Class 01

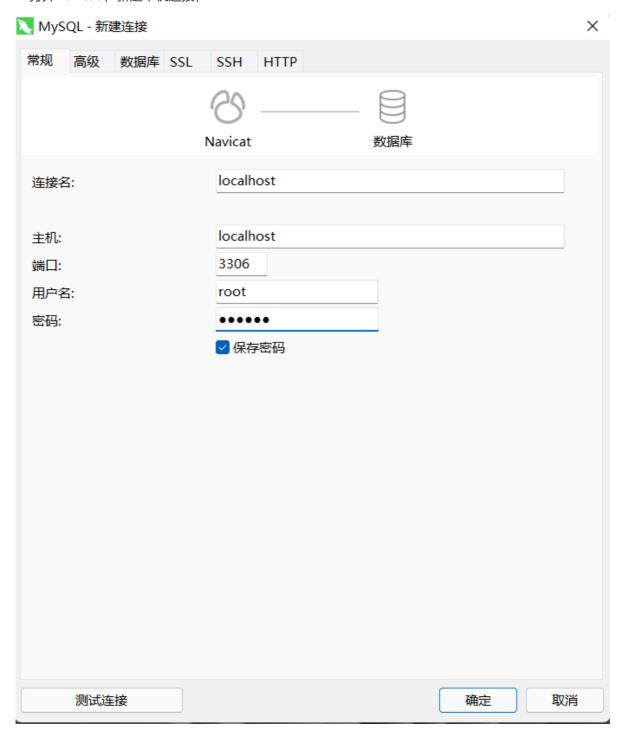
新建项目

新建项目文件夹 library , 文件夹下新建python文件 book_info.py文件

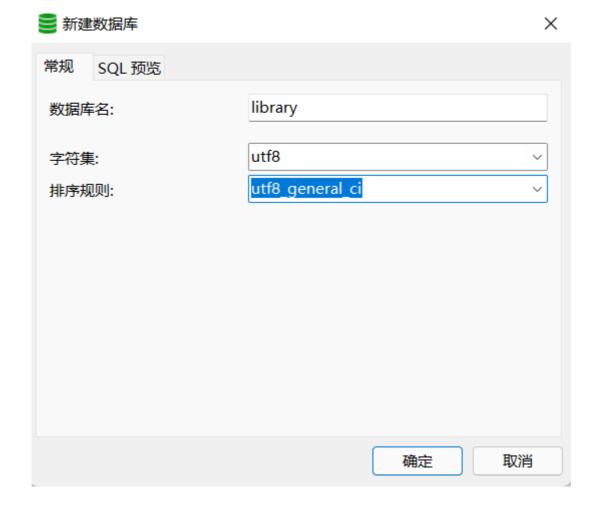
数据库

新建连接

打开Navicat,新建本机连接,



新建数据库



新建表

新建图书信息表 book_info ,列:

id,主键,不是null,自动递增

book_id, 图书号, 唯一, 可以为空

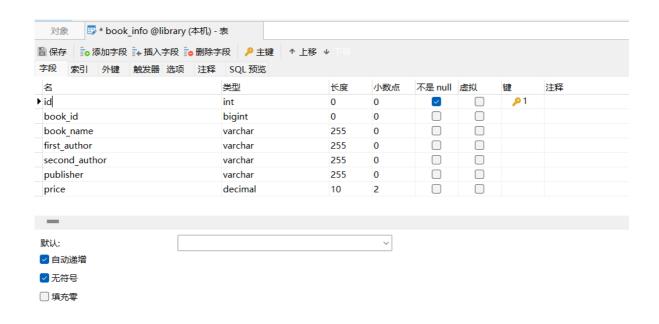
book_name, 书名,字符集: utf8_utf8_general_ci

first_author, 主编 ,字符集: utf8 utf8_general_ci

second_author, 副编 ,字符集: utf8 utf8_general_ci

publisher, 出版社,字符集: utf8_utf8_general_ci

price, 价格



连接数据库

安装pymysql

```
1 #在终端输入,安装pymysql模块
2 pip install pymysql
```

连接pymysql

```
import pymysql #引入pymysql
1
2
3
   #建立数据库连接
   db =
   pymysql.connect(host='localhost',user='root',password='123456',database='lib
   rary')
5
   cursor = db.cursor()
6
7
   #写sql语句
   sql = 'SELECT * FROM book_info'
8
9
   #提交sql语句
10
   cursor.execute(sql)
11
   #接收返回值
12
   res = cursor.fetchall()
13
                             #接受所有的返回值
14
   #res = cursor.fetchone() #接收返回值中的第一条
15
   print(res)
16
   #关闭数据库连接
17
18
   db.close()
```

```
C:\ProgramData\Anaconda3\python.exe E:/pythonproject/scrapy01/dytt/chep05.py
((1, 9787115530219, 'Python程序设计', '肖朝晖,李春忠,李海强', '张保龙,马健,刘睿', '人民邮电出版社', Decimal('59.80')),)
Process finished with exit code 0
```

图书信息模块

整体结构

```
import pymysql
1
2
3
    while True:
       print('''
4
5
           1. 查询图书信息
6
          2.新增图书信息
7
           3.修改图书信息
8
           4.删除图书信息
9
           5.搜索图书信息
10
           6.退出程序
       ''')
11
12
       option = input('请输入对应功能: ')
13
       #建立数据库连接
14
       db = pymysql.connect(host='localhost', user='root', password='123456',
    database='library')
15
       cursor = db.cursor() #建立游标
16
17
       if option=='1':
18
           #查询功能
19
           pass
20
21
       elif option=='2':
22
           #新增功能
23
           pass
24
25
       elif option=='3':
26
           #修改功能
27
           pass
28
29
       elif option=='4':
30
           #删除功能
31
           pass
32
       elif option=='5':
33
34
           #搜索功能
35
           pass
36
37
       else :
38
           print('退出程序')
39
           break
40
41
       db.close()
```

查询功能

```
mysql查询语句
SELECT * FROM 表名 #查询全表
SELECT 列1,列2,列3 FROM 表名 #查询某列
SELECT 列1,列2,列3 FROM 表名 WHERE 列3=某个值 #条件查询
```

```
sql = 'SELECT * FROM book_info'
                            #查询全表
2
  cursor.execute(sql)
3
  results = cursor.fetchall()
4
  print('\n' + '*$' * 15)
5
 print('书本ID\t书名\t主编\t副编\t出版社\t价格')
6
 for result in results:
8
     result[4], result[5], result[6]))
9 | print('*$' * 15 + '\n')
```

Class 02

图书信息模块

新增功能

mysql插入语句

```
      1
      #插入一条数据

      2
      INSERT INTO 表名(列1,列2,列3) VALUE(整数,'字符串',浮点数)

      3
      INSERT INTO 表名 SET 列1=整数,列2='字符串',列3=浮点数

      4
      #插入多条数据

      5
      INSERT INTO 表名(列1,列2,列3) VALUES(整数,'字符串',浮点数),(整数,'字符串',浮点数)
```

```
book_id = int(input('请输入书本ID: '))
2
   book_name = input('请输入书名: ')
3
   first_author = input('请输入主编: ')
   second_author = input('请输入副编: ')
   publisher = input('请输入出版社: ')
   price = float(input('请输入价格: '))
6
   print('您要新增的图书信息
   为: ',book_id,book_name,first_author,second_author,publisher,price)
9
10
   choice = input('确定录入请输入Y,放弃录入请输入N: ')
   if choice=='Y' or choice=='y':
11
        sql = "INSERT INTO
12
   book_info(book_id,book_name,first_author,second_author,publisher,price)
   VALUES ({},'{}','{}','{}','{}',
   {})".format(book_id,book_name,first_author,second_author,publisher,price)
13
        cursor.execute(sql)
14
        db.commit()
15
        print('新增成功')
```

修改功能

mysql 更新语句

```
1  #更新一个字段的值
2  UPDATE 表名 SET 列1 = '你好' WHERE 列2 = 123
3  #更新多个字段的值
5  UPDATE 表名 SET 列1='abc',列2='efg' WHERE 列3 = 123
```

```
book_id = int(input('请输入书本ID: '))
1
 2
 3
   #查询数据库,检测书号是否存在
   sql = 'SELECT * FROM book_info where book_id=%s' % book_id
4
 5
   cursor.execute(sq1)
   result = cursor.fetchone()
6
 7
8
   #如果书号存在:
   if result:
9
       #打印出准备修改的图书信息
10
11
       print('您要修改的图书为:')
12
       result[3], result[4], result[5], result[6]))
13
       print('\n')
14
       #接收新的图书信息(用于更新)
15
16
       book_name = input('请输入书名: ')
17
       first_author = input('请输入主编: ')
18
       second_author = input('请输入副编: ')
19
       publisher = input('请输入出版社: ')
20
       price = float(input('请输入价格: '))
21
       print('您要修改信息为: ', book_id, book_name, first_author, second_author,
   publisher, price)
22
23
       #根据用户选择,是否录入数据库
       choice = input('确定录入请输入Y,放弃录入请输入N: ')
24
25
       if choice == 'Y' or choice == 'y':
          sql = "UPDATE book_info SET
26
   book_name='{}',first_author='{}',second_author='{}',publisher='{}',price={}
   book_id=%d;".format(book_name,first_author,second_author,publisher,price,boo
   k_id)
27
          cursor.execute(sql)
28
          db.commit()
29
          print('修改成功')
30
31
   #如果书号不存在
32
   else:
33
34
      print('书号不存在')
```

删除功能

mysql 删除语句

```
1 #条件删除
2 DELETE FROM 表名 WHERE 列1=123456
3 #删除全表
5 DELETE FROM 表名
```

```
book_id = int(input('请输入书本ID: '))
1
2
3
   # 查询数据库, 检测书号是否存在
   sql = 'SELECT * FROM book_info where book_id=%d' % book_id
5
   cursor.execute(sq1)
6
   result = cursor.fetchone()
7
8 # 如果书号存在:
   if result:
9
10
      print('您要删除的图书为:')
11
      result[3], result[4], result[5], result[6]))
12
      choice = input('确定删除请输入Y,放弃删除请输入N: ')
13
      # 根据用户选择,是否删除
14
     if choice == 'Y' or choice == 'y':
15
16
          sql = "DELETE FROM book_info where book_id = %d" %book_id
17
          cursor.execute(sql)
         db.commit()
18
19
         print('删除成功')
20
21 # 如果书号不存在,返回
22 else:
      print('书号不存在')
23
```

Class 03

图书信息模块

搜索功能

mysql 模糊查询 LIKE语句

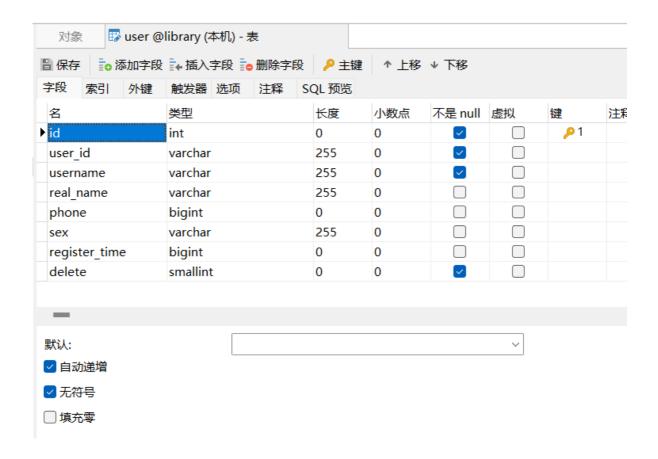
```
1 # %通配符,长度不限
2 # 匹配列1中 ,以T开头的 值
4 SELECT * FROM 表名 WHERE 列1 LIKE 'T%'
5 # 匹配列1中 , 以S结尾的 值
6 SELECT * FROM 表名 WHERE 列1 LIKE '%S'
7 # 匹配列1中 , 包含abc的值
8 SELECT * FROM 表名 WHERE 列1 LIKE '%abc%'
9 # 匹配列1或列2中 , 包含abc的值
10 SELECT * FROM 表名 WHERE 列1 LIKE '%abc%' OR 列2 LIKE '%abc%'
```

```
1 # __通配符, 长度为1
2 # 匹配列1中 , 以a开头只有两个字符的 值
4 SELECT * FROM 表名 WHERE 列1 LIKE 'a_'
5 # 匹配列1中 , 以S结尾只有两个字符的 值
6 SELECT * FROM 表名 WHERE 列1 LIKE '_S'
7 # 匹配列1中 , 以a开头b结尾,一共有4个字符的 值
8 SELECT * FROM 表名 WHERE 列1 LIKE 'a__b'
```

```
1 #模糊查询关键词
   key = input('请输入关键词:')
 3 | sql = "SELECT * FROM book_info WHERE book_name LIKE '%{}%' OR first_author
   LIKE '%{}%' OR publisher LIKE '%{}%' ".format(key,key,key)
   cursor.execute(sql)
   results = cursor.fetchall()
 5
 6
 7 # 打印出查询结果
8 print('*$' * 15)
9
   if results:
10
      print('书本ID'+'\t'+'书名'+'\t'+'主编'+'\t'+'副编'+'\t'+'出版社'+'\t'+'价
    格')
11
      for result in results:
           print('{}\t{}\t{}\t{}\t\}\t\{}\t'.format(result[1], result[2],
12
    result[3], result[4], result[5],result[6]))
13
14
15
      print('没有搜索到任何信息')
16 | print('*$' * 15)
```

用户模块

新建用户表



新建user.py

#整体结构

```
1
   import pymysql
2
3
   while True:
       print('''
4
5
           1. 查询所有用户
6
           2.新增用户
7
           3.修改用户
8
           4.删除用户
9
           5.搜索用户
10
           6.退出程序
       ''')
11
12
       option = input('请输入对应功能: ')
13
       #建立数据库连接
       db = pymysql.connect(host='localhost', user='root', password='123456',
14
   database='library')
       cursor = db.cursor() #建立游标
15
16
17
       if option=='1':
           #查询功能
18
19
           pass
20
21
       elif option=='2':
22
           #新增功能
23
           pass
24
       elif option=='3':
25
26
           #修改功能
27
           pass
```

```
28
29
        elif option=='4':
30
           #删除功能
31
           pass
32
33
       elif option=='5':
34
          #搜索功能
35
           pass
36
37
       else:
           print('退出程序')
38
39
           break
40
41
       db.close()
```

Class 04

用户模块

正则表达式

见《Python程序设计》 104-110页

字符串切片

```
1 str[起始位置:结束位置:步长]
2 #起始位置和结束位置可以为负数
```

时间戳

时间戳是指格林威治时间1970年01月01日00时00分00秒(北京时间1970年01月01日08时00分00秒) 起至现在的总秒数。

```
1 import time
2 #获取当前时间戳
3 print(time.time())
4 print(int(time.time()))
```

随机数

```
1import random2random.random()#0-1之间随机数3random.randint(3,15)#3-15之间随机整数4random.uniform(3.2,20.6)#两数之间随机浮点数5random.choice('ABCDEFG')#从序列中随机选择一个6random.sample('abcdefghijklmn', 5)#从多个字符中生成指定数量的随机字符
```

新增功能

```
1 # 正则匹配用户名
   pattern1 = r'^{a-z0-9A-z}{8,20}
   match1 = re.match(pattern1,input('请输入8-20位字母数字用户名:'))
4
   # 正则匹配手机号
   pattern2 = r'^1[3|4|5|8|9]\d{9}
   match2 = re.match(pattern2, input('请输入手机号码:'))
8
   #如果用户名和手机号匹配成功
9
   if match1 and match2:
10
       name = input('请输入名字: ')
11
       sex = input('请输入性别:')
12
       # 算法生成user_id
13
       user_id = str(time.time())[1:10] + match2.string[-3:] +
   random.choice('ABCDEFGHTWMJK')
14
       timestamp = int(time.time())
15
16
       #写入数据库
17
       sql = "INSERT INTO
   user(user_id,username,real_name,phone,sex,register_time) VALUE
    ('{}','{}','{}',{},'{}})".format(user_id,match1.string,name,
   int(match2.string),sex,timestamp)
18
       cursor.execute(sql)
19
       db.commit()
20
       print('新增成功')
21
22 #用户名和手机号匹配不成功
23
       print('请输入正确的用户名和密码')
24
```

Class 05