

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，

MySQL - 新建连接

常规 高级 数据库 SSL SSH HTTP

Navicat

数据库

连接名:

localhost

主机:

localhost

端口:

3306

用户名:

root


密码:

●●●●●●

☒ 保存密码

测试连接 确定 取消

新建数据库

 新建数据库 ×

常规

SQL 预览

数据库名:

library

字符集:

utf8

排序规则:

utf8_general_ci

确定

取消

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

对象 * book_info @library (本机) - 表

保存 添加字段 插入字段 删除字段 主键 上移 下移

字段 索引 外键 触发器 选项 注释 SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
book_id	bigint	0	0	<input type="checkbox"/>	<input type="checkbox"/>		
book_name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
first_author	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
second_author	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
publisher	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
price	decimal	10	2	<input type="checkbox"/>	<input type="checkbox"/>		

默认:

☒ 自动递增

☒ 无符号

☐ 填充零

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

```
1 import pymysql #引入pymysql
2
3 #建立数据库连接
4 db =
pymysql.connect(host='localhost',user='root',password='123456',database='library')
5 cursor = db.cursor()
6
7 #写sql语句
8 sql = 'SELECT * FROM book_info'
9 #提交sql语句
10 cursor.execute(sql)
11
12 #接收返回值
13 res = cursor.fetchall() #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
17 #关闭数据库连接
18 db.close()
```

```
C:\ProgramData\Anaconda3\python.exe E:/pythonproject/scrappy01/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

```
1  import pymysql
2
3  while True:
4      print('''
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',
15         database='library')
16     cursor = db.cursor() #建立游标
17
18     if option=='1':
19         #查询功能
20         pass
21
22     elif option=='2':
23         #新增功能
24         pass
25
26     elif option=='3':
27         #修改功能
28         pass
29
30     elif option=='4':
31         #删除功能
32         pass
33
34     elif option=='5':
35         #搜索功能
36         pass
37
38     else :
39         print('退出程序')
40         break
41
42     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

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

```
1 book_id = int(input('请输入书本ID: '))
2
3 #查询数据库,检测书号是否存在
4 sql = 'SELECT * FROM book_info where book_id=%s' % book_id
5 cursor.execute(sql)
6 result = cursor.fetchone()
7
8 #如果书号存在:
9 if result :
10     #打印出准备修改的图书信息
11     print('您要修改的图书为: ')
12     print('{}\t{}\t{}\t{}\t{}\t{}\t{}'.format(result[1], result[2],
13         result[3], result[4], result[5], result[6]))
14     print('\n')
15
16     #接收新的图书信息(用于更新)
17     book_name = input('请输入书名: ')
18     first_author = input('请输入主编: ')
19     second_author = input('请输入副编: ')
20     publisher = input('请输入出版社: ')
21     price = float(input('请输入价格: '))
22     print('您要修改信息为: ', book_id, book_name, first_author, second_author,
23         publisher, price)
24
25     #根据用户选择,是否录入数据库
26     choice = input('确定录入请输入Y,放弃录入请输入N: ')
27     if choice == 'Y' or choice == 'y':
28         sql = "UPDATE book_info SET
29             book_name='{}',first_author='{}',second_author='{}',publisher='{}',price={}
30             WHERE
31             book_id=%d;".format(book_name,first_author,second_author,publisher,price,boo
32                 k_id)
33         cursor.execute(sql)
34         db.commit()
35         print('修改成功')
36
37 #如果书号不存在
38 else:
39     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
4 #删除全表
5 DELETE FROM 表名
```

```
1 book_id = int(input('请输入书本ID: '))
2
3 # 查询数据库, 检测书号是否存在
4 sql = 'SELECT * FROM book_info where book_id=%d' % book_id
5 cursor.execute(sql)
6 result = cursor.fetchone()
7
8 # 如果书号存在:
9 if result:
10     print('您要删除的图书为: ')
11     print('{}\t{}\t{}\t{}\t{}\t{}\t{}'.format(result[1], result[2],
12         result[3], result[4], result[5], result[6]))
13     choice = input('确定删除请输入Y,放弃删除请输入N: ')
14
15     # 根据用户选择, 是否删除
16     if choice == 'Y' or choice == 'y':
17         sql = "DELETE FROM book_info WHERE book_id = %d" %book_id
18         cursor.execute(sql)
19         db.commit()
20         print('删除成功')
21
22     # 如果书号不存在, 返回
23 else:
24     print('书号不存在')
```

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

```

用户模块

新建用户表

对象 user @library (本机) - 表

保存

添加字段

插入字段

删除字段

主键

上移

下移

字段

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
user_id	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
username	varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
real_name	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
phone	bigint	0	0	<input type="checkbox"/>	<input type="checkbox"/>		
sex	varchar	255	0	<input type="checkbox"/>	<input type="checkbox"/>		
register_time	bigint	0	0	<input type="checkbox"/>	<input type="checkbox"/>		
delete	smallint	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

默认:

☒ 自动递增
 ☒ 无符号
 ☐ 填充零

新建user.py

#整体结构

```

1  import pymysql
2
3  while True:
4      print('''
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
14     db = pymysql.connect(host='localhost', user='root', password='123456',
15 database='library')
16     cursor = db.cursor() #建立游标
17
18     if option=='1':
19         #查询功能
20         pass
21
22     elif option=='2':
23         #新增功能
24         pass
25
26     elif option=='3':
27         #修改功能
28         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 :
38         print('退出程序')
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

```

1  import random
2  random.random()           #0-1之间随机数
3  random.randint(3,15)      #3-15之间随机整数
4  random.uniform(3.2,20.6)  #两数之间随机浮点数
5  random.choice('ABCDEFG')  #从序列中随机选择一个
6  random.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
2 pattern1 = r'^[a-z0-9A-Z]{8,20}$'
3 match1 = re.match(pattern1, input('请输入8-20位字母数字用户名: '))
4 # 正则匹配手机号
5 pattern2 = r'^1[3|4|5|8|9]\d{9}$'
6 match2 = re.match(pattern2, input('请输入手机号码: '))
7
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 else:
24     print('请输入正确的用户名和密码')

```

Class 05