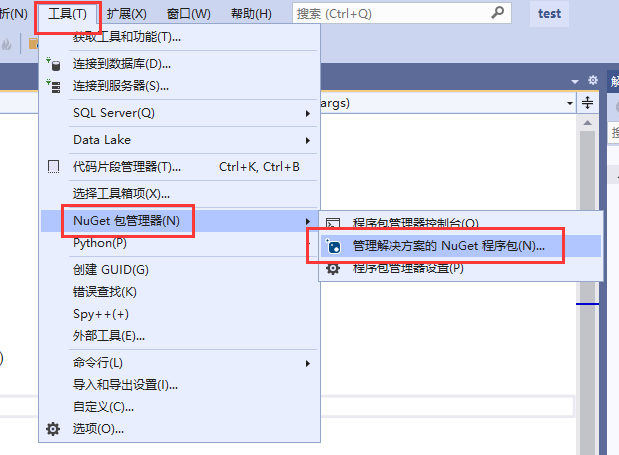
回到VS 2019，配置安装包路径：

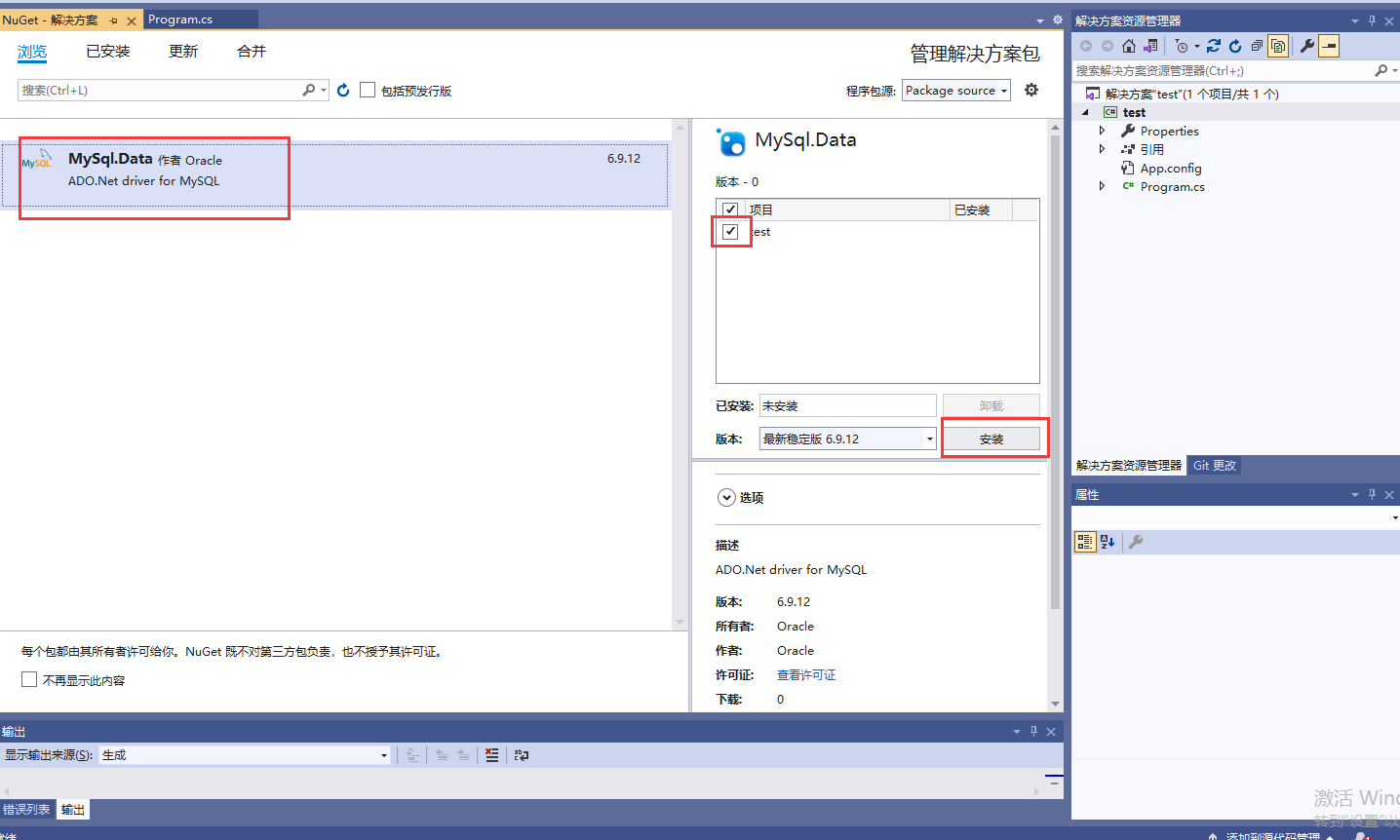






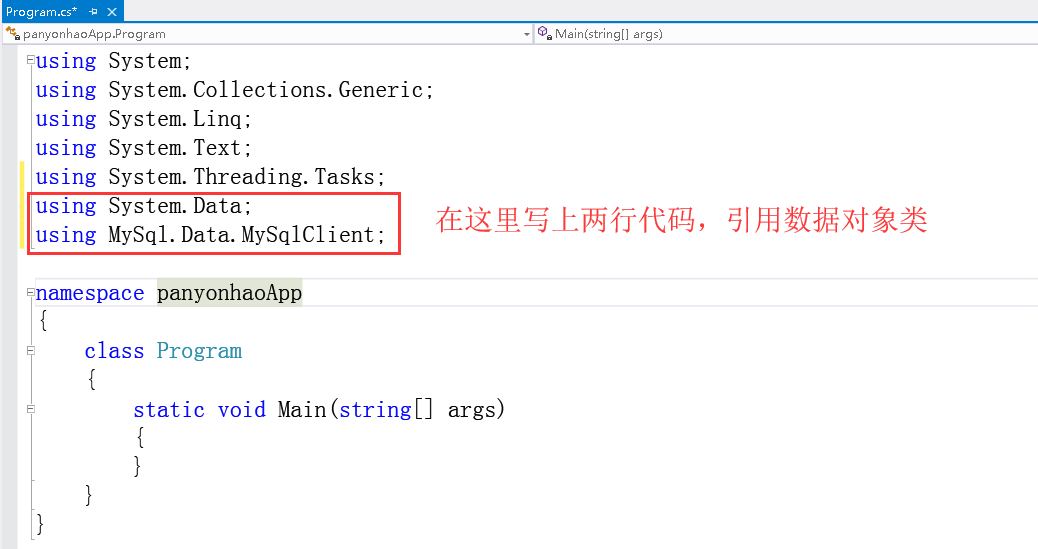
确定后，安装MySQL组件包：





1. 开始编码写程序

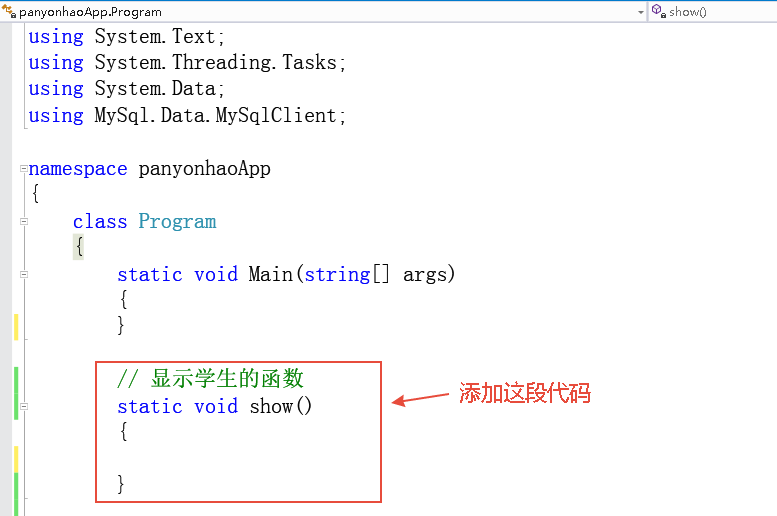
导入数据对象的命名空间：



注意：

以下//引导的绿色部分是注解，可以不用录入，但必须仔细读，读懂理解。

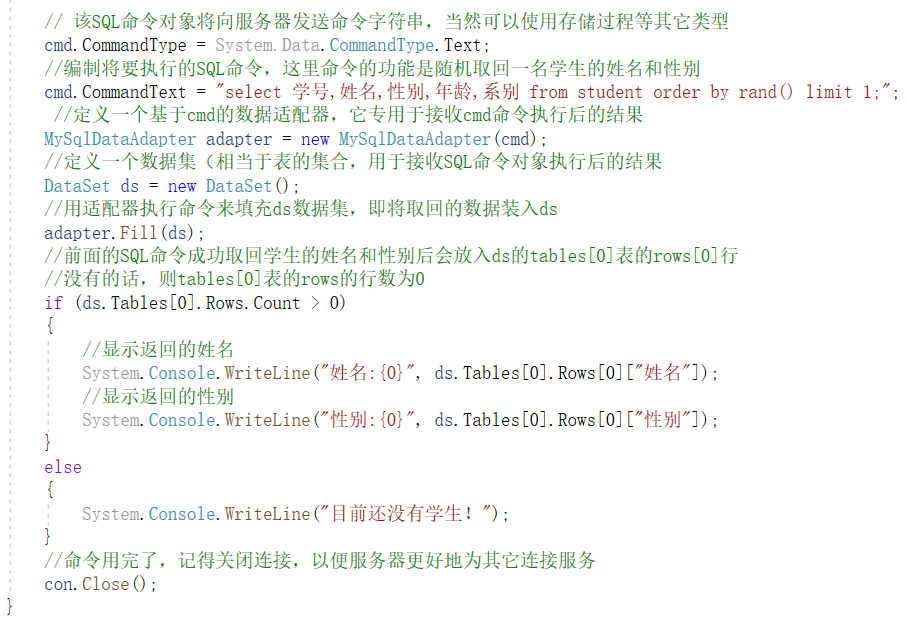
构造显示学生信息的show函数：



编写show函数体内部的代码：

（补充一点，如果有部分同学的机器上没有root有密码，所以下面conStr的字符串可以**去掉**“；password=密码” ）。





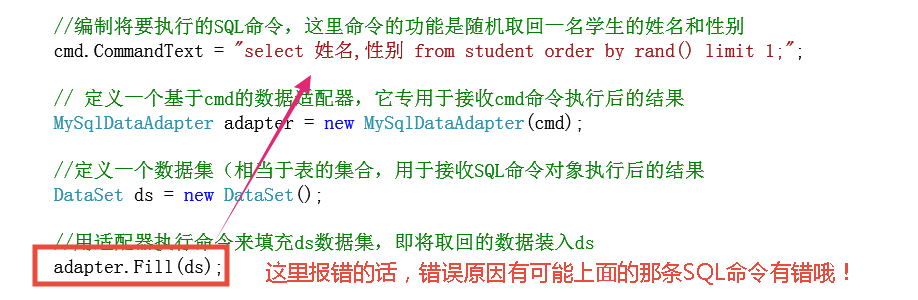
在Main函数中编写对show函数的调用：



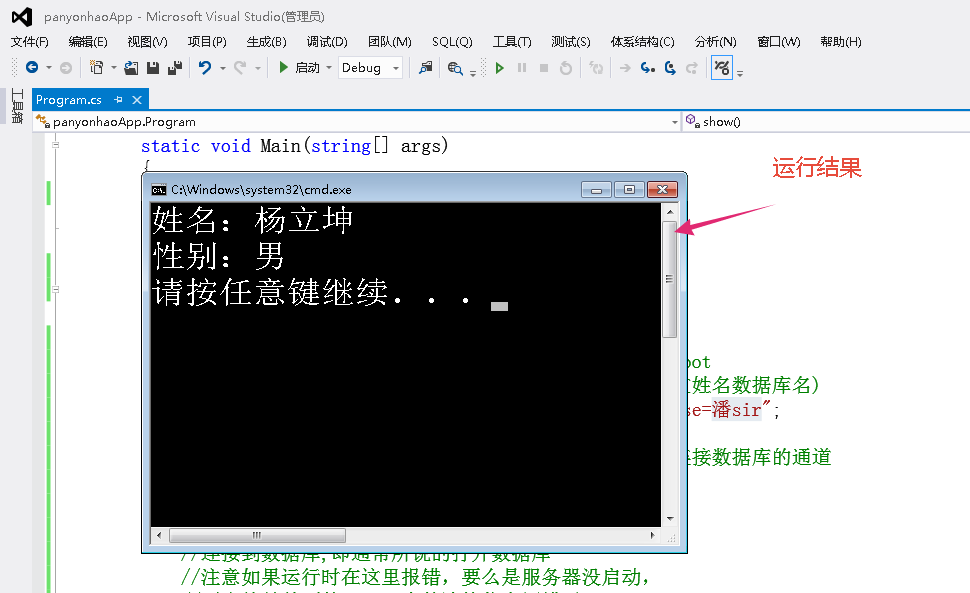
1. 执行程序

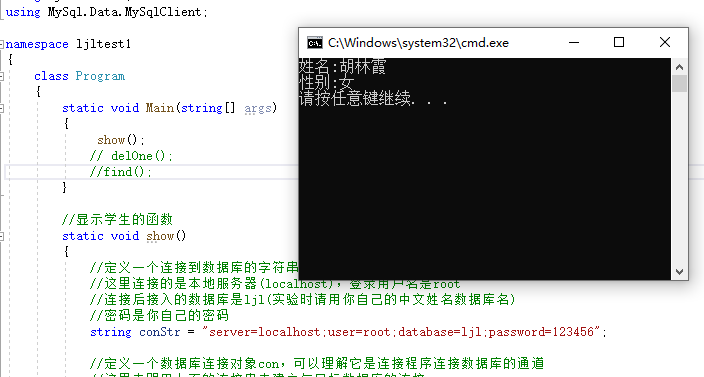
按**ctrl+F5**执行程序，如果程序有错，放心，一定是你输入的代码有错，注意是否输错了单词、使用了中文标点符号等常见错误。请仔细阅读出错信息，以帮助自己查找错误的原因及位置。



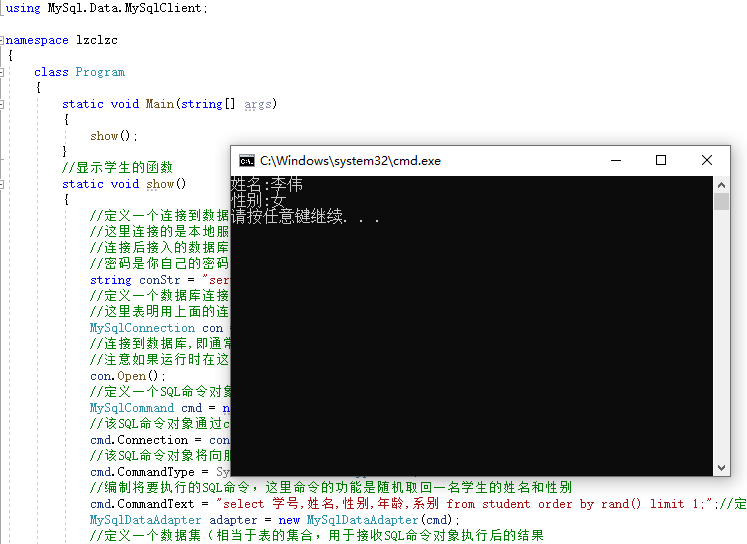


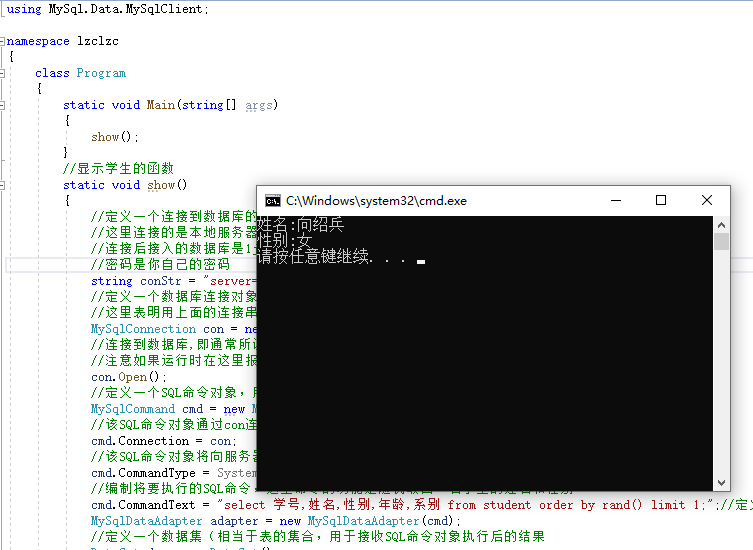
如果程序没错的话，则可以看见如下所示的结果（注意由于是随机抽取的一位学生，所以每次运行的结果是不同的，这里仅作参考）：

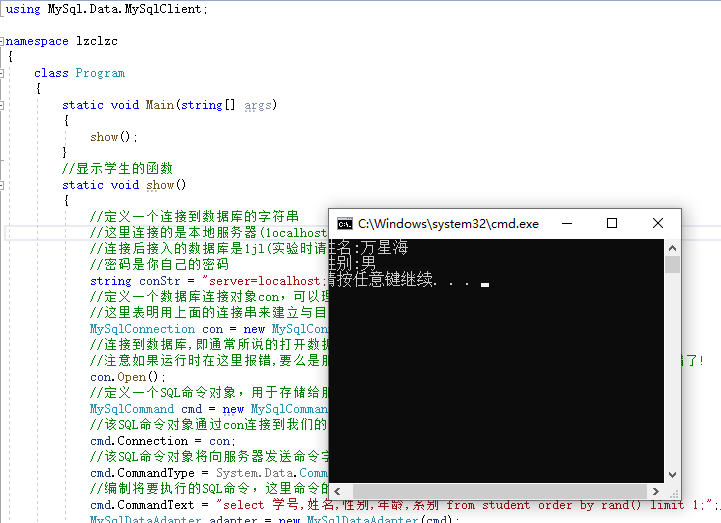




连续3次执行你的程序，将3张执行情况图粘贴在下面（要求跟我的参考图布局一样，将执行窗口和代码窗口一块儿完整截取下来）：

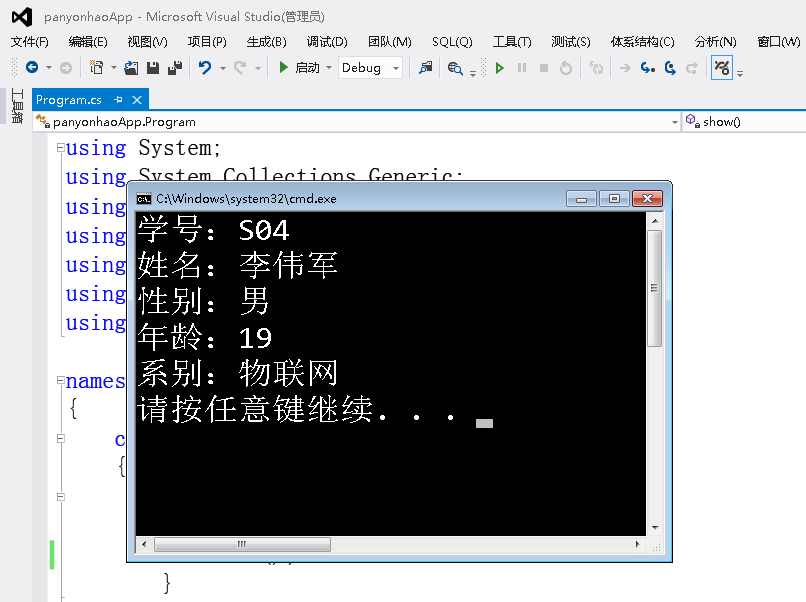






改造上述程序，给它增加同时显示学生的学号、年龄、系别的功能。

参考结果：



提示：取回学生的信息是程序中那条select命令串来决定的，取回多少信息，后面就可以writeline显示多少信息。

你改造的程序的源码：

//显示返回的姓名

System.Console.WriteLine("姓名:{0}",ds.Tables[0].Rows[0]["姓名"]);

//显示返回的性别

System.Console.WriteLine("性别:{0}",ds.Tables[0].Rows[0]["性别"]);

//显示返回的学号

System.Console.WriteLine("学号:{0}",ds.Tables[0].Rows[0]["学号"]);

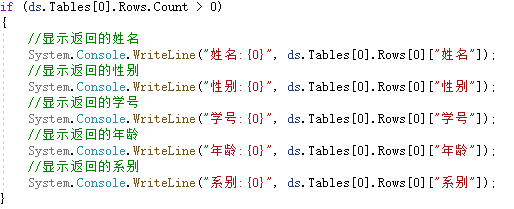
//显示返回的年龄

System.Console.WriteLine("年龄:{0}",ds.Tables[0].Rows[0]["年龄"]);

//显示返回的系别

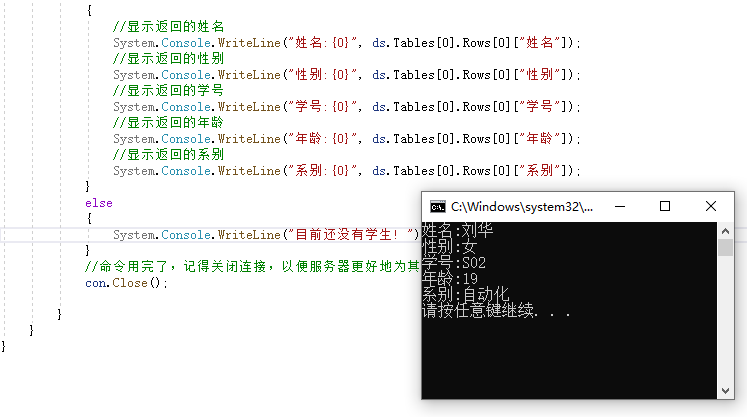
System.Console.WriteLine("系别:{0}",ds.Tables[0].Rows[0]["系别"]);

源代码截图：



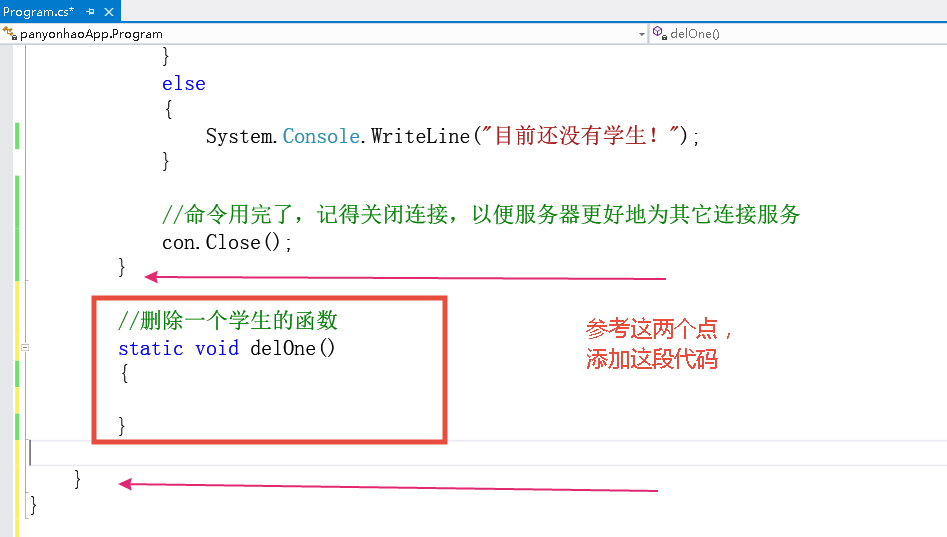
你改造的程序的3次成功执行情况图：







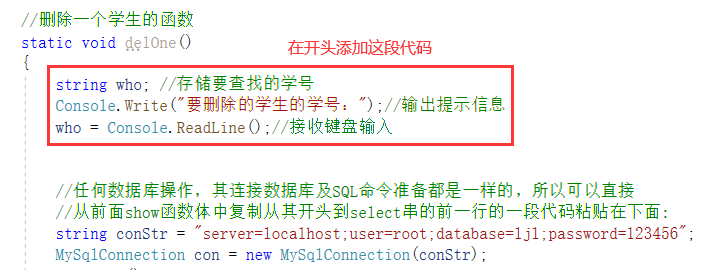
在show函数的后面，再构造一个delOne函数



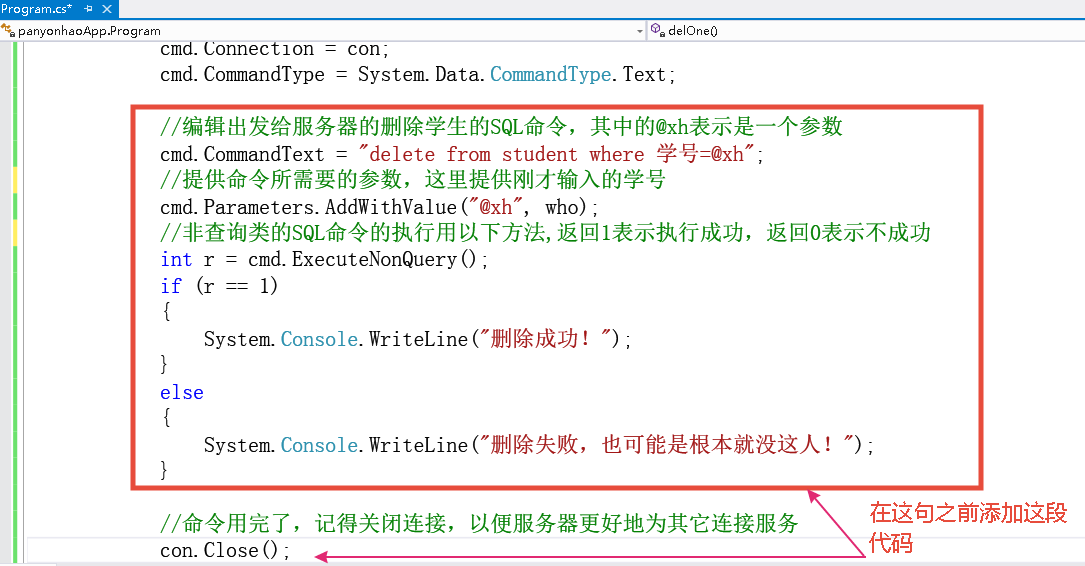
按提示在delOne函数体中粘贴或编写出如下的代码：



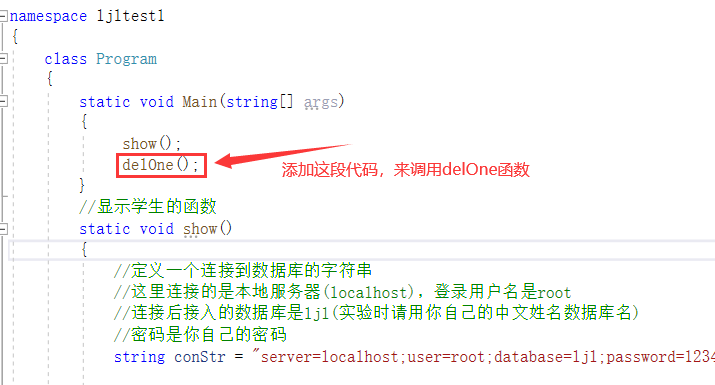
在delOne函数体开头添加代码：



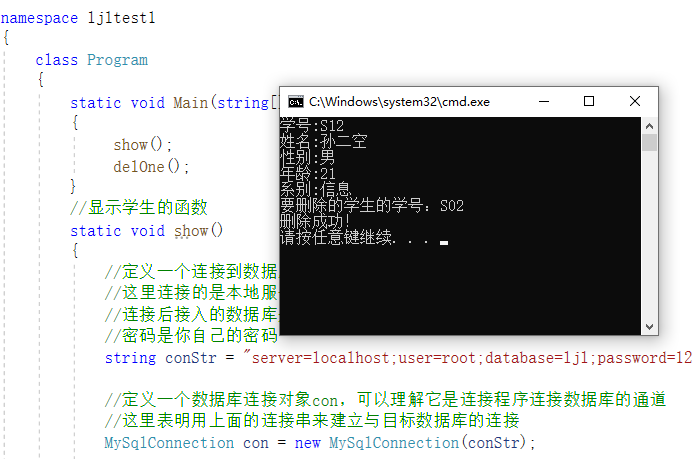
在delOne函数体中部添加代码：



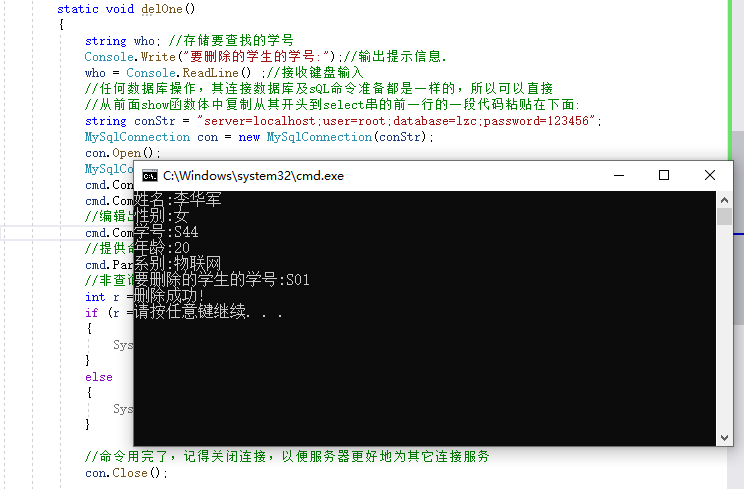
回到在主函数Main的函数体中，添加调用delOne函数的代码：



按**ctrl+F5**执行程序，有错就改错，没错的话，程序运行结果如下。输入这个显示出来的学号回车，应该能正常删除该学生（不能正常删除的话，说明程序 中delete那条命令有错，去改之）：

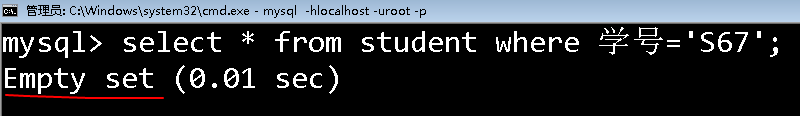


你的程序的成功执行情况图（截图要求跟我的图布局一致）：



如何真正验证是否已经删除了该学生呢？可以再在程序中写函数来实现。这里我们用一种原始的方法，即在MySQL 5.5 Command Line Client控制台中用select查询来验证，

即用：



查询刚才删除的学号的学生还在student表里吗。如果还在，说明程序是有问题的。如果如上图所示不存在这个学号，则表示已经删除了，程序是OK的。

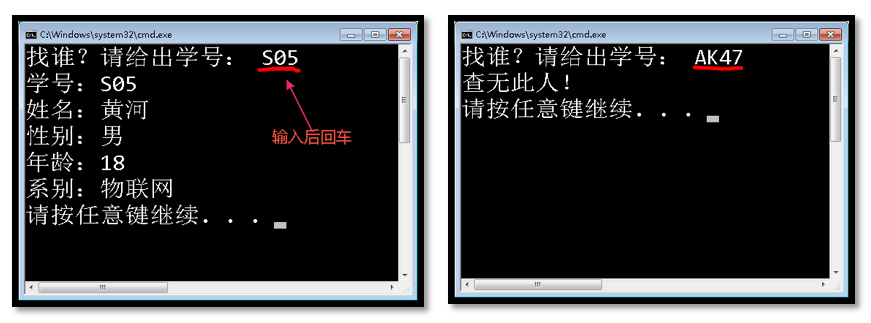
你验证是否删除成功的执行情况图：



1. 编程实现按学号查询学生的功能

可以就在本项目代码的基础上改，或增加find函数及调用来实现，或者新建一个项目来实现。

参考结果如下 ：



**你的核心代码：**

主函数里面加一个find()

static void find()

{

string who; //存储要查找的学号

Console.Write("找谁？请给出要查询的学生的学号：");//输出提示信息.

who = Console.ReadLine();//接收键盘输入

//任何数据库操作，其连接数据库及sQL命令准备都是一样的，所以可以直接

//从前面show函数体中复制从其开头到select串的前一行的一段代码粘贴在下面:

string conStr = "server=localhost;user=root;database=lzc;password=123456";

MySqlConnection con = new MySqlConnection(conStr);

con.Open();

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = con;

cmd.CommandType = System.Data.CommandType.Text;

//编辑出发给服务器的删除学生的SQL命令，其中的@xh表示是一个参数

cmd.CommandText = "select 学号,姓名,性别,年龄,系别 from student where 学号=@xh; ";

//提供命令所需要的参数，这里提供刚才输入的学号

cmd.Parameters.AddWithValue("@xh", who);

//非查询类的SQL命令的执行用以下方法,返回1表示执行成功，返回0表示不成功

MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);

//定义一个数据集（相当于表的集合，用于接收SQL命令对象执行后的结果

DataSet ds = new DataSet();

//用适配器执行命令来填充ds数据集，即将取回的数据装入ds

adapter.Fill(ds);

//前面的SQL命令成功取回学生的姓名和性别后会放入ds的tables[0]表的rows[0]行//没有的话，则tables[o]表的rows的行数为0

if (ds.Tables[0].Rows.Count > 0)

{

//显示返回的姓名

System.Console.WriteLine("姓名:{0}", ds.Tables[0].Rows[0]["姓名"]);

//显示返回的性别

System.Console.WriteLine("性别:{0}", ds.Tables[0].Rows[0]["性别"]);

//显示返回的学号

System.Console.WriteLine("学号:{0}", ds.Tables[0].Rows[0]["学号"]);

//显示返回的年龄

System.Console.WriteLine("年龄:{0}", ds.Tables[0].Rows[0]["年龄"]);

//显示返回的系别

System.Console.WriteLine("系别:{0}", ds.Tables[0].Rows[0]["系别"]);

}

else

{

System.Console.WriteLine("查无此人! ");

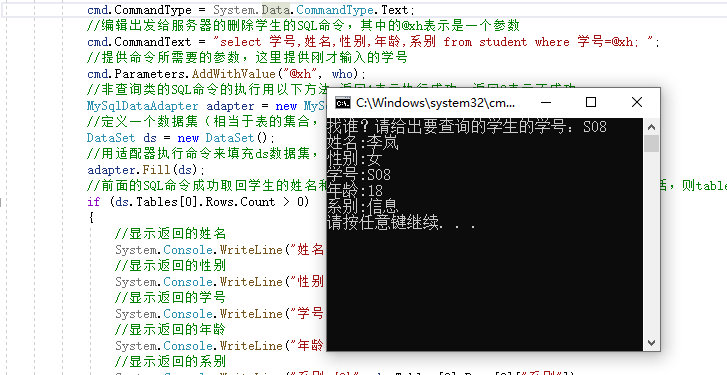
}

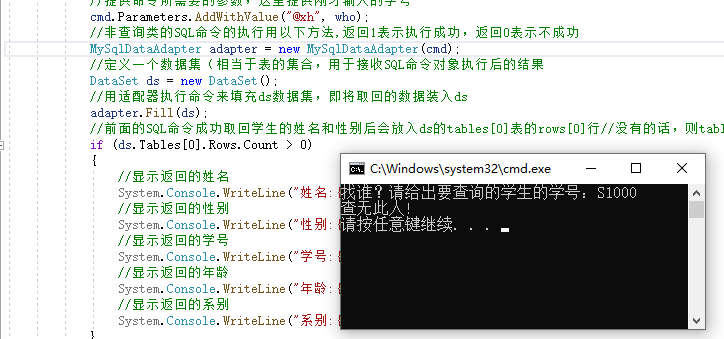
//命令用完了，记得关闭连接，以便服务器更好地为其它连接服务

con.Close();

}

**你的运行结果：**





**如果有兴趣有时间，还可以自行扩展出其它的数据库应用程序，比如list列表显示出所有学生，增加学生等等等等**

1. 关闭VS 2019

【实验结果】

**请将完成后的本实验报告提交到课程平台。**