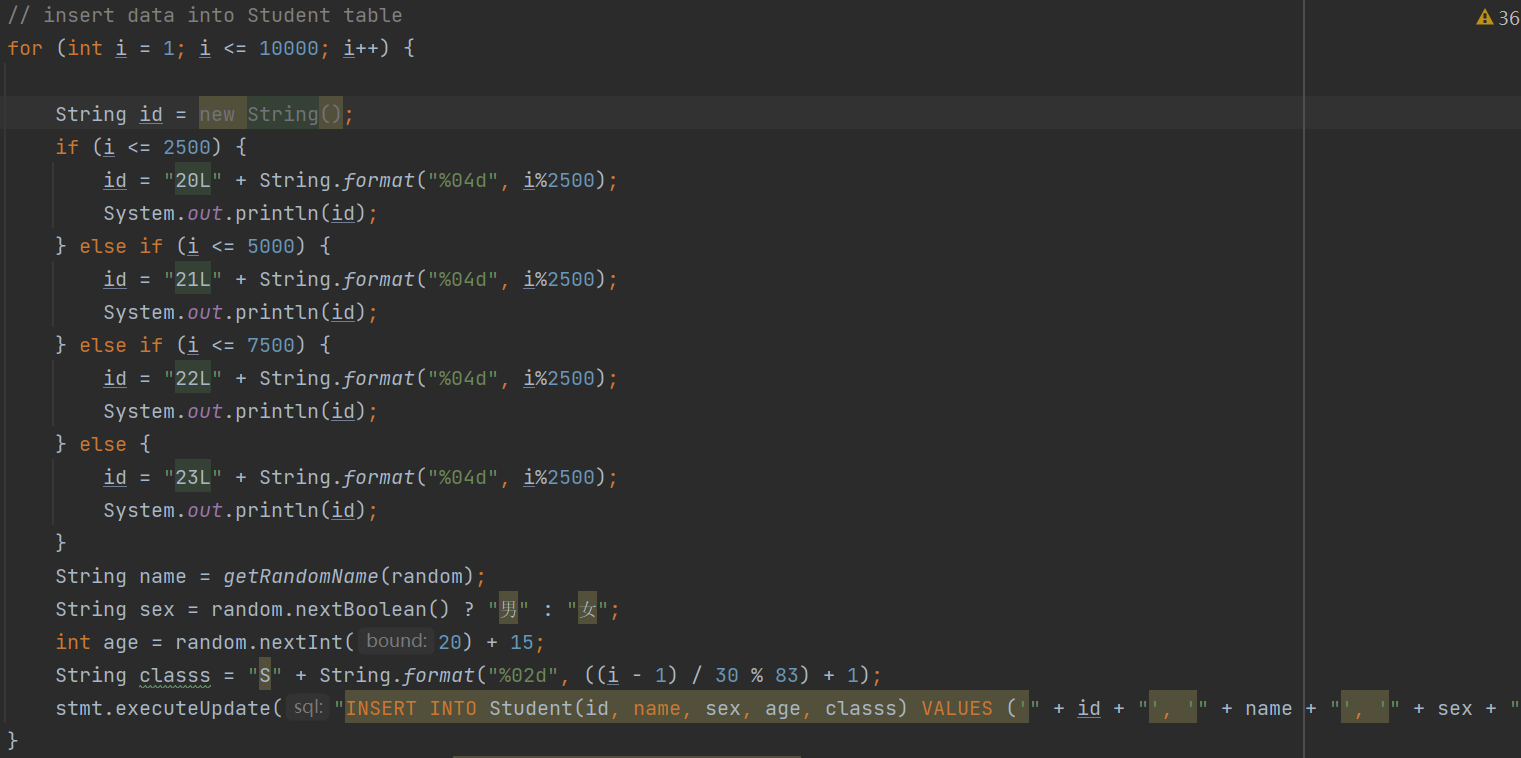
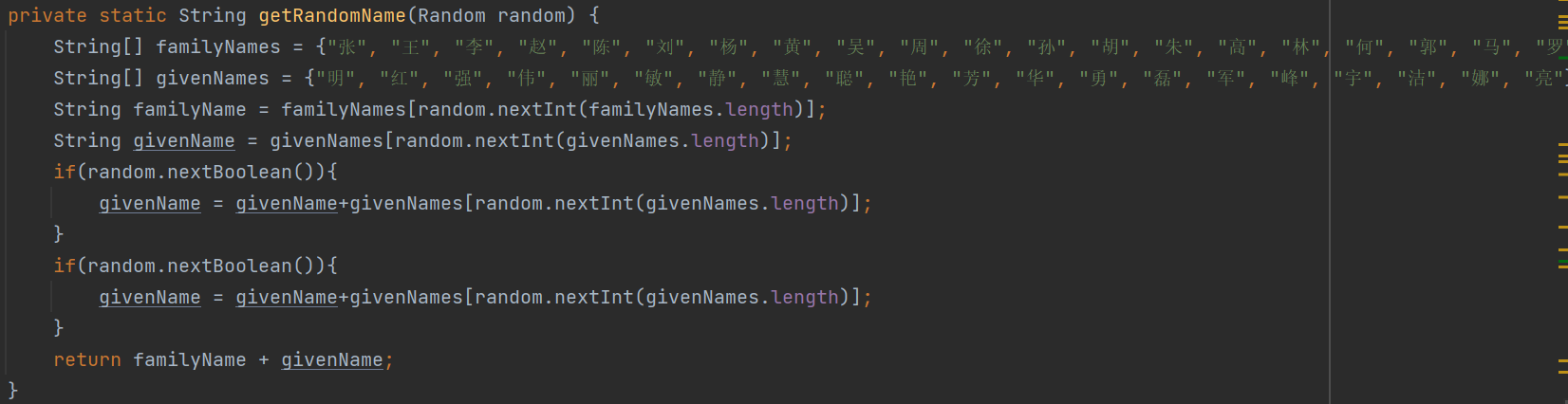
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2023年春季数据库系统**  **实验一** | | | | |
|  | | | | |
| 关系数据库应用系统开发实践 | | | | |
|  | 姓 名： | ： | 孙莹 |  |
| 学 号： | ： | 120L020414 |
| 班 级 ： | ： | 软件工程2037101 |
| 联系方式： | ： | 1099384634@qq.com |
| 二〇二三年4月 | | | | |

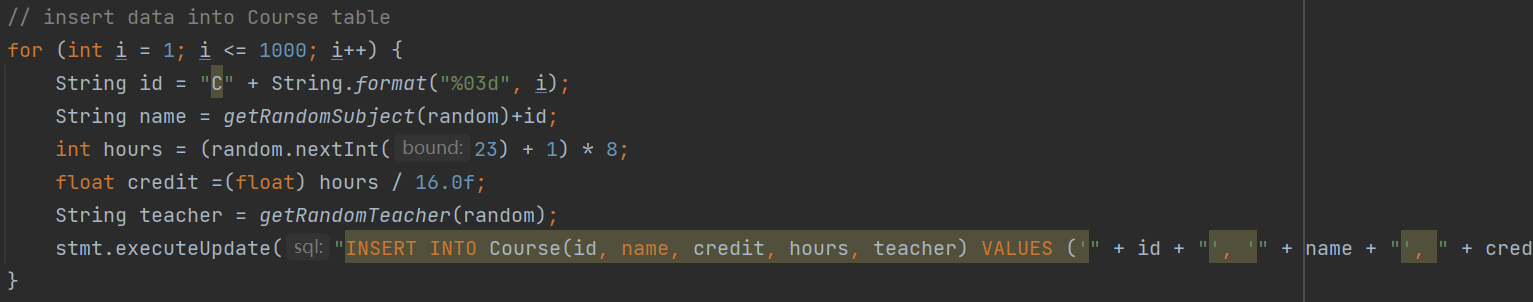
**一、数据结构**  
**1.库表设计**  
 我使用的关系数据库为“学生成绩管理数据库”，名为studentmanagement,该数据库中有三个表，分别对应学生信息（student）、课程信息（course）、成绩信息（sc）。  
 学生信息，包括学号（主键），学生姓名、学生性别、学生年龄及学生所在班级。所以Student表设计为(id char(8) Primary key, name char(10), sex char(2), age integer, classs char(6))。【如果用class，在java语言中处理不方便，所以我用的是“classs”】

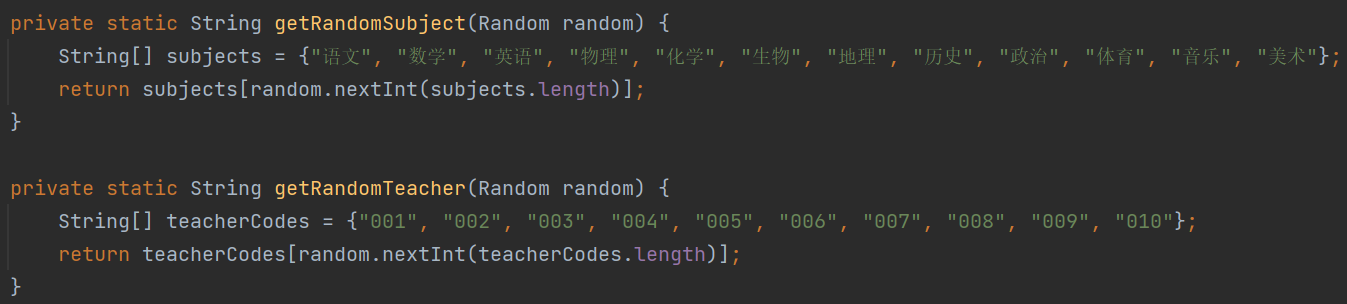
课程信息，包括课号（主键）、课程名称、课程学分、课程学时、授课教师。所以Course设计为(id char(8) Primary key, name char(30), credit float(1) , hours integer, teacher char(3))

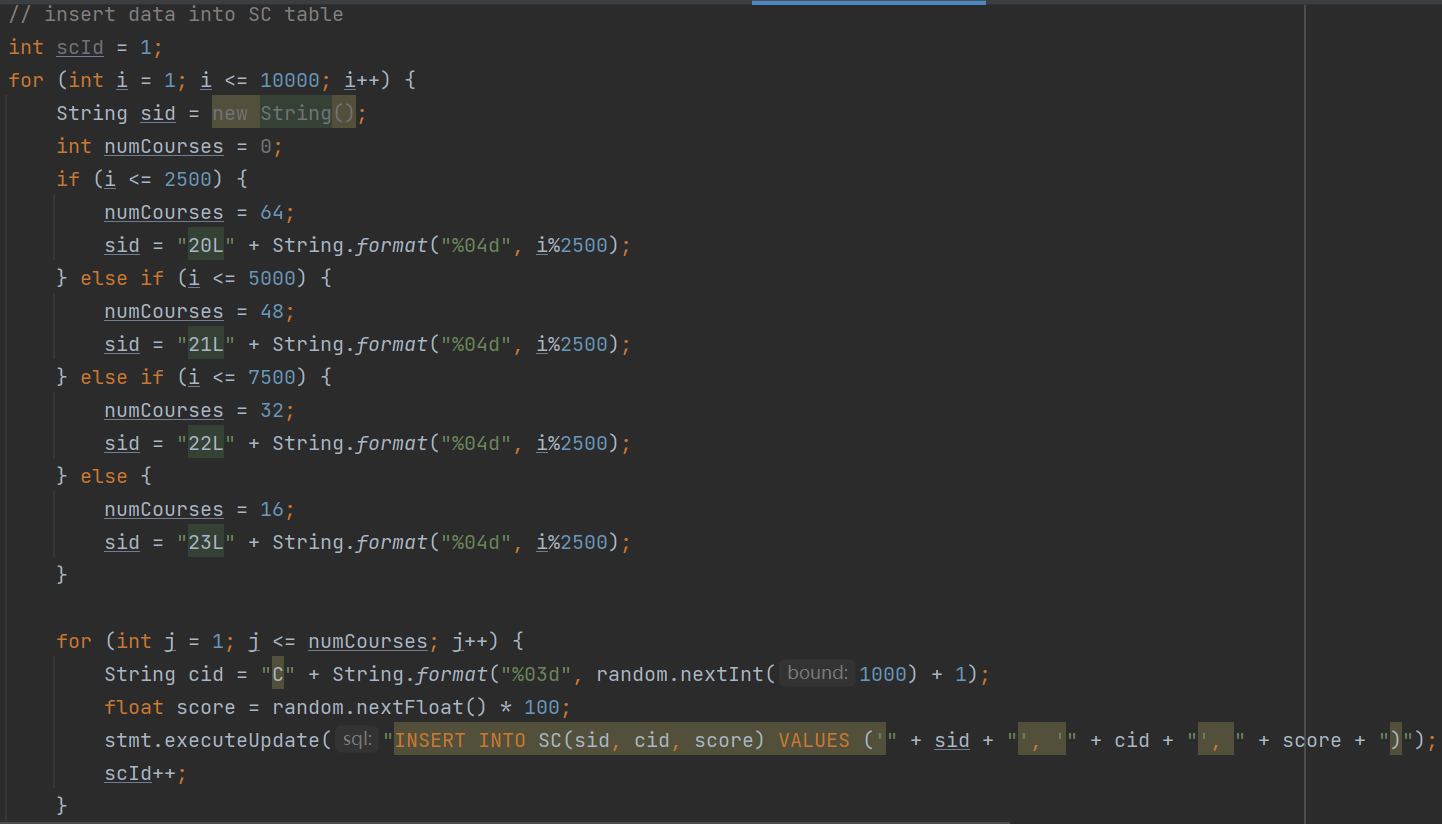
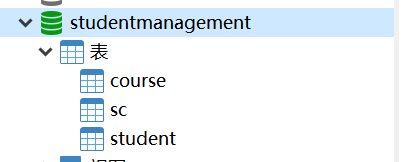
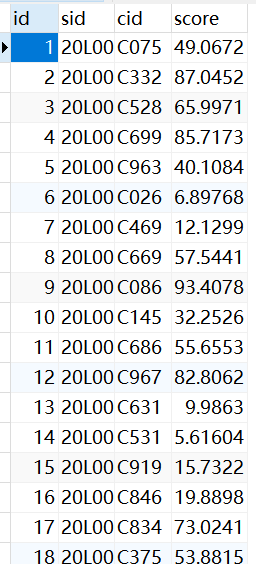
成绩信息，包括该成绩对应学生的学号、对应课程的课号、得分。所以SC被设计为(sid char(8) REFERENCES student id, cid char(4) REFERENCES course id, score float(1))。   
**2.记录生成** 学生生成：需要生成10000名学生，我让20级、21级、22级、23级分别有2500名。学生姓名由getRandomName函数生成，由随机生成1个字的姓和1~3个字的名组合得到；学生性别随机选择“男”或“女”；学生的年龄需要在15~35之间，由生成一个0~20的随机整数后加15得到；每个年级有若干个班级，每个学生随机分配一个班级号。

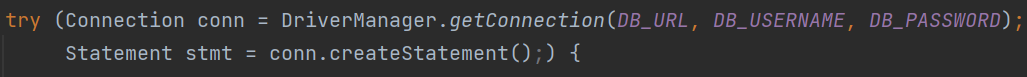
课程生成：需要生成1000门课程。每门课程的课号形如“C001”；课程的名称由课程类别（随机选择）和课号组合而成；课程的学时需要在8-192之间8的倍数，所以我先生成一个在1~24之间的整数，再乘以8得到学时；学分需要和学时成正比，所以学时除以16得到学分。



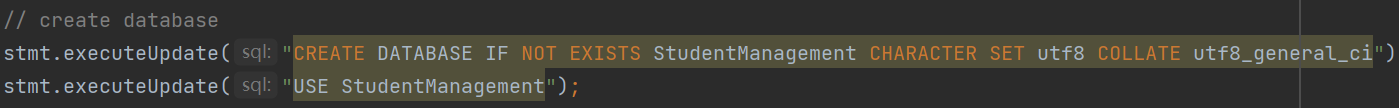
  
 成绩生成：结合前面学生信息的设计和成绩生成的要求，20级学生需有64门课的成绩，21级有48门课的成绩，22级有32门，23级有16门。成绩对应的课号、分数使用随机生成数得到。

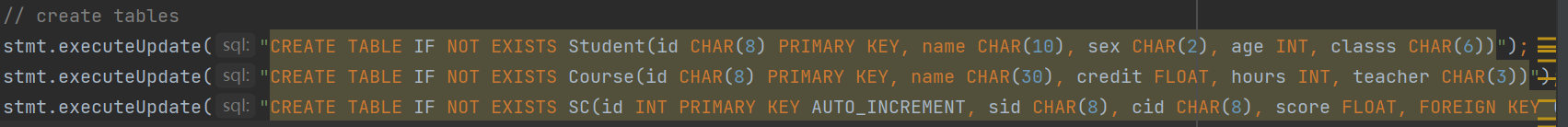
  
**3.结果展示**  
 使用数据库可视化软件Navicat可查看库表设计、结果展示的结果如下：  
 1）数据库：  
  
 2）course表、sc表、student表  
  

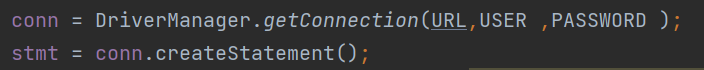
1. **设计与实现**

实验指导书中，很多功能分为好几步实现，例如DB/Table Building🡺Cursor/RecordSet🡺List，这一过程由编程8完成； 而List🡺Screen Output由编程5完成，在实验报告中我不再分步骤写，以免思路不清晰。  
**1. X1：数据的变化与基本控制【数据结构】**  
1）数据库创建  
 在进行数据库的创建时，我们首先需要定义好本机数据库的位置、用户名、用户密码，定义之后将其作为参数尝试连接本机数据库。  


连接成功后创建数据库，注意需要定义中文编码统一为UTF8，防止后面插入数据时出现中文乱码。然后使用 SQL语句进行数据库的创建

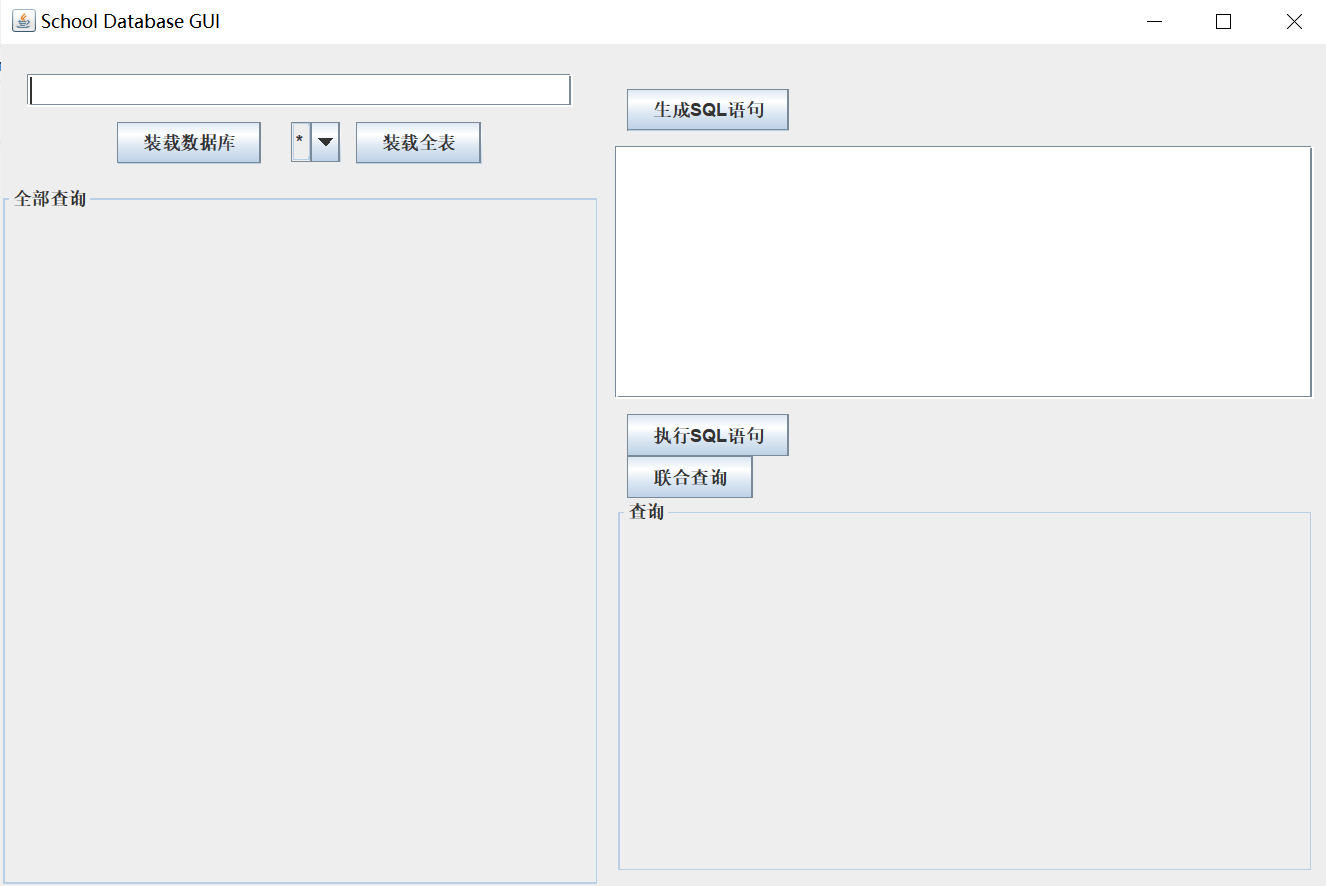


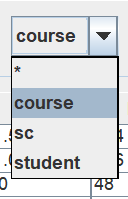
1. 数据表创建  
    在数据表的创建方面，我们需要确定表名称、表中字段的名称、类型、长度等信息，同时需要考虑设置主键、外键以及其他约束条件，然后使用SQL语句进行创建。  
     
   3）数据库连接与断开  
    创建完数据库、数据表后，就可以使用本机数据库地址、用户名、用户密码、数据库名连接该数据库，进行增删改查操作了。需要设置以下serverTimezone和useSSL否则会报错。我使用JDBC驱动连接。  
     
   4）游标的使用  
    使用JDBC驱动连接数据库，支持滚动游标，它能够在结果集中向前、向后或移动到某个特定的行。可以使用createStatement()方法创建一般的滚动游标。如下面代码的“stmt”即为游标。

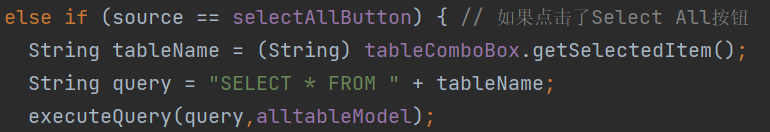


驱动程序也为JDBC API维护了游标，每当应用程序访问结果集中的一个新行时，驱动程序会通过网络将行数据传递回应用程序。在向前和向后移动CursorPosition时，驱动程序会根据需要检索所需的行并将其发送到应用程序。驱动程序还负责在关闭相关声明时，清除游标所需的资源，如打开的文件句柄和内存等。

所以程序中不需要我显式地手动使用游标，程序会自动创建和维护。

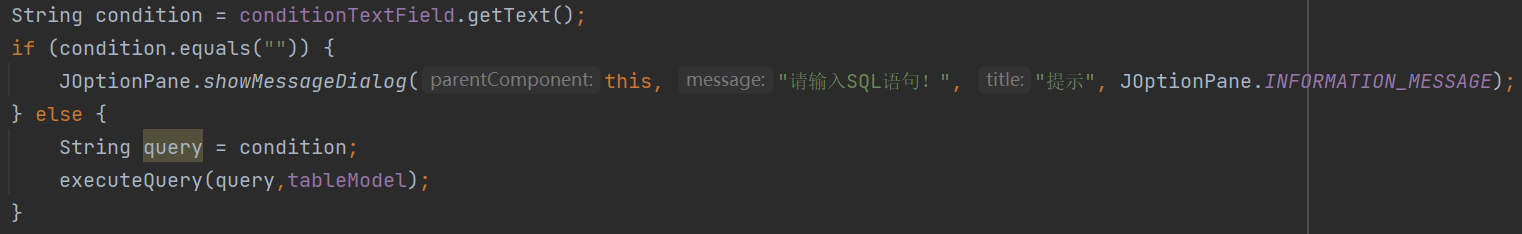
1. **X2：屏幕元素及其显示输出【用户界面】、X3：问题求解相关算法设计与函数实现【算法】**  
   1）界面设计  
    根据指导书，我初步设计应用界面如下。（后面还有丰富）  
     
   2）检索全表  
    选择想装载的表，点击“装载全表”按钮，程序从下面的选择框中获取表名后，调用executeQuery函数执行查询全表语句。



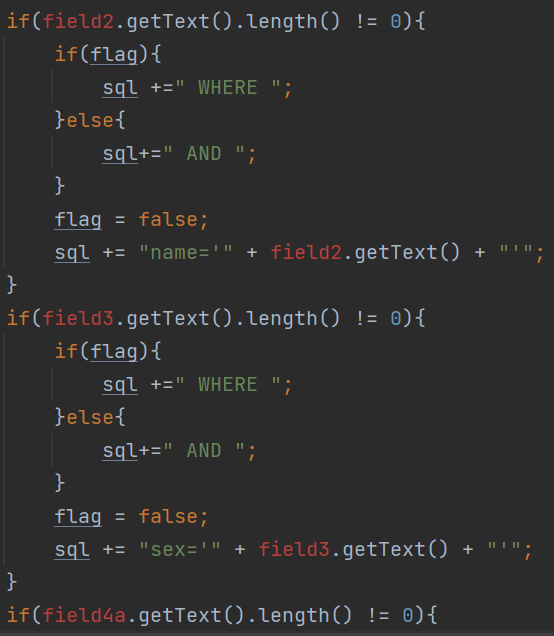
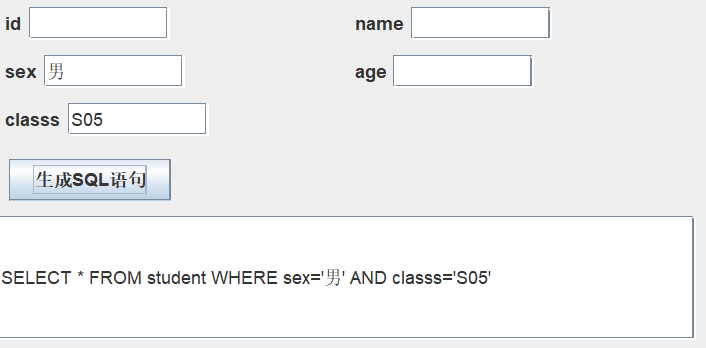


executeQuery函数使用上面的连接数据库时创建的stmt，调用其executeQuery功能。根据返回的数据，将所有的列名放入columnNames动态数组中，所有的数据放入data动态数组中，交由数据显示组件TableModel显示。

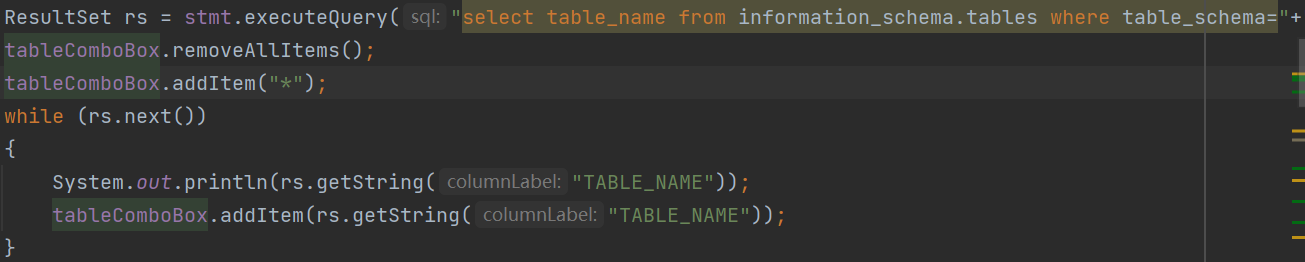
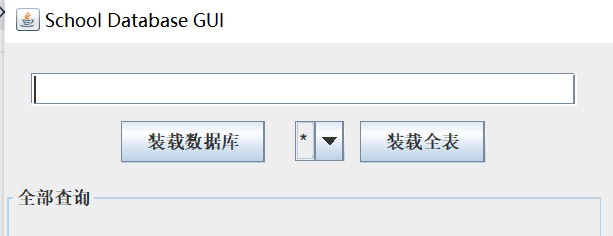


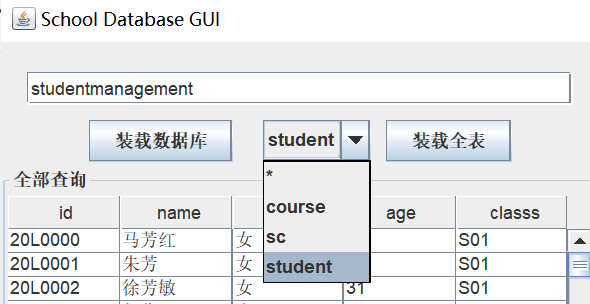
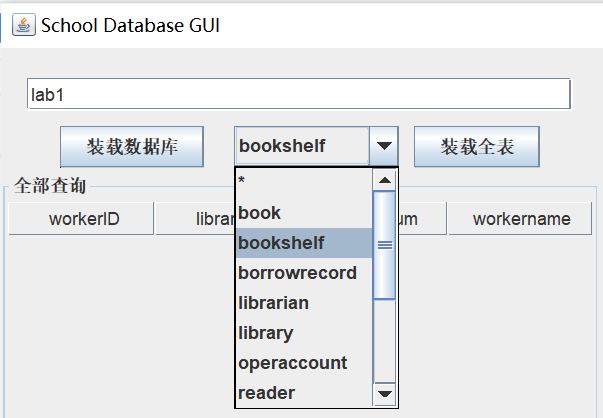
3）复杂SQL查询语句的运行  
 在输入框中输入SQL查询语句，点击“执行SQL语句查询”后，程序将获取到的语句交给executeQuery函数进行处理。显示流程和上面检索全表的显示流程类似。  


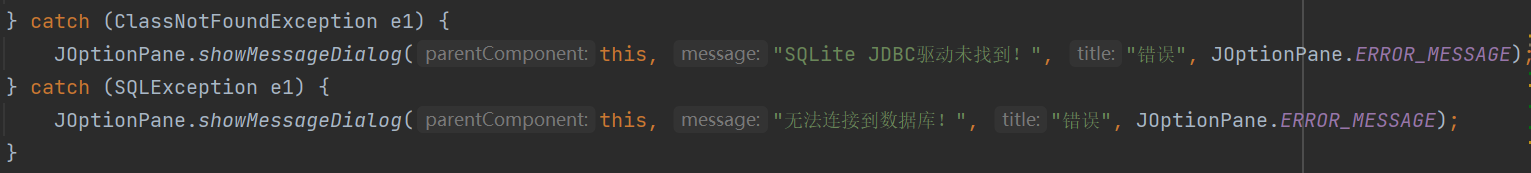
1. **X4：屏幕元素事件识别与控制【人机交互】**  
   1）交互式条件表达界面  
    对应student表的列名，设计对应的条件输入框。点击“生成SQL语句”按钮，程序会调用getSQL( )函数，将输入框中的条件按顺序组合，生成对应的查询语句。



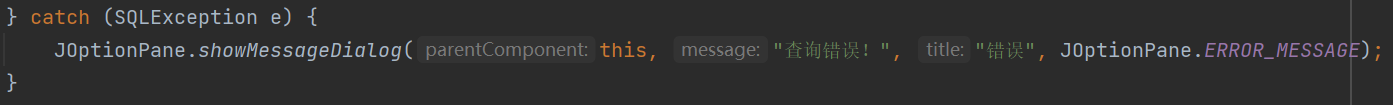
1. **X5：问题求解相关的算法优化与系统功能拓展【完整系统】**  
   1）装载任何一个数据库及其中的不多于3个的表  
    在前面连接数据时我已考虑到这一点，程序会从输入框中获取想连接的数据库名，然后和本机的数据库地址共同构成URL（如下），然后执行连接，这样就可以装载任意数据库。  
   String database = loadTextField.getText();

URL = URL + database + "?serverTimezone=GMT&useSSL=false";  
 连接后，查询该数据库中所有的表名，依次加入tableComboBox中，这样在后面就做到了可以选择装载此数据库中任意表。  
  
 如下，分别装载studentmangement和lab1数据库，会获取到不同的表名数据。  


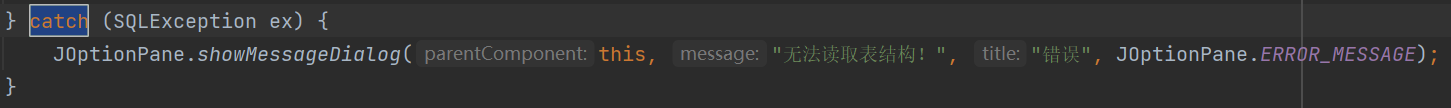
 

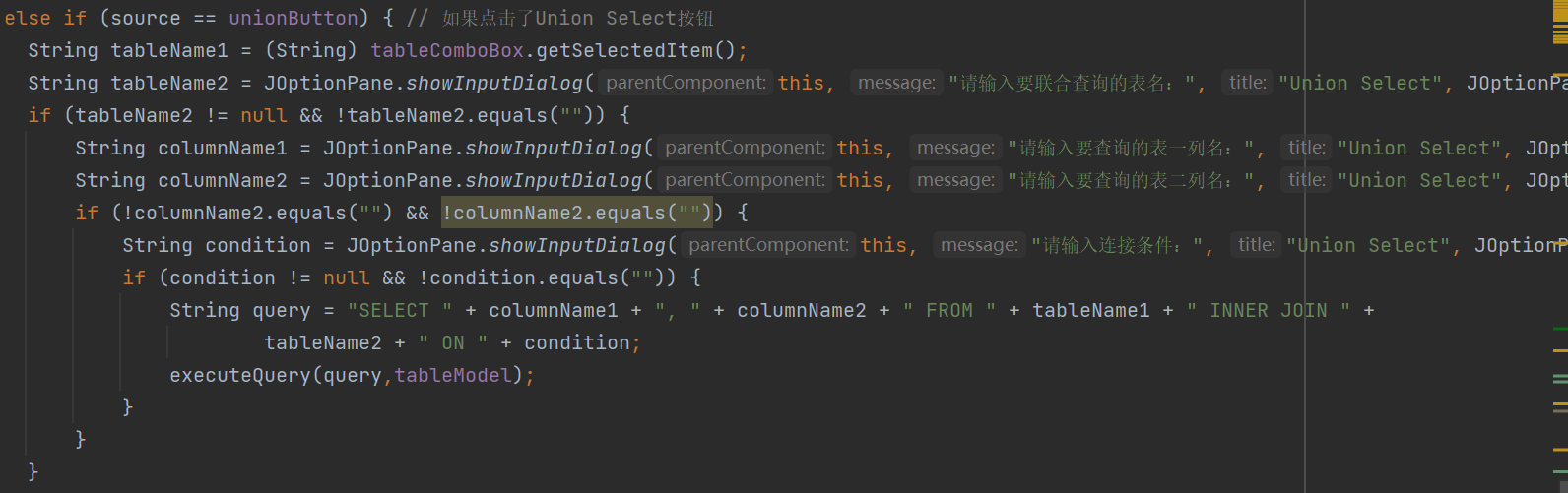
2）数据库访问状态监测与报警  
 连接、查询成功时，界面会正常显示。出现访问错误时，程序捕获错误信息，并以弹窗的形式显示。  
 连接数据库时，需要捕获数据库驱动异常、数据库连接异常。  


执行查询SQL语句时，需要捕获查询错误。

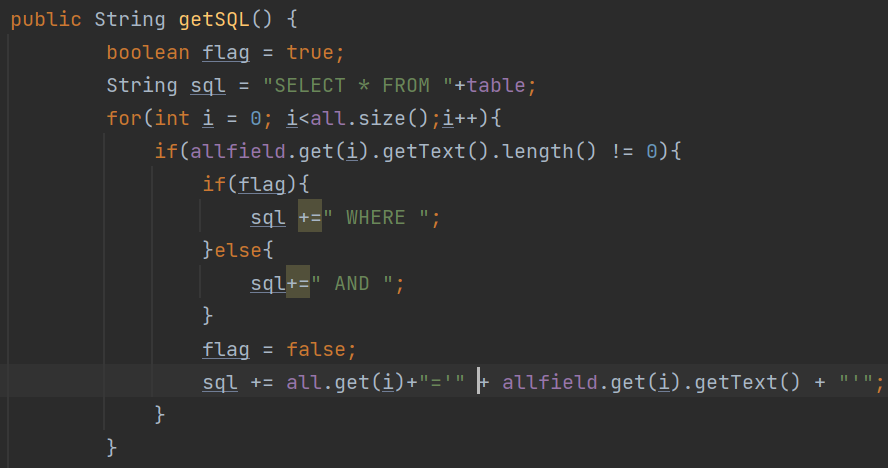


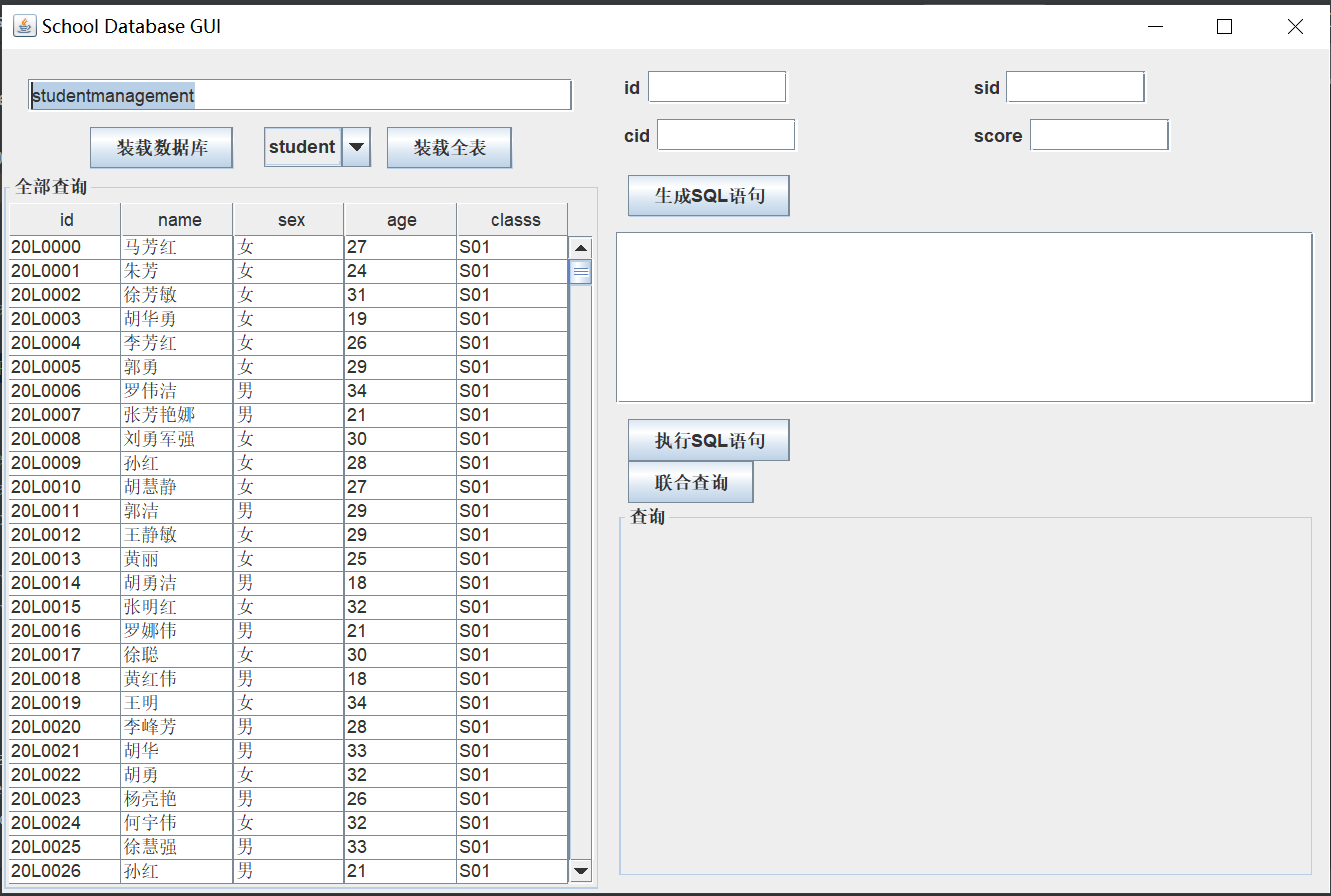
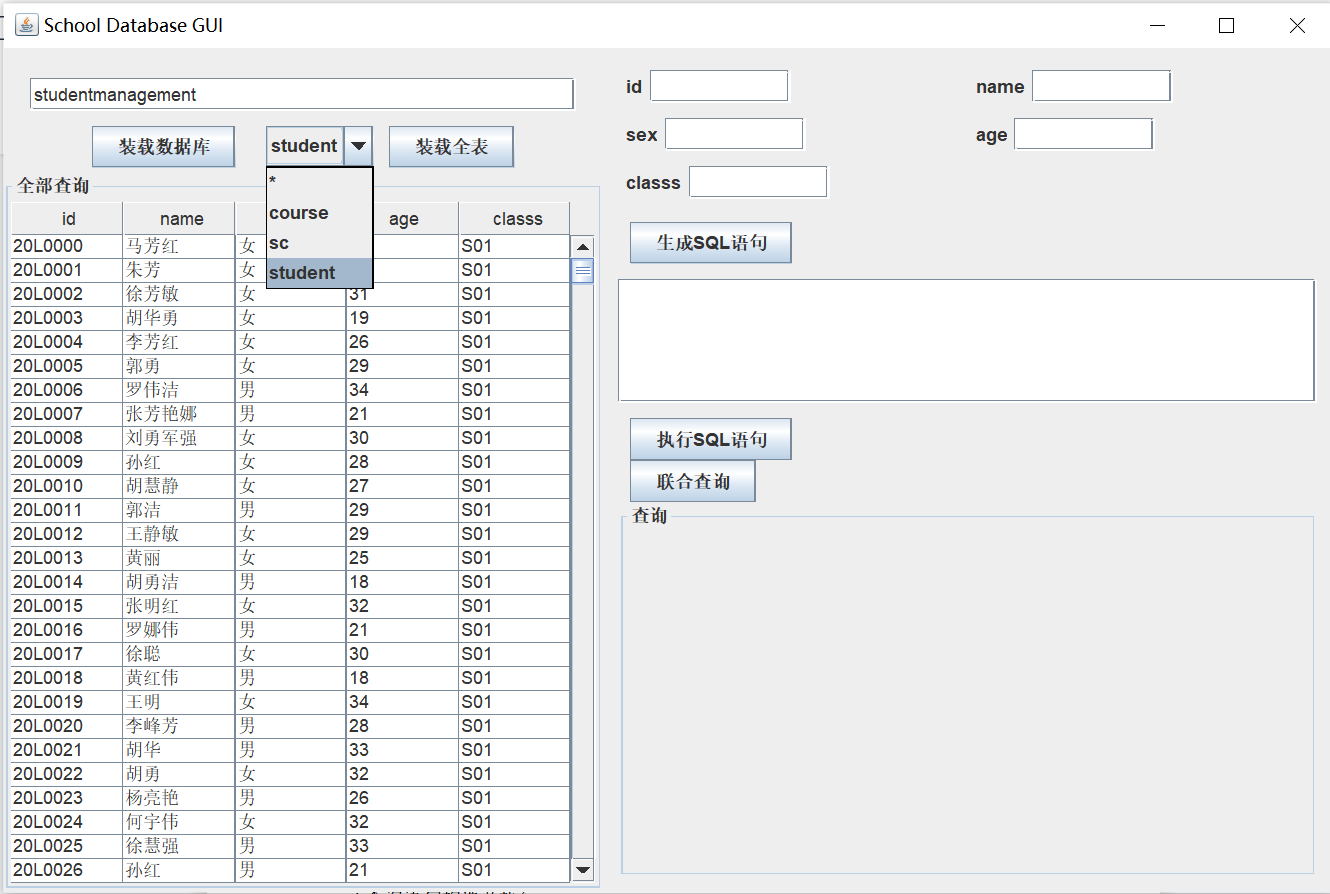
获取某个数据库的表名信息时，需要捕获无法读取表结构的查询错误。

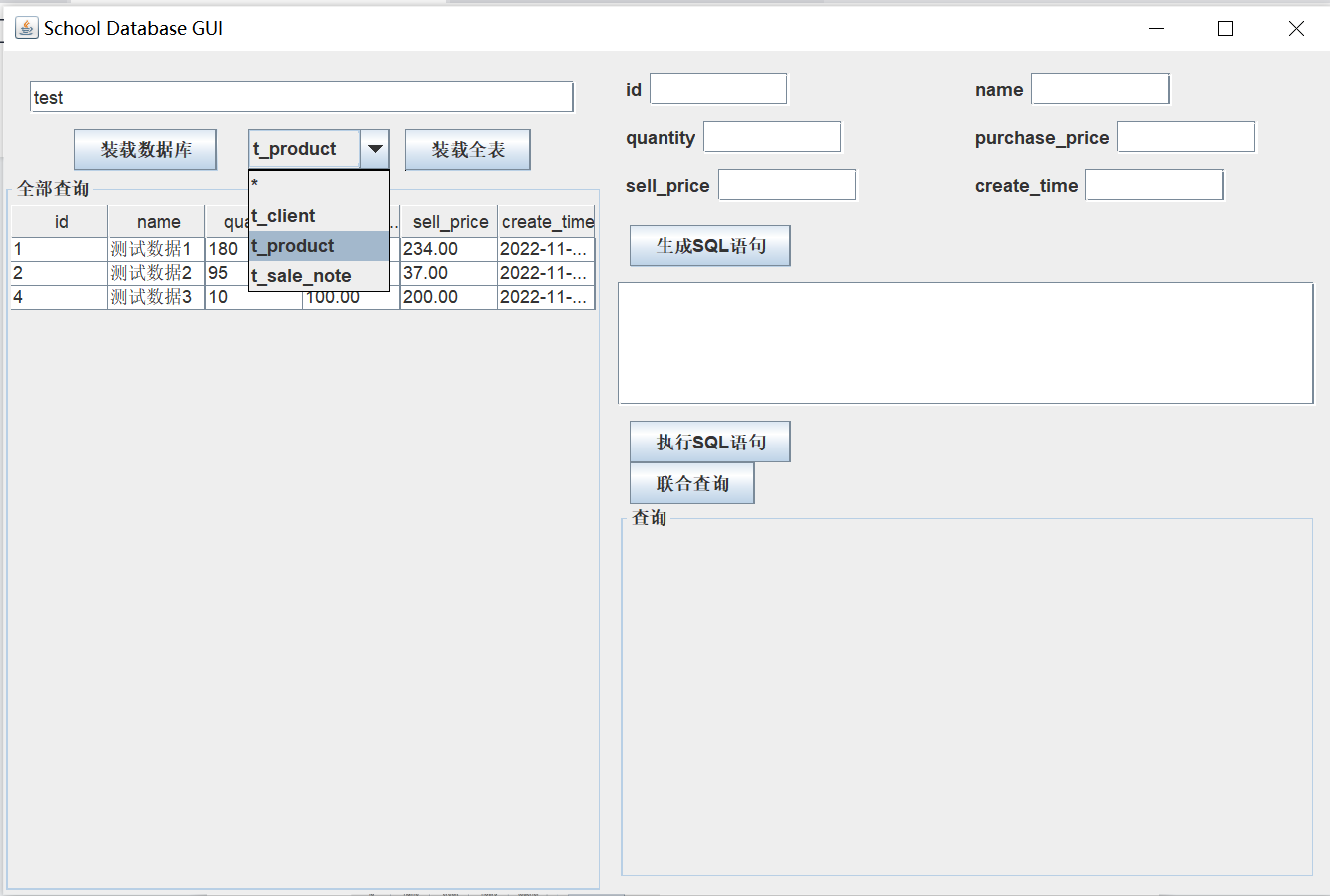
  
3）多表连接

点击“联合查询”按钮，程序会弹出弹窗，用户填入联合查询的表名、列名和联合查询条件，构造INNER JOIN的联合查询条件语句并执行。  
  
4）任何表/任何字段进行检索条件表

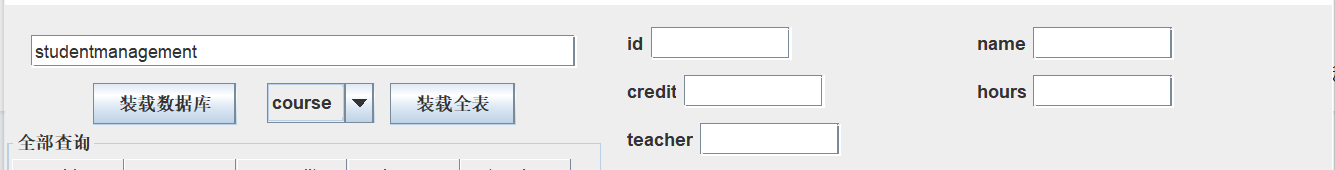
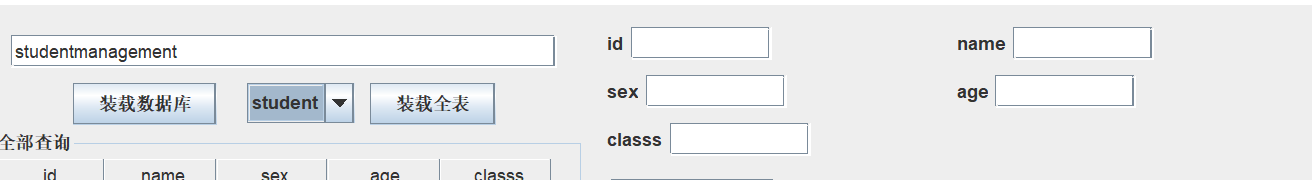
我重写了JPanel为一个QueryPanel，作为显示检索条件的组件。装载数据表时，将该表的所有列名作为参数调用update函数更新该QueryPanel，将所有可作为检索条件的输入框依次显示。同时重写getSQL函数，将其从原来连接固定的若干个检索条件变为可以连接任意个检索条件、构造SQL查询语句。



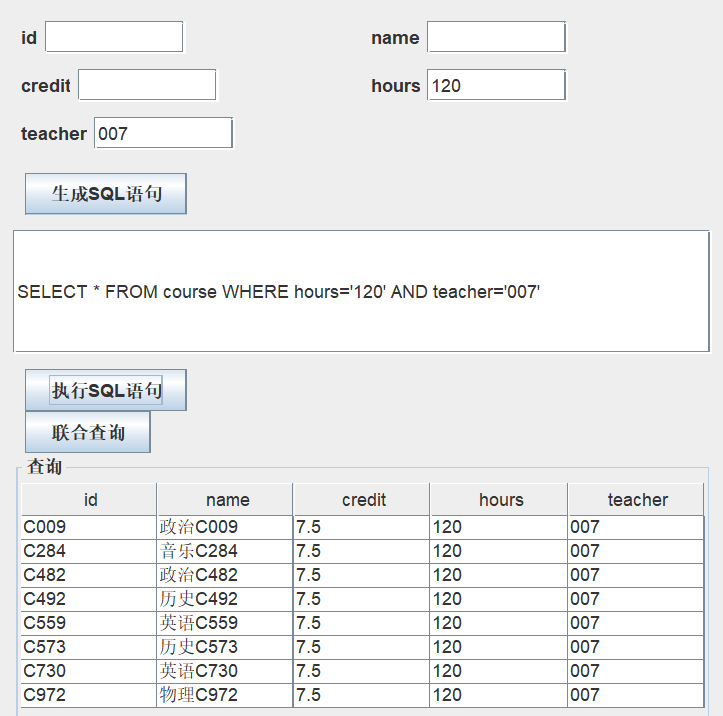
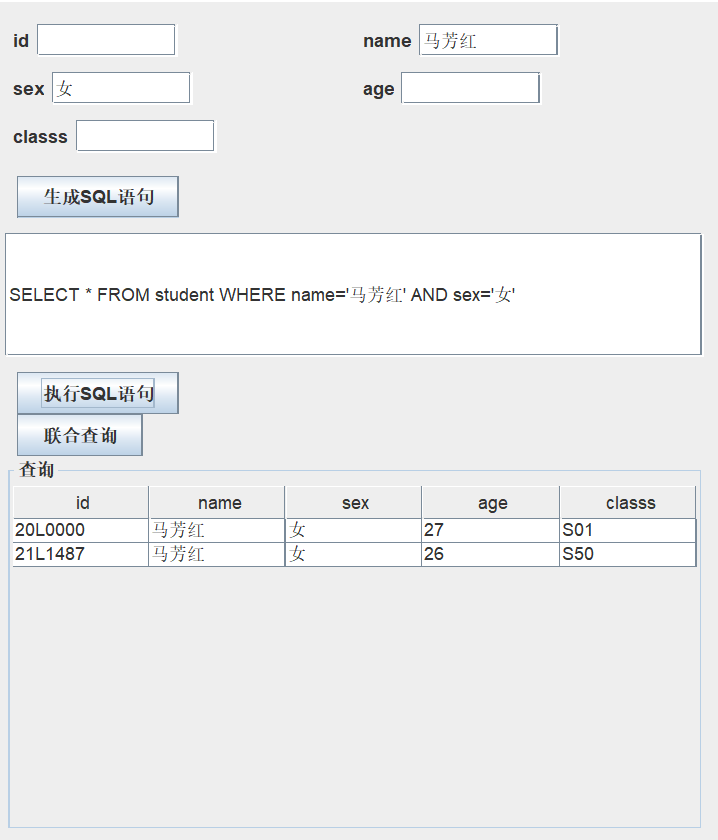
1. **运行结果**  
   **1.整体界面**  
     
   **2.装载任意数据库、任意表**  
   

  
**3.任意表的查询条件显示**

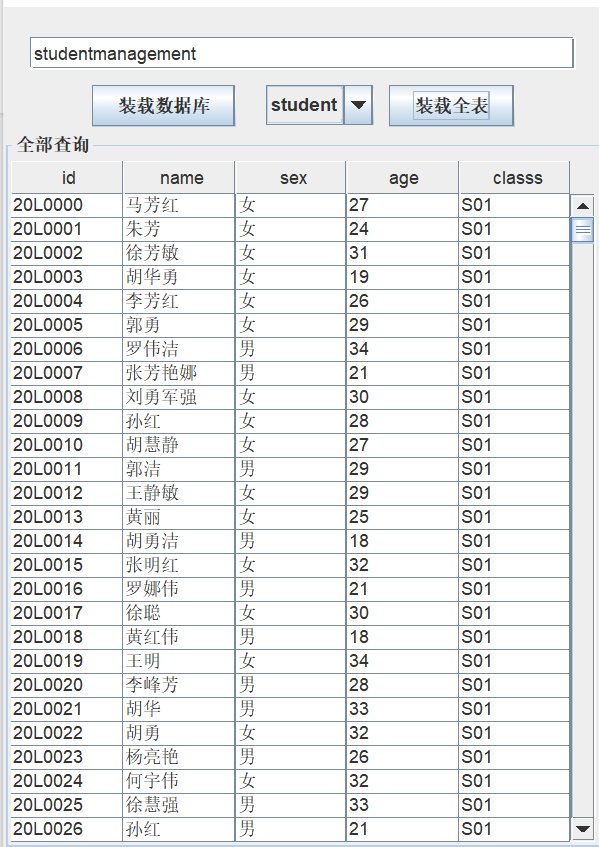
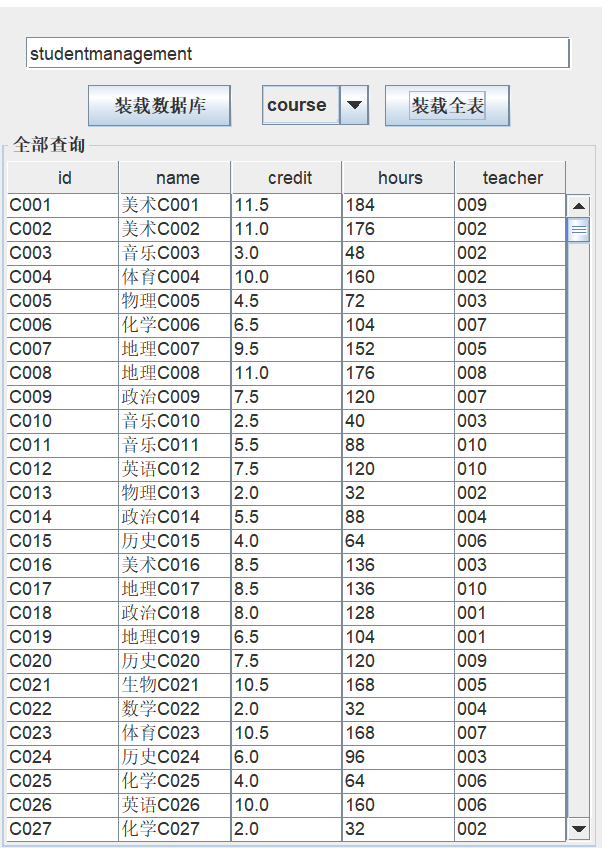
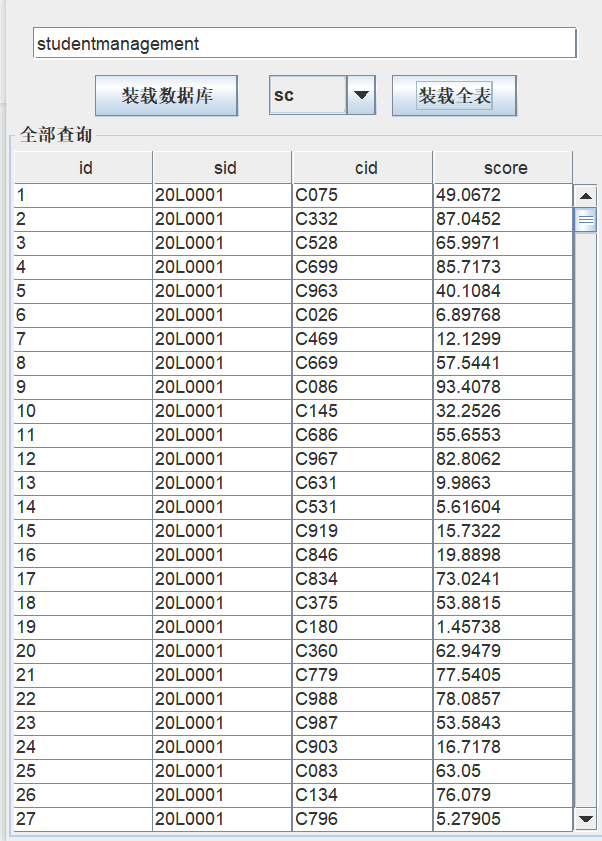
下面为student、course、sc表各自的查询条件显示。





**4.查询语句构造及执行** 

**5.查询表结果显示** 下面是student、course、sc表的查询结果显示

   **6.状态显示**  
