

数据库综合设计实验——图书管理系统

肖勇 PB18061258

2020 年 12 月 20 日

1 需求分析

1.1 题目要求

根据题目的设计要求，我们需要设计一个能对数据库进行数据查询，数据修改及执行存储过程的应用程序。其中数据查询能够无条件查、按书名查、按作者查并能按任意关键词查询。数据修改部分能进行插入、删除、更新。其中删除能够选中某一元组的某一属性进行删除则其他元组的该属性与其相等的元组均被删除，也可选中某一元组进行删除。关于存储过程即可以创建新的存储过程，也可以执行已经创建的存储过程。

1.2 数据字典

按照题目的要求，我们以图书为例，列出所需的数据项。

- 数据项：书名(title)
- 含义：图书的书名
- 类型：字符型
- 长度： 20
- 数据项：作者(author)
- 含义：该书的作者姓名
- 类型：字符型
- 长度： 20
- 数据项：价格
- 含义：图书价格
- 类型：字符型
- 长度： 20
- 数据项：索书号(title_id)
- 含义：唯一标识一本书
- 类型：字符型
- 长度： 20
- 数据项：出版社(press)
- 含义：该书的出版社
- 类型：字符型
- 长度： 20
- 数据项：出版日期(publish_time)
- 含义：出版的日期
- 类型：字符型

- 长度：20

数据结构定义为

- 数据结构：图书
- 含义：定义了一本书
- 包含：书名 title、作者 author、索书号 title_id、价格 price、出版社 press、出版时间 publish_time

其他类型的数据字典可以参照定义。由于本数据库较为简单，故对于数据存储、数据流图等不做具体设计。

2 概念结构设计与逻辑结构设计

根据题目要求，可以抽象出如下的实体及其属性

- 图书：书名 title、作者 author、索书号 title_id、价格 price、出版社 press、出版时间 publish_time

2.1 实体依赖模式

从实体依赖模式来看，图书的主键为图书号，其它各项均依赖于图书号。

2.2 数据完整性约束

由于本实验只有一张表，所以只用考虑主键约束即可

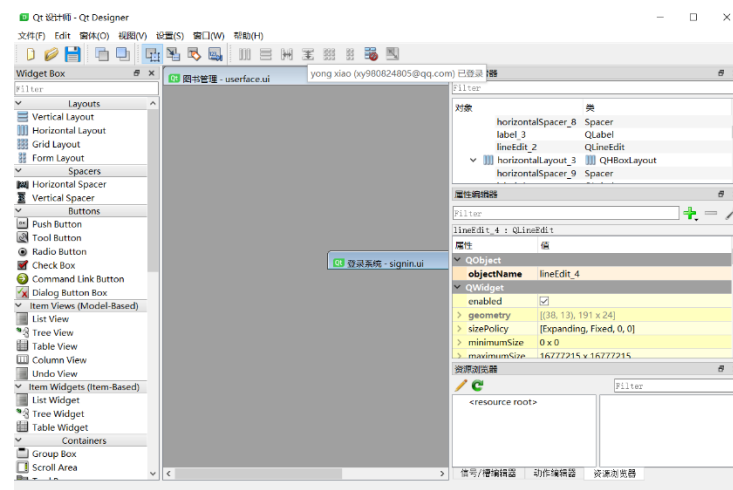
3 数据库物理结构设计

数据库的物理结构也十分重要，在本次设计中，采用应用比较广泛的服务器-客户端模式。由于数据量比较小，拟采用单文件形式的 SQLite 数据

4 数据库实施

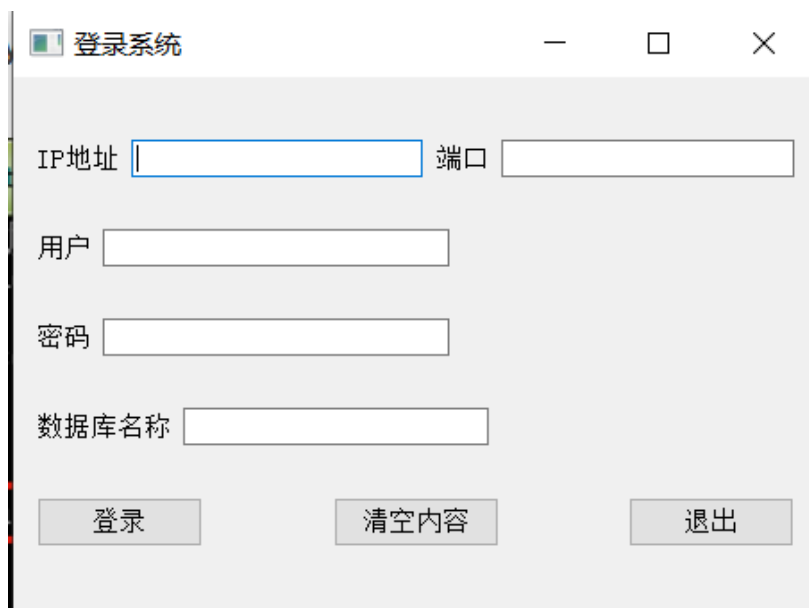
4.1 实施工具

本次数据库实验采用 python 进行编程，利用 python 中的 pymysql 库来连接 mysql 数据库并对其进行数据处理，而利用 PyQt4 库来进行图形化界面设计。进行图形界面设计首先利用 Qt designer 设计图形界面，再利用 pyuic4 将生成的结果转化为 python 程序，再在程序中对图形界面程序与数据库中数据进行处理。Qt designer 界面如下：




4.2 安全性——用户登录认证

利用 PyQt4 库中的 QtDesigner 设计登录界面，再利用 pyuic4 工具将其转化为 python 程序 `signin.py`，并对其该程序进行修改使其能按照要求连接数据库，其登录界面如下：

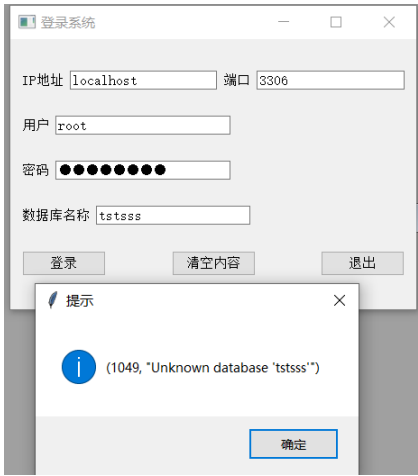


若点击退出，则程序结束。

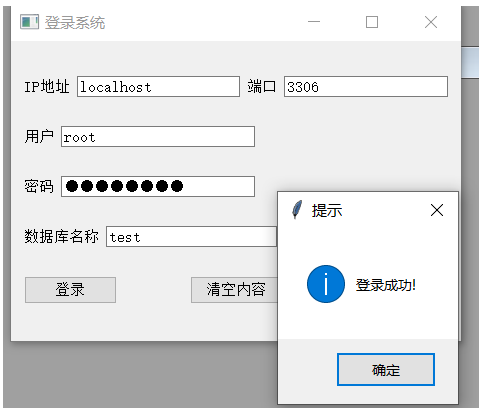
若输入相应的正确信息，点击登录即可进入主界面，若信息未填写完整则会进行提醒效果如下：



其中密码部分会自动以黑点覆盖，防止偷窥密码，符合日常设计。
点击清空内容，即可清空填写的所有内容，方便用户重新填写。
而若填入信息有误，则会以弹窗的形式弹出相应错误，提示用户进行修改。



若所填信息完整且正确，则会有弹窗提示登录成功，则可进入主界面，其效果如下：



4.3 主界面

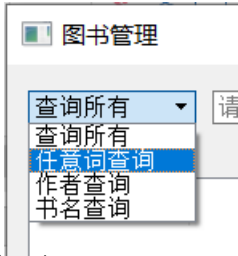
登录时所有信息正确则可登陆数据库，进入主界面。



主界面主要分为 3 个区域，区域 1 用来进行数据的增删改查及显示所得的操作结果，区域 2 用来输入创建或者执行存储过程的语句，区域 3 用来显示执行存储过程的结果。

4.4 图书查询功能

区域 1 中左边有一项用来选择查询方式共有四种：查询所有、任意词查询、作者查询、书名查询，默认为查询所有。



直接点击查询，查询所有结果如下：

图书管理

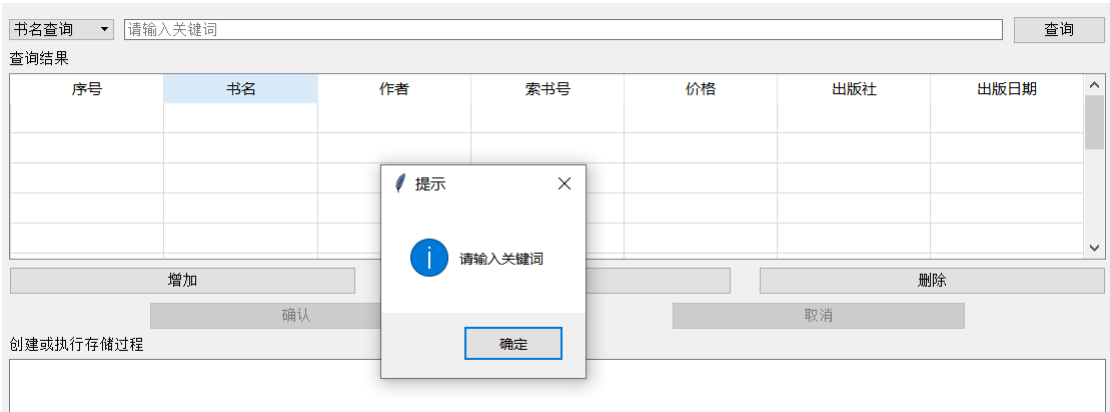
查询所有 请输入关键词 查询

查询结果

序号	书名	作者	索书号	价格	出版社	出版日期
1	Expert systems	R.Ullman	H3067	17.00	ustc	1994
2	Expert	R.Ullman	H3067	17.00	ustc	1994
3	Expert	R	H3067	17.00	ustc	1994
4	Expert	R	H3067	18.00	tc	1994
5	Exrt	R	H3067	18.00	tc	1994

右边有滑条可以进行滑动方便查看信息。

而若要选择其他三种查询方式，则要输入关键词，在输入框中有”请输入关键词”的提醒，能提高用户体验。若没有输入关键词，则会清空显示栏并将弹出弹窗提醒。



书名查询：

选择书名查询，在关键词栏输入“Expert”，点击查询：

图书管理

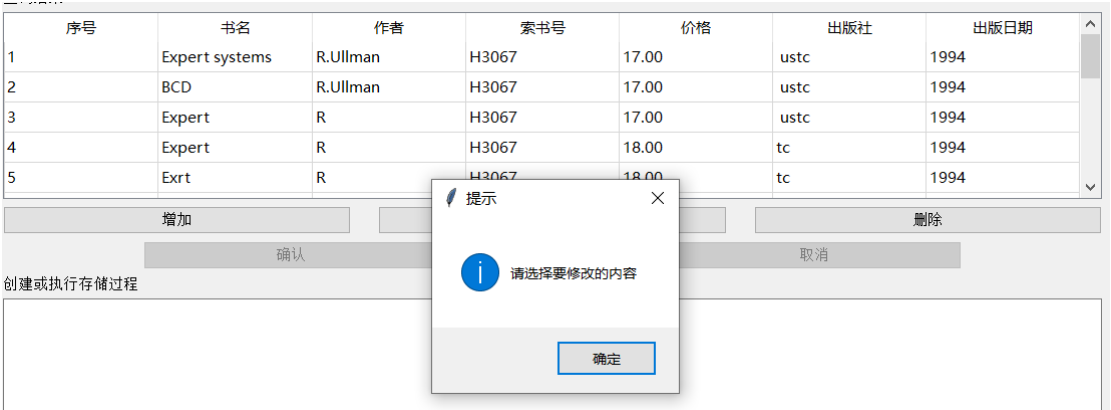
书名查询 Expert 查询

查询结果

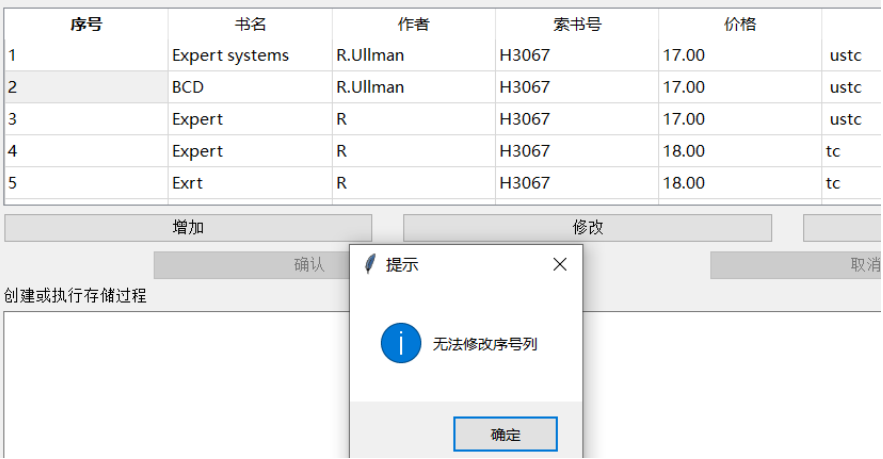
序号	书名	作者	索书号	价格	出版社	出版日期
1	Expert systems	R.Ullman	H3067	17.00	ustc	1994
2	Expert	R.Ullman	H3067	17.00	ustc	1994
3	Expert	R	H3067	17.00	ustc	1994
4	Expert	R	H3067	18.00	tc	1994

要插入，则要首先在结果显示栏中有数据才能选择插入的位置，若没有结果，则增加键无法点击：

所以一般修改前，查询要修改的信息。显示结果之后，要选择某一元组的某一属性进行修改，若不选择任何属性，则会弹窗提醒，提示用户进行选择：



而由于第一列“序号”是系统的属性，用户不能修改，所以如果用户选择该属性中的任何一个元素，均会有弹窗提醒：



选择可修改的元素进行修改，在修改过程中，增加、修改、删除键失效，防止用户误点，提高体验。在修改过程中只有确认键和取消键有效，点击确认键后，则只修改修改一开始选中的元素，而修改过程中若不小心修改了其他元素，点击确认后其他元素数据不会改变，减小了误点而导致其他数据遭到意想不到的修改。点击取消键则会取消修改，不会有任何数据遭到修改：



如一开始选中序号 1 书名为“Expert systems”的元素，点击修改，使其修改为“Database”，且将作者修改成“xy”，点击确认后，再查询所有：

查询所有

请输入关键词

查询

查询结果

序号	书名	作者	索书号	价格	出版社	出版日期
1	Database	R.Ullman	H3067	17.00	ustc	1994
2	BCD	R.Ullman	H3067	17.00	ustc	1994
3	Expert	R	H3067	17.00	ustc	1994
4	Expert	R	H3067	18.00	tc	1994
5	Exrt	R	H3067	18.00	tc	1994

增加

修改

删除

确认

取消

创建或执行存储过程

发现只有一开始选中的书名才能被修改，而作者未被修改，结果与实验设计一致。

4.7 图书信息删除功能

要删除，也要首先在结果显示栏中有数据才能选择删除的数据，若没有结果，则删除键无法点击：

查询结果

序号	书名	作者	索书号	价格	出版社	出版日期

增加

修改

删除

所以一般删除前，查询要删除的信息，待在显示栏中出现后再选择要删除的数据。显示结果之后，若不选择任何数据，则会弹窗提醒，提示用户进行选择：

序号	书名	作者	索书号	价格	出版社	出版日期
1	R	R	A3067	20.00	WTO	2020
2	ABC	XY	A456	66.66	MMT	2050
3	ABC	A	A123	20.00	MIT	2050
4	R	A	A3067	50	ustc	2020

增加

修改

删除

确认

取消

创建或执行存储过程

提示

请选择要删除的行

确定

若选择某一行，则会删除该行的数据：

序号	书名	作者	索书号	价格	出版社	出版日期
1	R	R	A3067	20.00	WTO	2020
2	ABC	XY	A456	66.66	MMT	2050
3	ABC	A	A123	20.00	MIT	2050
4	R	A	A3067	50	ustc	2020

选择第三行的数据，点击删除结果如下：

序号	书名	作者	索书号	价格	出版社	出版日期
1	R	R	A3067	20.00	WTO	2020
2	ABC	XY	A456	66.66	MMT	2050
3	R	A	A3067	50	ustc	2020

增加

修改

删除

确认

取消

而若选择某行的某个属性，则其他元组的该属性值与该元素相等的所有元组均会被删除：

序号	书名	作者	索书号	价格	出版社	出版日期
1	R	R	A3067	20.00	WTO	2020
2	ABC	XY	A456	66.66	MMT	2050
3	R	A	A3067	50	ustc	2020

如选择书名为“R”，点击删除：

查询所有

请输入关键词

查询

查询结果

序号	书名	作者	索书号	价格	出版社	出版日期
1	ABC	XY	A456	66.66	MMT	2050

发现书名等于“R”的所有元组均被删除，此功能可以依据特定属性值来进行批量删除，提高了删除效率。

4.8 存储过程的创建与执行

在区域 2 中输入符合 MySQL 语法的存储过程，然后点击区域 3 中的创建按钮即可创建新的存储过程，若没有输入而点击了创建则会弹出弹窗进行提醒：

创建或执行存储过程

提示

创建的存储过程内容不能为空

确定

执行结果

创建

执行

清除内容

输入一个简单的创建存储过程的语句：

创建或执行存储过程

```
create procedure matches()
begin
select * from xybook;
end
```

点击创建：

创建或执行存储过程

```
create procedure matches()
begin
select *from xybook;
end
```

执行结果

提示

创建成功!

确定

我们再另外创建一个错误的存储过程：

创建或执行存储过程

```
create procedure hhh()
begin
select *from ;
end
```

执行结果

提示

(1064, "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\n\nend' at line 3")

确定

发现弹出弹窗提示发生错误，提示用户语法错误，方便用户进行修改。

下面进行调用，首先点击清除内容按键，则会将创建输入框中的内容清空，再在其中输入调用的语句，并点击执行按钮：

查询结果

序号	书名	作者	索书号	价格	出版社	出版日期
1	ABC	XY	A456	66.66	MMT	2050
2	ABD	xy	A123	99.99	MIT	2025

增加

修改

删除

确认

取消

创建或执行存储过程

call matches

执行结果

((('ABC', 'XY', 'A456', '66.66', 'MMT', '2050'), ('ABD', 'xy', 'A123', '99.99', 'MIT', '2025')))

提示

执行成功!

确定

而如果调用语句，存在问题：

创建或执行存储过程

call masssssss

执行结果

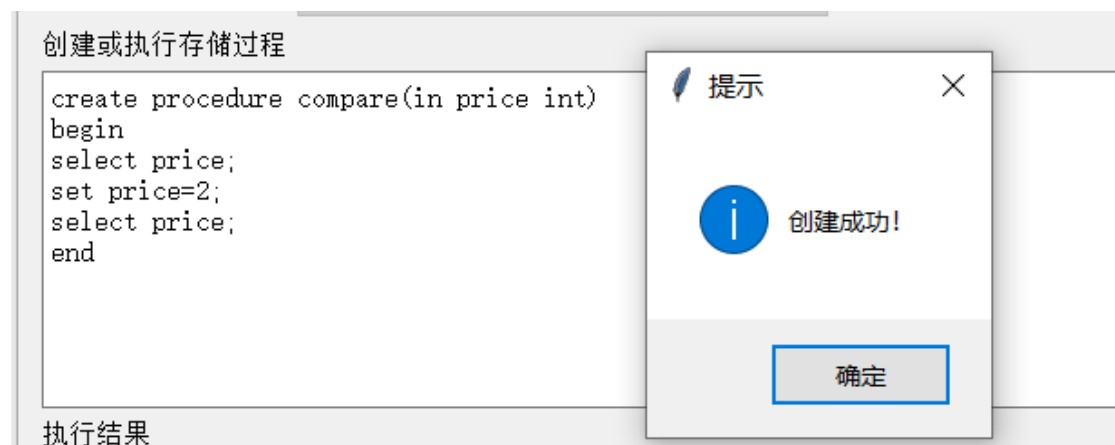
提示

(1305, 'PROCEDURE test.masssssss does not exist')

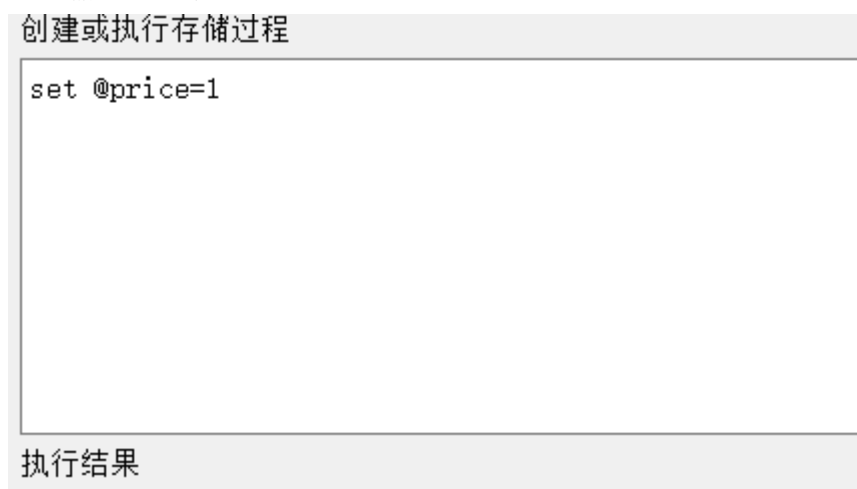
确定

则会利用弹窗进行相应的提醒。

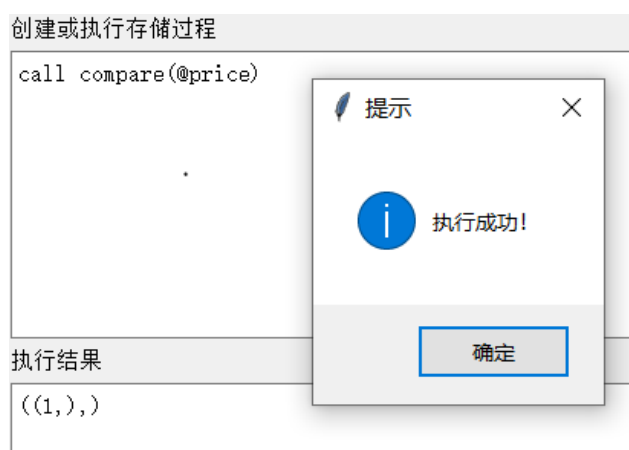
创建有参数的存储过程:



设置输入的参数:



执行:



5 总结

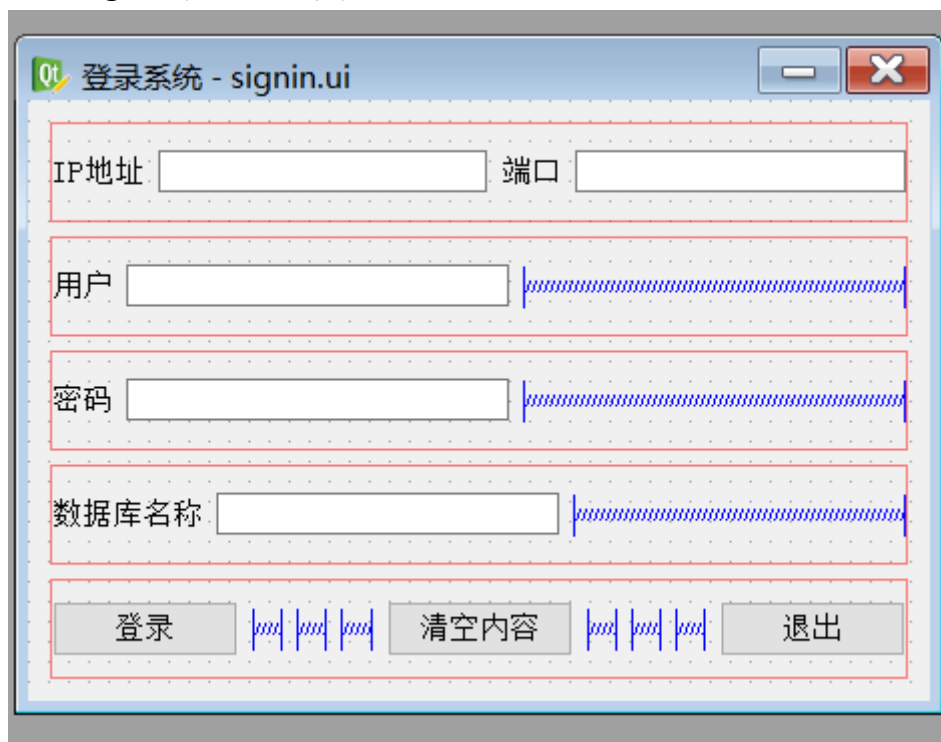
本次实验中使用 python 的 pyqt4 库和 pymysql 库对该系统进行开发，除了使我更深入了解了数据库的开发流程，也对图形界面设计有了更加深入的了解。且本次考虑的细节较多，能够在图形化界面中及时提醒用户，提高了用户体验。

备注：

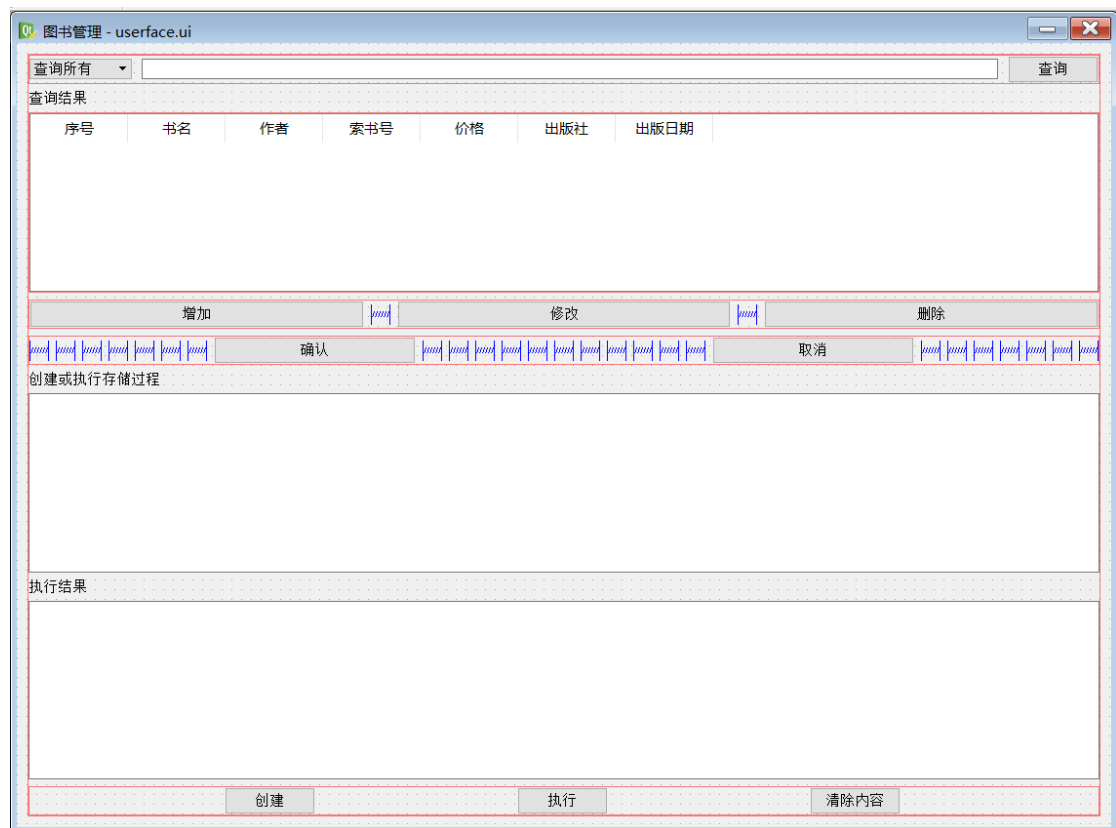
程序采用 utf-8，本地数据可能需要调整，若不调整，最好不要使用中文否则会出现乱码，建议使用英文。

附录：

1. Qt designer 设计的登录界面



2. Qt designer 设计的主界面



2. 登录界面的程序 signin.py

```
1.     # -*- coding: utf-8 -*-
2.
3.     # Form implementation generated from reading ui file 'signin.ui'
4.     #
5.     # Created by: PyQt4 UI code generator 4.11.4
6.     #
7.     # WARNING! All changes made in this file will be lost!
8.
9.     from PyQt4 import QtCore, QtGui
10.
11.     try:
12.         _fromUtf8 = QtCore.QString.fromUtf8
13.     except AttributeError:
14.         def _fromUtf8(s):
15.             return s
16.
17.     try:
18.         _encoding = QtGui.QApplication.UnicodeUTF8
19.         def _translate(context, text, disambig):
20.             return QtGui.QApplication.translate(context, text, disambig, _encoding)
```

```
21. except AttributeError:
22.     def _translate(context, text, disambig):
23.         return QtGui.QApplication.translate(context, text, disambig)
24.
25. class Ui_Form1(object):
26.     def setupUi(self, Form):
27.         Form.setObjectName(_fromUtf8("Form"))
28.         Form.resize(450, 300)
29.         self.verticalLayout = QtGui.QVBoxLayout(Form)
30.         self.verticalLayout.setObjectName(_fromUtf8("verticalLayout"))
31.         self.horizontalLayout = QtGui.QHBoxLayout()
32.         self.horizontalLayout.setObjectName(_fromUtf8("horizontalLayout"))
33.         self.label = QtGui.QLabel(Form)
34.         self.label.setObjectName(_fromUtf8("label"))
35.         self.horizontalLayout.addWidget(self.label)
36.         self.lineEdit = QtGui.QLineEdit(Form)
37.         self.lineEdit.setText(_fromUtf8(""))
38.         self.lineEdit.setObjectName(_fromUtf8("lineEdit"))
39.         self.horizontalLayout.addWidget(self.lineEdit)
40.         self.label_2 = QtGui.QLabel(Form)
41.         self.label_2.setObjectName(_fromUtf8("label_2"))
42.         self.horizontalLayout.addWidget(self.label_2)
43.         self.lineEdit_3 = QtGui.QLineEdit(Form)
44.         self.lineEdit_3.setEnabled(True)
45.         sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Preferred, QtGui.QS
            izePolicy.Fixed)
46.         sizePolicy.setHorizontalStretch(0)
47.         sizePolicy.setVerticalStretch(0)
48.         sizePolicy.setHeightForWidth(self.lineEdit_3.sizePolicy().hasHeightF
            orWidth())
49.         self.lineEdit_3.setSizePolicy(sizePolicy)
50.         self.lineEdit_3.setObjectName(_fromUtf8("lineEdit_3"))
51.         self.horizontalLayout.addWidget(self.lineEdit_3)
52.         self.verticalLayout.addLayout(self.horizontalLayout)
53.         self.horizontalLayout_2 = QtGui.QHBoxLayout()
54.         self.horizontalLayout_2.setObjectName(_fromUtf8("horizontalLayout_2"
            ))
55.         self.label_3 = QtGui.QLabel(Form)
56.         self.label_3.setObjectName(_fromUtf8("label_3"))
57.         self.horizontalLayout_2.addWidget(self.label_3)
58.         self.lineEdit_2 = QtGui.QLineEdit(Form)
59.         self.lineEdit_2.setObjectName(_fromUtf8("lineEdit_2"))
60.         self.horizontalLayout_2.addWidget(self.lineEdit_2)
```

```
61.         spacerItem = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding,
QtGui.QSizePolicy.Minimum)
62.         self.horizontalLayout_2.addItem(spacerItem)
63.         self.verticalLayout.addLayout(self.horizontalLayout_2)
64.         self.horizontalLayout_3 = QtGui.QHBoxLayout()
65.         self.horizontalLayout_3.setObjectName(_fromUtf8("horizontalLayout_3"
))
66.         self.label_4 = QtGui.QLabel(Form)
67.         self.label_4.setObjectName(_fromUtf8("label_4"))
68.         self.horizontalLayout_3.addWidget(self.label_4)
69.         self.lineEdit_4 = QtGui.QLineEdit(Form)
70.         self.lineEdit_4.setObjectName(_fromUtf8("lineEdit_4"))
71.         self.horizontalLayout_3.addWidget(self.lineEdit_4)
72.         spacerItem1 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding,
QtGui.QSizePolicy.Minimum)
73.         self.horizontalLayout_3.addItem(spacerItem1)
74.         self.verticalLayout.addLayout(self.horizontalLayout_3)
75.         self.horizontalLayout_4 = QtGui.QHBoxLayout()
76.         self.horizontalLayout_4.setObjectName(_fromUtf8("horizontalLayout_4"
))
77.         self.label_5 = QtGui.QLabel(Form)
78.         self.label_5.setObjectName(_fromUtf8("label_5"))
79.         self.horizontalLayout_4.addWidget(self.label_5)
80.         self.lineEdit_5 = QtGui.QLineEdit(Form)
81.         self.lineEdit_5.setObjectName(_fromUtf8("lineEdit_5"))
82.         self.horizontalLayout_4.addWidget(self.lineEdit_5)
83.         spacerItem2 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding,
QtGui.QSizePolicy.Minimum)
84.         self.horizontalLayout_4.addItem(spacerItem2)
85.         self.verticalLayout.addLayout(self.horizontalLayout_4)
86.         self.horizontalLayout_5 = QtGui.QHBoxLayout()
87.         self.horizontalLayout_5.setObjectName(_fromUtf8("horizontalLayout_5"
))
88.         self.pushButton = QtGui.QPushButton(Form)
89.         self.pushButton.setObjectName(_fromUtf8("pushButton"))
90.         self.horizontalLayout_5.addWidget(self.pushButton)
91.         spacerItem3 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding,
QtGui.QSizePolicy.Minimum)
92.         self.horizontalLayout_5.addItem(spacerItem3)
93.         spacerItem4 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding,
QtGui.QSizePolicy.Minimum)
94.         self.horizontalLayout_5.addItem(spacerItem4)
95.         spacerItem5 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding,
QtGui.QSizePolicy.Minimum)
```



```
96.         self.horizontalLayout_5.addItem(spacerItem5)
97.         self.pushButton_3 = QtGui.QPushButton(Form)
98.         self.pushButton_3.setObjectName(_fromUtf8("pushButton_3"))
99.         self.horizontalLayout_5.addWidget(self.pushButton_3)
100.        spacerItem6 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding
    , QtGui.QSizePolicy.Minimum)
101.        self.horizontalLayout_5.addItem(spacerItem6)
102.        spacerItem7 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding
    , QtGui.QSizePolicy.Minimum)
103.        self.horizontalLayout_5.addItem(spacerItem7)
104.        spacerItem8 = QtGui.QSpacerItem(40, 20, QtGui.QSizePolicy.Expanding
    , QtGui.QSizePolicy.Minimum)
105.        self.horizontalLayout_5.addItem(spacerItem8)
106.        self.pushButton_2 = QtGui.QPushButton(Form)
107.        self.pushButton_2.setObjectName(_fromUtf8("pushButton_2"))
108.        self.horizontalLayout_5.addWidget(self.pushButton_2)
109.        self.verticalLayout.addLayout(self.horizontalLayout_5)
110.
111.        self.retranslateUi(Form)
112.        QtCore.QMetaObject.connectSlotsByName(Form)
113.
114.    def retranslateUi(self, Form):
115.        Form.setWindowTitle(_translate("Form", "登录系统", None))
116.        self.label.setText(_translate("Form", "IP 地址", None))
117.        self.label_2.setText(_translate("Form", "端口", None))
118.        self.label_3.setText(_translate("Form", "用户", None))
119.        self.label_4.setText(_translate("Form", "密码", None))
120.        self.lineEdit_4.setEchoMode(QtGui.QLineEdit.Password)
121.        self.label_5.setText(_translate("Form", "数据库名称", None))
122.        self.pushButton.setText(_translate("Form", "登录", None))
123.        self.pushButton_3.setText(_translate("Form", "清空内容", None))
124.        self.pushButton_2.setText(_translate("Form", "退出", None))
```

3. 主界面的程序 userface.py

```
1.  # -*- coding: utf-8 -*-
2.
3.  # Form implementation generated from reading ui file 'userface.ui'
4.  #
5.  # Created by: PyQt4 UI code generator 4.11.4
6.  #
7.  # WARNING! All changes made in this file will be lost!
8.
9.  from PyQt4 import QtCore, QtGui
10.
11.  try:
12.      _fromUtf8 = QtCore.QString.fromUtf8
13.  except AttributeError:
14.      def _fromUtf8(s):
15.          return s
16.
17.  try:
18.      _encoding = QtGui.QApplication.UnicodeUTF8
19.      def _translate(context, text, disambig):
20.          return QtGui.QApplication.translate(context, text, disambig, _encoding)
21.  except AttributeError:
22.      def _translate(context, text, disambig):
23.          return QtGui.QApplication.translate(context, text, disambig)
24.
25.  class Ui_Form(object):
26.      def setupUi(self, Form):
27.          Form.setObjectName(_fromUtf8("Form"))
28.          Form.resize(1122, 804)
29.          self.verticalLayout_3 = QtGui.QVBoxLayout(Form)
30.          self.verticalLayout_3.setObjectName(_fromUtf8("verticalLayout_3"))
31.          self.verticalLayout_2 = QtGui.QVBoxLayout()
32.          self.verticalLayout_2.setObjectName(_fromUtf8("verticalLayout_2"))
33.          self.horizontalLayout = QtGui.QHBoxLayout()
34.          self.horizontalLayout.setSizeConstraint(QtGui.QLayout.SetMinimumSize)
35.          self.horizontalLayout.setSpacing(10)
36.          self.horizontalLayout.setObjectName(_fromUtf8("horizontalLayout"))
37.          self.comboBox = QtGui.QComboBox(Form)
38.          self.comboBox.setObjectName(_fromUtf8("comboBox"))
39.          self.comboBox.addItem(_fromUtf8(""))
40.          self.comboBox.addItem(_fromUtf8(""))
41.          self.comboBox.addItem(_fromUtf8(""))
```

```
42.         self.comboBox.addItem(_fromUtf8(""))
43.         self.horizontalLayout.addWidget(self.comboBox)
44.         self.lineEdit = QtGui.QLineEdit(Form)
45.         self.lineEdit.setBaseSize(QtCore.QSize(0, 2))
46.         self.lineEdit.setObjectName(_fromUtf8("lineEdit"))
47.         self.horizontalLayout.addWidget(self.lineEdit)
48.         self.pushButton = QtGui.QPushButton(Form)
49.         self.pushButton.setObjectName(_fromUtf8("pushButton"))
50.         self.horizontalLayout.addWidget(self.pushButton)
51.         self.verticalLayout_2.addLayout(self.horizontalLayout)
52.         self.label = QtGui.QLabel(Form)
53.         self.label.setObjectName(_fromUtf8("label"))
54.         self.verticalLayout_2.addWidget(self.label)
55.         self.verticalLayout = QtGui.QVBoxLayout()
56.         self.verticalLayout.setObjectName(_fromUtf8("verticalLayout"))
57.         self.tableWidget = QtGui.QTableWidget(Form)
58.         self.tableWidget.setObjectName(_fromUtf8("tableWidget"))
59.         self.tableWidget.setColumnCount(7)
60.         self.tableWidget.setRowCount(0)
61.         item = QtGui.QTableWidgetItem()
62.         self.tableWidget.setHorizontalHeaderItem(0, item)
63.         item = QtGui.QTableWidgetItem()
64.         self.tableWidget.setHorizontalHeaderItem(1, item)
65.         item = QtGui.QTableWidgetItem()
66.         self.tableWidget.setHorizontalHeaderItem(2, item)
67.         item = QtGui.QTableWidgetItem()
68.         self.tableWidget.setHorizontalHeaderItem(3, item)
69.         item = QtGui.QTableWidgetItem()
70.         self.tableWidget.setHorizontalHeaderItem(4, item)
71.         item = QtGui.QTableWidgetItem()
72.         self.tableWidget.setHorizontalHeaderItem(5, item)
73.         item = QtGui.QTableWidgetItem()
74.         self.tableWidget.setHorizontalHeaderItem(6, item)
75.         self.verticalLayout.addWidget(self.tableWidget)
76.         self.verticalLayout_2.addLayout(self.verticalLayout)
77.         self.horizontalLayout_2 = QtGui.QHBoxLayout()
78.         self.horizontalLayout_2.setObjectName(_fromUtf8("horizontalLayout_2"))
79.         self.pushButton_3 = QtGui.QPushButton(Form)
80.         self.pushButton_3.setObjectName(_fromUtf8("pushButton_3"))
81.         self.horizontalLayout_2.addWidget(self.pushButton_3)
82.         spacerItem = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtGui.QSizePolicy.Minimum)
83.         self.horizontalLayout_2.addItem(spacerItem)
```

```
84.         self.pushButton_4 = QtGui.QPushButton(Form)
85.         self.pushButton_4.setObjectName(_fromUtf8("pushButton_4"))
86.         self.horizontalLayout_2.addWidget(self.pushButton_4)
87.         spacerItem1 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtG
            ui.QSizePolicy.Minimum)
88.         self.horizontalLayout_2.addItem(spacerItem1)
89.         self.pushButton_2 = QtGui.QPushButton(Form)
90.         self.pushButton_2.setObjectName(_fromUtf8("pushButton_2"))
91.         self.horizontalLayout_2.addWidget(self.pushButton_2)
92.         self.verticalLayout_2.addLayout(self.horizontalLayout_2)
93.         self.horizontalLayout_4 = QtGui.QHBoxLayout()
94.         self.horizontalLayout_4.setObjectName(_fromUtf8("horizontalLayout_4"
            ))
95.         spacerItem2 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtG
            ui.QSizePolicy.Minimum)
96.         self.horizontalLayout_4.addItem(spacerItem2)
97.         spacerItem3 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtG
            ui.QSizePolicy.Minimum)
98.         self.horizontalLayout_4.addItem(spacerItem3)
99.         spacerItem4 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, QtG
            ui.QSizePolicy.Minimum)
100.        self.horizontalLayout_4.addItem(spacerItem4)
101.        spacerItem5 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Qt
            Gui.QSizePolicy.Minimum)
102.        self.horizontalLayout_4.addItem(spacerItem5)
103.        spacerItem6 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Qt
            Gui.QSizePolicy.Minimum)
104.        self.horizontalLayout_4.addItem(spacerItem6)
105.        spacerItem7 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Qt
            Gui.QSizePolicy.Minimum)
106.        self.horizontalLayout_4.addItem(spacerItem7)
107.        spacerItem8 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Qt
            Gui.QSizePolicy.Minimum)
108.        self.horizontalLayout_4.addItem(spacerItem8)
109.        self.pushButton_7 = QtGui.QPushButton(Form)
110.        self.pushButton_7.setObjectName(_fromUtf8("pushButton_7"))
111.        self.horizontalLayout_4.addWidget(self.pushButton_7)
112.        spacerItem9 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Qt
            Gui.QSizePolicy.Minimum)
113.        self.horizontalLayout_4.addItem(spacerItem9)
114.        spacerItem10 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            tGui.QSizePolicy.Minimum)
115.        self.horizontalLayout_4.addItem(spacerItem10)
```

```
116.         spacerItem11 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
117.         self.horizontalLayout_4.addItem(spacerItem11)
118.         spacerItem12 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
119.         self.horizontalLayout_4.addItem(spacerItem12)
120.         spacerItem13 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
121.         self.horizontalLayout_4.addItem(spacerItem13)
122.         spacerItem14 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
123.         self.horizontalLayout_4.addItem(spacerItem14)
124.         spacerItem15 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
125.         self.horizontalLayout_4.addItem(spacerItem15)
126.         spacerItem16 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
127.         self.horizontalLayout_4.addItem(spacerItem16)
128.         spacerItem17 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
129.         self.horizontalLayout_4.addItem(spacerItem17)
130.         spacerItem18 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
131.         self.horizontalLayout_4.addItem(spacerItem18)
132.         spacerItem19 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
133.         self.horizontalLayout_4.addItem(spacerItem19)
134.         self.pushButton_8 = QtGui.QPushButton(Form)
135.         self.pushButton_8.setObjectName(_fromUtf8("pushButton_8"))
136.         self.horizontalLayout_4.addWidget(self.pushButton_8)
137.         spacerItem20 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
138.         self.horizontalLayout_4.addItem(spacerItem20)
139.         spacerItem21 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
140.         self.horizontalLayout_4.addItem(spacerItem21)
141.         spacerItem22 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
142.         self.horizontalLayout_4.addItem(spacerItem22)
143.         spacerItem23 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
144.         self.horizontalLayout_4.addItem(spacerItem23)
145.         spacerItem24 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
            QtGui.QSizePolicy.Minimum)
```

```
146.         self.horizontalLayout_4.addItem(spacerItem24)
147.         spacerItem25 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
        QtGui.QSizePolicy.Minimum)
148.         self.horizontalLayout_4.addItem(spacerItem25)
149.         spacerItem26 = QtGui.QSpacerItem(20, 20, QtGui.QSizePolicy.Fixed, Q
        QtGui.QSizePolicy.Minimum)
150.         self.horizontalLayout_4.addItem(spacerItem26)
151.         self.verticalLayout_2.addLayout(self.horizontalLayout_4)
152.         self.label_2 = QtGui.QLabel(Form)
153.         self.label_2.setObjectName(_fromUtf8("label_2"))
154.         self.verticalLayout_2.addWidget(self.label_2)
155.         self.textEdit_2 = QtGui.QTextEdit(Form)
156.         self.textEdit_2.setObjectName(_fromUtf8("textEdit_2"))
157.         self.verticalLayout_2.addWidget(self.textEdit_2)
158.         self.label_3 = QtGui.QLabel(Form)
159.         self.label_3.setObjectName(_fromUtf8("label_3"))
160.         self.verticalLayout_2.addWidget(self.label_3)
161.         self.textEdit = QtGui.QTextEdit(Form)
162.         self.textEdit.setObjectName(_fromUtf8("textEdit"))
163.         self.verticalLayout_2.addWidget(self.textEdit)
164.         self.horizontalLayout_3 = QtGui.QHBoxLayout()
165.         self.horizontalLayout_3.setObjectName(_fromUtf8("horizontalLayout_3
        "))
166.         self.pushButton_5 = QtGui.QPushButton(Form)
167.         sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSize
        Policy.Fixed)
168.         sizePolicy.setHorizontalStretch(0)
169.         sizePolicy.setVerticalStretch(0)
170.         sizePolicy.setHeightForWidth(self.pushButton_5.sizePolicy().hasHeig
        htForWidth())
171.         self.pushButton_5.setSizePolicy(sizePolicy)
172.         self.pushButton_5.setObjectName(_fromUtf8("pushButton_5"))
173.         self.horizontalLayout_3.addWidget(self.pushButton_5)
174.         self.pushButton_9 = QtGui.QPushButton(Form)
175.         sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSize
        Policy.Fixed)
176.         sizePolicy.setHorizontalStretch(0)
177.         sizePolicy.setVerticalStretch(0)
178.         sizePolicy.setHeightForWidth(self.pushButton_9.sizePolicy().hasHeig
        htForWidth())
179.         self.pushButton_9.setSizePolicy(sizePolicy)
180.         self.pushButton_9.setObjectName(_fromUtf8("pushButton_9"))
181.         self.horizontalLayout_3.addWidget(self.pushButton_9)
182.         self.pushButton_6 = QtGui.QPushButton(Form)
```

```
183.         sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed, QtGui.QSize
        Policy.Fixed)
184.         sizePolicy.setHorizontalStretch(0)
185.         sizePolicy.setVerticalStretch(0)
186.         sizePolicy.setHeightForWidth(self.pushButton_6.sizePolicy().hasHeig
        htForWidth())
187.         self.pushButton_6.setSizePolicy(sizePolicy)
188.         self.pushButton_6.setObjectName(_fromUtf8("pushButton_6"))
189.         self.horizontalLayout_3.addWidget(self.pushButton_6)
190.         self.verticalLayout_2.addLayout(self.horizontalLayout_3)
191.         self.verticalLayout_3.addLayout(self.verticalLayout_2)
192.
193.         self.retranslateUi(Form)
194.         QtCore.QMetaObject.connectSlotsByName(Form)
195.
196.     def retranslateUi(self, Form):
197.         Form.setWindowTitle(_translate("Form", "图书管理", None))
198.         self.comboBox.setItemText(0, _translate("Form", "查询所有", None))
199.         self.comboBox.setItemText(1, _translate("Form", "任意词查询
        ", None))
200.         self.comboBox.setItemText(2, _translate("Form", "作者查询", None))
201.         self.comboBox.setItemText(3, _translate("Form", "书名查询", None))
202.         self.pushButton.setText(_translate("Form", "查询", None))
203.         self.label.setText(_translate("Form", "查询结果", None))
204.         item = self.tableWidget.horizontalHeaderItem(0)
205.         item.setText(_translate("Form", "序号", None))
206.         item = self.tableWidget.horizontalHeaderItem(1)
207.         item.setText(_translate("Form", "书名", None))
208.         item = self.tableWidget.horizontalHeaderItem(2)
209.         item.setText(_translate("Form", "作者", None))
210.         item = self.tableWidget.horizontalHeaderItem(3)
211.         item.setText(_translate("Form", "索书号", None))
212.         item = self.tableWidget.horizontalHeaderItem(4)
213.         item.setText(_translate("Form", "价格", None))
214.         item = self.tableWidget.horizontalHeaderItem(5)
215.         item.setText(_translate("Form", "出版社", None))
216.         item = self.tableWidget.horizontalHeaderItem(6)
217.         item.setText(_translate("Form", "出版日期", None))
218.         self.pushButton_3.setText(_translate("Form", "增加", None))
219.         self.pushButton_4.setText(_translate("Form", "修改", None))
220.         self.pushButton_2.setText(_translate("Form", "删除", None))
221.         self.pushButton_7.setText(_translate("Form", "确认", None))
222.         self.pushButton_8.setText(_translate("Form", "取消", None))
```

```

223.         self.label_2.setText(_translate("Form", "创建或执行存储过程
        ", None))
224.         self.label_3.setText(_translate("Form", "执行结果", None))
225.         self.pushButton_5.setText(_translate("Form", "创建", None))
226.         self.pushButton_9.setText(_translate("Form", "执行", None))
227.         self.pushButton_6.setText(_translate("Form", "清除内容", None))

```

4. 主程序 surface.py

```

1.  import sys
2.  from userface import Ui_Form
3.  from signin import Ui_Form1
4.  from PyQt4.QtGui import *
5.  from PyQt4.QtCore import *
6.
7.  import tkinter.messagebox
8.  from tkinter import *
9.
10. import pymysql
11.
12. # class MainWindow(QWidget):
13. #     def __init__(self, parent=None):
14. #         QWidget.__init__(self, parent)
15. #         self.ui = Ui_Form()
16. #         self.ui.setupUi(self)
17. #
18. def link_db(h,p,u,w,d):
19.     # conn = pymysql.connect(host='202.38.88.99',
20.     #                         port=1434,
21.     #                         user='student',
22.     #                         password='student',
23.     #                         db="student",
24.     #                         charset='utf8')
25.     # conn = pymysql.connect(host='localhost',
26.     #                         port=3306,
27.     #                         user='root',
28.     #                         password='security',
29.     #                         db="test",
30.     #                         charset='utf8')
31.     conn = pymysql.connect(host=h,
32.                             port=p,
33.                             user=u,
34.                             password=w,
35.                             db=d,
36.                             charset='utf8')

```



```
37.
38.     return conn
39.
40. #默认连接
41. conn = link_db('localhost',3306,'root','security','test')
42. #conn = link_db('localhost',3306,'root','security','test')
43. cur = conn.cursor()
44.     #处理更新
45.
46. #初始化
47. host=''
48. db=''
49. port=0
50. user=''
51. password=''
52. class MainWindow(QWidget,Ui_Form):
53.     #定义当前行,列
54.     currentrow=0
55.     currentcolumn=0
56.     #mode 用来确认确定键时时修改还是增加 1 为增加, -1 为修改 11
57.     mode=0
58.     #为修改记录数据
59.     title=''
60.     author=''
61.     title_id=''
62.     price=''
63.     press=''
64.     publish_time=''
65.     def __init__(self, parent=None):
66.         super(MainWindow, self).__init__()
67.         self.setupUi(self)
68.         #将一部分现在不用的按键无效
69.         self.pushButton_2.setEnabled(False)
70.         self.pushButton_4.setEnabled(False)
71.         # 隐藏列表头
72.         self.tableWidget.verticalHeader().setVisible(False)
73.         #增加可以为真
74.         self.pushButton_3.setEnabled(True)
75.         #创建存储过程也是可以的
76.         self.pushButton_5.setEnabled(True)
77.         #清除内容也是可以的
78.         self.pushButton_6.setEnabled(True)
79.         self.pushButton_7.setEnabled(False)
80.         self.pushButton_8.setEnabled(False)
```

```

81.         #执行存储过程也是可以的
82.         self.pushButton_9.setEnabled(True)
83.
84.         #使表格无法编辑
85.         self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)
86.
87.         self.lineEdit.setPlaceholderText('请输入关键词')
88.         self.pushButton.clicked.connect(self.pushB_check_Clicked)
89.         self.pushButton_2.clicked.connect(self.pushB_delete_Clicked)
90.         self.pushButton_3.clicked.connect(self.pushB_add_Clicked)
91.         self.pushButton_4.clicked.connect(self.pushB_change_Clicked)
92.         self.pushButton_7.clicked.connect(self.pushB_assure_Clicked)
93.         self.pushButton_8.clicked.connect(self.pushB_cancel_Clicked)
94.         self.pushButton_5.clicked.connect(self.pushB_create_Clicked)
95.         self.pushButton_9.clicked.connect(self.pushB_excute_Clicked)
96.         self.pushButton_6.clicked.connect(self.pushB_clear_Clicked)
97.
98.
99.         #查询定义函数
100.        def pushB_check_Clicked(self):
101.            #读取选择的模式和关键词构造 sql 语句
102.            method = self.comboBox.currentText()
103.            if method=='查询所有':
104.                sql = "SELECT * FROM xybook;"
105.            else:
106.                info = self.lineEdit.text() # 读取输入的信息不能为空
107.                if info=='':
108.                    #清除表中已有数据
109.                    self.tableWidget.clearContents()
110.                    root1 = Tk()
111.                    root1.withdraw() # 隐藏主窗口
112.                    tkinter.messagebox.showinfo(title='提示', message='请输入关键词')
113.                    return 0
114.                if method=='任意词查询':
115.                    sql = "SELECT * FROM xybook WHERE author like '%%s%%' or title like '%%s%%' or author like '%%s%%' or price like '%%s%%' or press like '%%s%%' or title_id like '%%s%%' or publish_time like '%%s%%'" % (
116.                        info, info, info, info, info, info, info)
117.
118.                elif method=='作者查询':
119.                    sql = "SELECT * FROM xybook WHERE author like '%%s%%'" % info
120.                elif method=="书名查询":

```

```

121.             sql = "SELECT * FROM xybook WHERE title like '%%s%%'" % in
           fo
122.
123.         #清除掉文本
124.         #self.lineEdit.clear()
125.         # 执行 sql 语句
126.         cur.execute(sql)
127.         data = cur.fetchall()
128.         rows=len(data)
129.
130.         #自适应调整列宽度
131.         self.tableWidget.horizontalHeader().setStretchLastSection(True)
132.         self.tableWidget.horizontalHeader().setResizeMode(QHeaderView.Stret
           ch)
133.         #行不为 0，则不是数据不为空
134.         if rows!=0:
135.             #隐藏表头的数字
136.             self.tableWidget.verticalHeader().setVisible(False)
137.             columns=len(data[0])
138.             self.tableWidget.setRowCount(rows)
139.             self.tableWidget.setColumnCount(columns+1)
140.             #使增删改按键有效
141.             self.pushButton_2.setEnabled(True)
142.             self.pushButton_4.setEnabled(True)
143.             self.pushButton_3.setEnabled(True)
144.             #将数据打印到相应的行列上
145.             for i in range(0,rows):
146.                 item = QTableWidgetItem(str(i+1))
147.                 self.tableWidget.setItem(i,0, item)
148.                 for j in range(0,columns):
149.                     item=QTableWidgetItem(str(data[i][j]))
150.                     self.tableWidget.setItem(i,j+1,item)
151.             #未读取到任何数据
152.             else:
153.                 # #清除表中已有数据
154.                 self.tableWidget.clearContents()
155.                 #使增删改无效
156.                 self.pushButton_2.setEnabled(False)
157.                 self.pushButton_4.setEnabled(False)
158.                 self.pushButton_3.setEnabled(False)
159.                 #将表格设置为 0 行 0 列
160.                 # self.tableWidget.setRowCount(0)
161.                 # self.tableWidget.setColumnCount(0)
162.                 root = Tk()

```

```

163.         root.withdraw() # 隐藏主窗口
164.         tkinter.messagebox.showinfo(title='提示', message='未查找到任何相
            关数据')
165.
166.     #定义删除后的调序函数,输入参数是要删除的行
167.     def fixorder(self,row):
168.         totalrow=self.tableWidget.rowCount()
169.         for i in range(row,totalrow):
170.             item = QTableWidgetItem(str(i))
171.             self.tableWidget.setItem(i, 0, item)
172.         print('okfix')
173.
174.     #定义删除
175.     def pushB_delete_Clicked(self):
176.         #获得本行本列
177.         self.currentrow = self.tableWidget.currentRow()
178.         print(self.currentrow)
179.         #若没有选中任何行提示
180.         if (self.currentrow==-1):
181.             root2 = Tk()
182.             root2.withdraw()
183.             tkinter.messagebox.showinfo(title='提示', message='请选择要删除的
                行')
184.             return 0
185.         currentcolumn=self.tableWidget.currentColumn()
186.         # self.tableWidget.removeRow(currentrow)
187.         # print(currentrow) # 获取当前选中的列
188.         # print(currentcolumn) # 获取当前选中的行
189.
190.
191.         #若是选中某行的第0列即选中的是序号 则直接删除本行的数据即可
192.         if currentcolumn==0:
193.             sql="delete from xybook where title like '%s' and author like
                '%s' and title_id like '%s' and price like '%s' and press like '%s' and p
                ublish_time like '%s'" % (
194.                 self.tableWidget.item(self.currentrow,1).text(), self.tableWidg
                et.item(self.currentrow,2).text(), self.tableWidget.item(self.currentrow,3).
                text(),
195.                 self.tableWidget.item(self.currentrow,4).text(), self.tableWidg
                et.item(self.currentrow,5).text(), self.tableWidget.item(self.currentrow,6).
                text())
196.             cur.execute(sql)
197.             self.fixorder(self.currentrow)
198.             self.tableWidget.removeRow(self.currentrow)

```

```

199.
200.     #选中的是第一列
201.     elif currentcolumn==1:
202.         #删除数据库中相同的数据
203.         content=self.tableWidget.item(self.currentrow,1).text()#要删除的
        数据
204.         sql = "delete from xybook where title like '%s'"%(content)
205.         cur.execute(sql)
206.         #删除那些与之相同在表中的数据
207.         totalrow=self.tableWidget.rowCount()
208.         j=0
209.         for i in range(0,totalrow):
210.             if (self.tableWidget.item(j,1).text())==content:
211.                 self.fixorder(j)
212.                 self.tableWidget.removeRow(j)
213.             else:
214.                 j=j+1
215.         #选中的是第二列
216.         elif currentcolumn == 2:
217.             # 删除数据库中相同的数据
218.             content = self.tableWidget.item(self.currentrow, 2).text() #
        要删除的数据
219.             sql = "delete from xybook where author like '%s'" % (content)
220.             cur.execute(sql)
221.             # 删除那些与之相同在表中的数据
222.             totalrow = self.tableWidget.rowCount()
223.             j = 0
224.             for i in range(0, totalrow):
225.                 if (self.tableWidget.item(j, 2).text()) == content:
226.                     self.fixorder(j)
227.                     self.tableWidget.removeRow(j)
228.                 else:
229.                     j = j + 1
230.         #第三列的话
231.         elif currentcolumn== 3:
232.             # 删除数据库中相同的数据
233.             content = self.tableWidget.item(self.currentrow, 3).text() #
        要删除的数据
234.             sql = "delete from xybook where title_id like '%s'" % (content
        )
235.             cur.execute(sql)
236.             # 删除那些与之相同在表中的数据
237.             totalrow = self.tableWidget.rowCount()

```

```

238.         #用 j 来控制删除的位置
239.         j = 0
240.         for i in range(0, totalrow):
241.             if (self.tableWidget.item(j, 3).text()) == content:
242.                 self.fixorder(j)
243.                 print(j)
244.                 self.tableWidget.removeRow(j)
245.             else:
246.                 j = j + 1
247.
248.
249.         # 第四列的话
250.         elif currentcolumn == 4:
251.             # 删除数据库中相同的数据
252.             content = self.tableWidget.item(self.currentrow, 4).text() #
要删除的数据
253.             print(content)
254.             sql = "delete from xybook where price like '%s'" % (content)
255.             cur.execute(sql)
256.             # 删除那些与之相同在表中的数据
257.             totalrow = self.tableWidget.rowCount()
258.             j = 0
259.             for i in range(0, totalrow):
260.                 if (self.tableWidget.item(j, 4).text()) == content:
261.                     self.fixorder(j)
262.                     self.tableWidget.removeRow(j)
263.                 else:
264.                     j = j + 1
265.
266.         # 第五列的话
267.         elif currentcolumn == 5:
268.             # 删除数据库中相同的数据
269.             content = self.tableWidget.item(self.currentrow, 5).text() #
要删除的数据
270.             sql = "delete from xybook where press like '%s'" % (content)
271.             cur.execute(sql)
272.             # 删除那些与之相同在表中的数据
273.             totalrow = self.tableWidget.rowCount()
274.             j = 0
275.             for i in range(0, totalrow):
276.                 if (self.tableWidget.item(j, 5).text()) == content:
277.                     self.fixorder(j)
278.                     self.tableWidget.removeRow(j)
279.                 else:
280.                     j = j + 1

```

```

280.         # 第六列的话
281.         elif currentcolumn== 6:
282.             # 删除数据库中相同的数据
283.             content = self.tableWidget.item(self.currentrow, 6).text() #
                要删除的数据
284.             sql = "delete from xybook where publish_time like '%s'" % (con
                tent)
285.             cur.execute(sql)
286.             # 删除那些与之相同在表中的数据
287.             totalrow = self.tableWidget.rowCount()
288.             j = 0
289.             for i in range(0, totalrow):
290.                 if (self.tableWidget.item(j,6).text()) == content:
291.                     self.fixorder(j)
292.                     self.tableWidget.removeRow(j)
293.                 else:
294.                     j = j + 1
295.             #当删除到没有的时候使增删改无效
296.             if self.tableWidget.rowCount()==0:
297.                 # 将一部分现在不用的按键无效
298.                 self.pushButton_2.setEnabled(False)
299.                 self.pushButton_4.setEnabled(False)
300.                 self.pushButton_3.setEnabled(False)
301.
302.             #调整插入后的序号
303.             def add_fix_order(self,row):
304.                 totalrow = self.tableWidget.rowCount()
305.                 for i in range(row, totalrow):
306.                     item = QTableWidgetItem(str(i+1))
307.                     self.tableWidget.setItem(i, 0, item)
308.
309.             #定义增加函数
310.             def pushB_add_Clicked(self):
311.                 self.pushButton_2.setEnabled(False)
312.                 self.pushButton_4.setEnabled(False)
313.                 self.pushButton_3.setEnabled(False)
314.                 self.pushButton_5.setEnabled(False)
315.                 self.pushButton_6.setEnabled(False)
316.                 self.pushButton_7.setEnabled(True)
317.                 self.pushButton_8.setEnabled(True)
318.                 self.mode=1
319.                 self.currentrow=self.tableWidget.currentRow()#获取当前选中的行
320.                 #若没有选中时,默认为 0
321.                 if (self.currentrow==-1):

```

```
322.         self.currentrow=0
323.         #插入一行
324.         self.tableWidget.insertRow(self.currentrow)
325.         #全部初始化为 0
326.         # item = QTableWidgetItem(str(''))
327.         # for i in range(1, 7):
328.         #     self.tableWidget.setItem(self.currentrow, i, item)
329.         #调整序号
330.         self.add_fix_order(self.currentrow)
331.         #设置双击时触发编辑
332.         self.tableWidget.setEditTriggers(QAbstractItemView.DoubleClicked)
333.
334.         # self.tableWidget.setItemDelegateForColumn(2, EmptyDelegate(self))
335.
336.         #要设置其他行不可编辑只有该行可以编辑
337.         # item1 = QTableWidgetItem()
338.         # item1.setFlags(Qt.ItemIsSelectable|Qt.ItemIsEnabled)
339.         # rows=self.tableWidget.rowCount()
340.         # for i in range(0,rows):
341.         #     if i!=self.currentrow:
342.         #         for j in range(0,7):
343.         #             self.tableWidget.setItem(i, j,item1)
344.         #     else:
345.         #         # QTableWidgetItem(self.tableWidget.item(i, 0)).setFlags(Q
346.         #             t.ItemIsSelectable | Qt.ItemIsEnabled)
347.         #         self.tableWidget.setItem(i,0, item1)#设置序号不可编辑
348.
349.         # item2 = QTableWidgetItem()
350.         # item2.setFlags(Qt.ItemFlag(63))
351.         # print(self.currentrow)
352.         # item1= QTableWidgetItem()
353.         # item1.setFlags(Qt.ItemFlag(63))
354.         # print('ok')
355.         # self.tableWidget.setItem(self.currentrow, 1, item1)
356.         #定义修改函数
357.         def pushB_change_Clicked(self):
358.             self.pushButton_2.setEnabled(False)
359.             self.pushButton_4.setEnabled(False)
360.             self.pushButton_3.setEnabled(False)
361.             self.pushButton_5.setEnabled(False)
362.             self.pushButton_6.setEnabled(False)
363.             self.pushButton_7.setEnabled(True)
364.             self.pushButton_8.setEnabled(True)
```



```

364.         self.mode = -1
365.         self.currentrow= self.tableWidget.currentRow()  # 获取当前选中的行
366.         if self.currentrow==-1:
367.             #把按钮重置
368.             self.pushButton_2.setEnabled(True)
369.             self.pushButton_4.setEnabled(True)
370.             self.pushButton_3.setEnabled(True)
371.             self.pushButton_5.setEnabled(False)
372.             self.pushButton_6.setEnabled(False)
373.             self.pushButton_7.setEnabled(False)
374.             self.pushButton_8.setEnabled(False)
375.             root6= Tk()
376.             root6.withdraw()  # 隐藏主窗口
377.             tkinter.messagebox.showinfo(title='提示', message='请选择要修改的
            内容')
378.             return 0
379.             #获取当前列
380.             self.currentcolumn=self.tableWidget.currentColumn()
381.             #若为序号则无法修改
382.             if self.currentcolumn==0:
383.                 # 把按钮重置
384.                 self.pushButton_2.setEnabled(True)
385.                 self.pushButton_4.setEnabled(True)
386.                 self.pushButton_3.setEnabled(True)
387.                 self.pushButton_5.setEnabled(False)
388.                 self.pushButton_6.setEnabled(False)
389.                 self.pushButton_7.setEnabled(False)
390.                 self.pushButton_8.setEnabled(False)
391.                 root8 = Tk()
392.                 root8.withdraw()  # 隐藏主窗口
393.                 tkinter.messagebox.showinfo(title='提示', message='无法修改序号列
            ')
394.                 return 0
395.
396.             # 获取原来数据
397.             self.title = self.tableWidget.item(self.currentrow, 1).text()
398.             self.author = self.tableWidget.item(self.currentrow, 2).text()
399.             self.title_id = self.tableWidget.item(self.currentrow, 3).text()
400.             self.price = self.tableWidget.item(self.currentrow, 4).text()
401.             self.press = self.tableWidget.item(self.currentrow, 5).text()
402.             self.publish_time = self.tableWidget.item(self.currentrow, 6).text(
            )
403.             # 设置双击时触发编辑
404.             self.tableWidget.setEditTriggers(QAbstractItemView.DoubleClicked)

```

```

405.         #则只让一个单元格可修改
406.
407.         # self.tableWidget.setItemDelegateForColumn(2, EmptyDelegate(self))
408.         # 要设置其他行不可编辑只有该行可以编辑
409.         # item1 = QTableWidgetItem()
410.         # item1.setFlags(Qt.ItemIsSelectable|Qt.ItemIsEnabled)
411.         # rows=self.tableWidget.rowCount()
412.         #
413.         # for i in range(0,rows):
414.         #     if i!=currentrow:
415.         #         for j in range(0,7):
416.         #             self.tableWidget.setItem(i, j,item1)
417.         # else:
418.         # QTableWidgetItem(self.tableWidget.item(i, 0)).setFlags(Qt.ItemIsSelectable | Qt.ItemIsEnabled)
419.         # self.tableWidget.setItem(i,0, item1)#设置序号不可编辑
420.         def pushB_assure_Clicked(self):
421.
422.
423.             #是增加模式
424.             if self.mode==1:
425.                 #提取数据
426.                 try:
427.                     title=self.tableWidget.item(self.currentrow,1).text()
428.                     author=self.tableWidget.item(self.currentrow,2).text()
429.                     title_id = self.tableWidget.item(self.currentrow,3).text()
430.
431.                     price = self.tableWidget.item(self.currentrow, 4).text()
432.                     press = self.tableWidget.item(self.currentrow, 5).text()
433.                     publish_time = self.tableWidget.item(self.currentrow, 6).text()
434.
435.                     # 若有数据是空的
436.                     except Exception as err:
437.                         root13 = Tk()
438.                         root13.withdraw() # 隐藏主窗口
439.                         tkinter.messagebox.showinfo(title='提示', message='请将数据补充完整')
440.
441.                         return 0
442.
443.                 # if title==' ' or author==' ' or title_id==' ' or price==' ' or press==' ' or publish_time==' ':
444.                 #     # 插入数据库中

```

```

444.         try:
445.             sql = "insert into xybook values('%s','%s','%s','%s','%s','%s','%s')"% (title,author,title_id,price,press,publish_time)
446.             cur.execute(sql)
447.
448.         except Exception as err:
449.             root4 = Tk()
450.             root4.withdraw() # 隐藏主窗口
451.             tkinter.messagebox.showinfo(title='提示', message='新增数据失败!')
452.         return 0
453.
454.         # 使表格无法编辑
455.         self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)
456.         self.pushButton_2.setEnabled(True)
457.         self.pushButton_4.setEnabled(True)
458.         self.pushButton_3.setEnabled(True)
459.         self.pushButton_5.setEnabled(True)
460.         self.pushButton_6.setEnabled(True)
461.         self.pushButton_7.setEnabled(False)
462.         self.pushButton_8.setEnabled(False)
463.         self.mode=0
464.         #是修改模式
465.         elif self.mode==-1:
466.             #提取数据
467.             try:
468.                 #获取当前修改的内容
469.                 content=self.tableWidget.item(self.currentrow,self.currentcolumn).text()
470.                 #生成 sql 语句
471.
472.                 if self.currentcolumn == 1:
473.                     sql = "update xybook set title='%s' where title like '%s' and author like '%s' and title_id like '%s' and price like '%s' and press like '%s' and publish_time like '%s'" % \
474.                         (content,self.title,self.author,self.title_id,self.price,self.press,self.publish_time)
475.                     cur.execute(sql)
476.                 elif self.currentcolumn ==2:
477.                     sql = "update xybook set author='%s' where title like '%s' and author like '%s' and title_id like '%s' and price like '%s' and press like '%s' and publish_time like '%s'" % \

```

```

478.             (content,self.title,self.author,self.title_id,self.f.price,self.press,self.publish_time)
479.             cur.execute(sql)
480.             if self.currentcolumn == 3:
481.                 sql = "update xybook set title_id='%s' where title like '%s' and author like '%s' and title_id like '%s' and price like '%s' and press like '%s' and publish_time like '%s'" % \
482.                     (content,self.title,self.author,self.title_id,self.f.price,self.press,self.publish_time)
483.                 cur.execute(sql)
484.                 if self.currentcolumn == 4:
485.                     sql = "update xybook set price='%s' where title like '%s' and author like '%s' and title_id like '%s' and price like '%s' and press like '%s' and publish_time like '%s'" % \
486.                         (content,self.title,self.author,self.title_id,self.f.price,self.press,self.publish_time)
487.                     cur.execute(sql)
488.                     if self.currentcolumn == 5:
489.                         sql = "update xybook set press='%s' where title like '%s' and author like '%s' and title_id like '%s' and price like '%s' and press like '%s' and publish_time like '%s'" % \
490.                             (content,self.title,self.author,self.title_id,self.f.price,self.press,self.publish_time)
491.                         cur.execute(sql)
492.                         if self.currentcolumn == 6:
493.                             sql = "update xybook set publish_time='%s' where title like '%s' and author like '%s' and title_id like '%s' and price like '%s' and press like '%s' and publish_time like '%s'" % \
494.                                 (content,self.title,self.author,self.title_id,self.f.price,self.press,self.publish_time)
495.                             cur.execute(sql)
496.                             # 删除那些与之相同在表中的数据
497.                             except Exception as err:
498.                                 root7 = Tk()
499.                                 root7.withdraw() # 隐藏主窗口
500.                                 tkinter.messagebox.showinfo(title='提示', message='修改数据失败!')
501.                                 return 0
502.
503. # 使表格无法编辑
504. self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)
505. self.pushButton_2.setEnabled(True)
506. self.pushButton_4.setEnabled(True)

```

```
507.         self.pushButton_3.setEnabled(True)
508.         self.pushButton_5.setEnabled(True)
509.         self.pushButton_6.setEnabled(True)
510.         self.pushButton_7.setEnabled(False)
511.         self.pushButton_8.setEnabled(False)
512.         #状态回归置 0
513.         self.mode=0
514.
515.     def pushB_cancel_Clicked(self):
516.         # 控制按钮
517.         self.pushButton_2.setEnabled(True)
518.         self.pushButton_4.setEnabled(True)
519.         self.pushButton_3.setEnabled(True)
520.         self.pushButton_5.setEnabled(True)
521.         self.pushButton_6.setEnabled(True)
522.         self.pushButton_7.setEnabled(False)
523.         self.pushButton_8.setEnabled(False)
524.         #若是增加模式
525.         if self.mode==1:
526.             self.mode=0
527.
528.         #删除该行
529.         self.fixorder(self.currentrow)
530.         self.tableWidget.removeRow(self.currentrow)
531.         # 使表格无法编辑
532.         self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)
533.     elif self.mode ==-1:
534.         self.mode = 0
535.         # 控制按钮
536.         self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)
537.         # 删除该行
538.         if self.currentcolumn==1:
539.             item = QTableWidgetItem(str(self.title))
540.         elif self.currentcolumn==2:
541.             item = QTableWidgetItem(str(self.author))
542.         elif self.currentcolumn == 3:
543.             item = QTableWidgetItem(str(self.title_id))
544.         elif self.currentcolumn == 4:
545.             item = QTableWidgetItem(str(self.price))
546.         elif self.currentcolumn == 5:
547.             item = QTableWidgetItem(str(self.press))
548.         elif self.currentcolumn == 6:
```

```

549.             item = QTableWidgetItem(str(self.publish_time))
550.             # 使表格无法编辑
551.             self.tableWidget.setItem(self.currentrow,self.currentcolumn, it
em)
552.         #创建存储过程
553.         def pushB_create_Clicked(self):
554.             #获取其中的内容
555.             sql =self.textEdit_2.toPlainText()
556.             # sql = "CREATE PROCEDURE demo_proc(IN p_in int,OUT p_out int)\
557.             # BEGIN \
558.             #     set p_out=2; \
559.             # END"
560.             sql.replace('\n', '\\')
561.             if sql=='':
562.                 root11 = Tk()
563.                 root11.withdraw() # 隐藏主窗口
564.                 tkinter.messagebox.showinfo(title='提示', message='创建的存储过程
内容不能为空')
565.                 return 0
566.
567.             # cur.execute(sql)
568.             try:
569.                 cur.execute(sql)
570.             except Exception as err:
571.                 root12 = Tk()
572.                 root12.withdraw() # 隐藏主窗口
573.                 tkinter.messagebox.showinfo(title='提示', message=err)
574.                 return 0
575.
576.                 root12 = Tk()
577.                 root12.withdraw() # 隐藏主窗口
578.                 tkinter.messagebox.showinfo(title='提示', message='创建成功! ')
579.                 return 0
580.         #执行存储过
581.         def pushB_excute_Clicked(self):
582.             # 获取其中的内容
583.             sql = self.textEdit_2.toPlainText()
584.             sql.replace('\n', '\\')
585.             if len(sql)==0:
586.                 root22 = Tk()
587.                 root22.withdraw() # 隐藏主窗口
588.                 tkinter.messagebox.showinfo(title='提示', message='执行不能为
空!')
589.                 return 0

```

```
590.
591.     try:
592.         cur.execute(sql)
593.     except Exception as er:
594.         root21 = Tk()
595.         root21.withdraw() # 隐藏主窗口
596.         tkinter.messagebox.showinfo(title='提示', message=er)
597.         return 0
598.     res = cur.fetchall()
599.     res=str(res)
600.     #         #将其打印在结果栏
601.     self.textEdit.setPlainText(res)
602.     #将其打印在结果栏
603.
604.
605.
606.     # #首先判断语法是否正确
607.     # if len(sql)<=5 or sql[0]!='c' or sql[1]!='a' or sql[2]!='l' or sq
608.     l[3]!='l' or sql[4]!=' ':
609.         #         root21 = Tk()
610.         #         root21.withdraw() # 隐藏主窗口
611.         #         tkinter.messagebox.showinfo(title='提示', message='语法出现错
612.         误!')
613.         #         return 0
614.         #
615.         # #有括号的情况
616.         # if '(' in sql:
617.         #         # 没有右括号
618.         #         if '(' not in sql:
619.         #             root23 = Tk()
620.         #             root23.withdraw() # 隐藏主窗口
621.         #             tkinter.messagebox.showinfo(title='提示', message='语法出
622.         现错误!')
623.         #             return 0
624.         #             pl=sql.find('(')#左括号位置
625.         #             #提取过程名
626.         #             proc=sql[5:pl]
627.         #             #提取参数
628.         #             pr=sql.find(')')
629.         #             parameter=sql[pl+1:pr]
630.         #             if len(parameter)==0:
631.         #                 #则不用参数
632.         #                 #直接执行即可
633.         #                 try:
```

```

631.         #             cur.callproc(proc)
632.         #             except EXCEPTION as er:
633.         #                 root24 = Tk()
634.         #                 root24.withdraw() # 隐藏主窗口
635.         #                 tkinter.messagebox.showinfo(title='提示
        #                 ', message=er)
636.         #                 return 0
637.         #                 res = cur.fetchall()
638.         #                 #将其打印在结果栏
639.         #                 self.textEdit.setPlainText(res)
640.         #             else:#有参数的情况
641.         #                 #有多个参数的情况
642.         #                 if ',' in parameter:
643.         #                     paralist=parameter.split(',')
644.         #
645.         #
646.         #
647.         #
648.         #
649.         #
650.         #
651.         #             #有多个参数的情况
652.         #             if ',' in parameter:
653.         #                 print(proc)
654.         #                 print(parameter)
655.
656.
657.
658.
659.
660.         # # 调用 p1 存储过程，传入 4 个参数
661.         # cusr.callproc('p1', args=(1, 2, 3, 4))
662.         #
663.         # # 返回获得的集合，即存储函数中的 SELECT * FROM tmp; 结果
664.         # res1 = cusor.fetchall()
665.         # print(res1)
666.         #
667.         # if sql == '':
668.         #     root11 = Tk()
669.         #     root11.withdraw() # 隐藏主窗口
670.         #     tkinter.messagebox.showinfo(title='提示', message='创建的存储
        #     过程内容不能为空')
671.         #     return 0
672.

```



```

673.         # cur.execute(sql)
674.         # try:
675.         #     cur.execute(sql)
676.         # except Exception as err:
677.         #     root12 = Tk()
678.         #     root12.withdraw() # 隐藏主窗口
679.         #     tkinter.messagebox.showinfo(title='提示', message=err)
680.         #     return 0
681.
682.         root12 = Tk()
683.         root12.withdraw() # 隐藏主窗口
684.         tkinter.messagebox.showinfo(title='提示', message='执行成功! ')
685.         return 0
686.
687. #清除存储过程
688.     def pushB_clear_Clicked(self):
689.         self.textEdit.clear()
690.         self.textEdit_2.clear()
691.
692. #定义登录界面的类
693. class SignWindow(QWidget, Ui_Form1):
694.     def __init__(self, parent=None):
695.         super(SignWindow, self).__init__()
696.         self.setupUi(self)
697.         # 使按键使能
698.         self.pushButton_2.setEnabled(True)
699.         self.pushButton.setEnabled(True)
700.         self.pushButton.clicked.connect(self.pushB_sign_Clicked)
701.         self.pushButton_2.clicked.connect(self.pushB_exit_Clicked)
702.         self.pushButton_3.clicked.connect(self.pushB_clear_Clicked)
703.
704.
705.     def pushB_sign_Clicked(self):
706.
707.         host=self.lineEdit.text() # 读取输入的信息不能为空
708.         port=self.lineEdit_3.text()
709.         user=self.lineEdit_2.text() # 读取输入的信息不能为空
710.         password=self.lineEdit_4.text() # 读取输入的信息不能为空
711.         db=self.lineEdit_5.text() # 读取输入的信息不能为空
712.         if host==' ' or port==' ' or user==' ' or password==' ' or db==' ':
713.             root31 = Tk()
714.             root31.withdraw() # 隐藏主窗口
715.             tkinter.messagebox.showinfo(title='提示', message='请将登录信息填
完整!')

```

```
716.         return 0
717.     else:
718.         try:
719.             conn = link_db(host,int(port), user, password,db)
720.             cur = conn.cursor()
721.         except Exception as er:
722.             root32 = Tk()
723.             root32.withdraw() # 隐藏主窗口
724.             tkinter.messagebox.showinfo(title='提示', message=er)
725.             return 0
726.     root33 = Tk()
727.     root33.withdraw() # 隐藏主窗口
728.     tkinter.messagebox.showinfo(title='提示', message='登录成功!')
729.     myapp.close()
730.     return 0
731. #清空
732. def pushB_clear_Clicked(self):
733.
734.     self.lineEdit.clear() #
735.     self.lineEdit_3.clear()
736.     self.lineEdit_2.clear() #
737.     self.lineEdit_4.clear() #
738.     self.lineEdit_5.clear() #
739.
740.
741. #退出
742. def pushB_exit_Clicked(self):
743.     sys.exit(0)
744.
745.
746. if __name__ == "__main__":
747.     app = QApplication(sys.argv)
748.     # myapp=SignWindow()
749.     #显示登录界面
750.     myapp = SignWindow()
751.     myapp.show()
752.     app.exec_()
753.     #显示管理系统界面
754.     myapp1 = MainWindow()
755.     myapp1.show()
756.     app.exec_()
757.
758. conn.commit()
759. cur.close()
```

```
760. conn.close()
```