## Django ORM使用

“”

首先我们应该有对应软件的思维：

我们在进行正常的数据库操作的时候，首先需要建立表格

在ORM操作中，我们需要使用python中的类来类比表格

类名相当于表格的名称

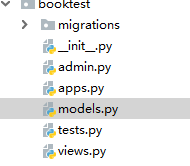
对象相当于表格的行

类添加的每一条属性都相当于表格中的每一列

“”

1. 类与表的转化实现过程
2. 创建类

首先我们创建类需要在给定的models.py模块中创建，双击进入



根据需求，我们依照上方的思想创建类



# models模块中已经为我们引入了models模块

# 我们创建的类需要继承父类Model，这样我们就可以使用封装好的方法进行操作

~ models.CharField(max\_length=20)：创建的列需要的数据是CharField类型的，虽大长度为20

~ models.DateTimeField()：创建的列需要的数据是DateTimeField类型的。

~ default=True：默认值是True

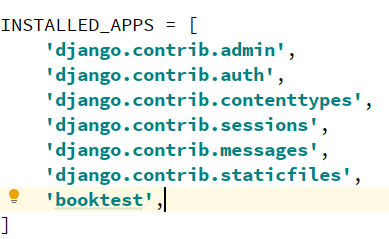
# 我们在创建表时所需的数据类型在这里都有封装，可以直接选择使用

# 我们类属性的名称会作为表的列名

外键的创建：外键名 = Models.Foreignkey(外键关联的类名, on\_delete=models.CASCADE)

1. 类与表的转化

进入setting文件夹中，配置数据库。



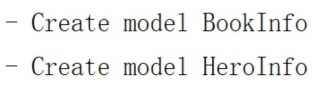
# 在最后一行按照格式将我们创建的应用文件夹名称填写进去

生成迁移文件：根据模型类生成sql语句

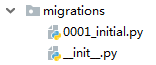
命令：python manage.py makemigrations 回车



显示创建即为成功



我们在migrations文件夹中会出现这样一个文件

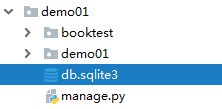


执行迁移：执行sql语句生成表

命令：python manage.py migrate



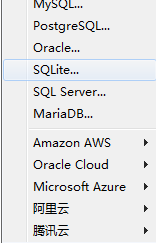
执行成功后我们会在外层目录中找到数据库文件



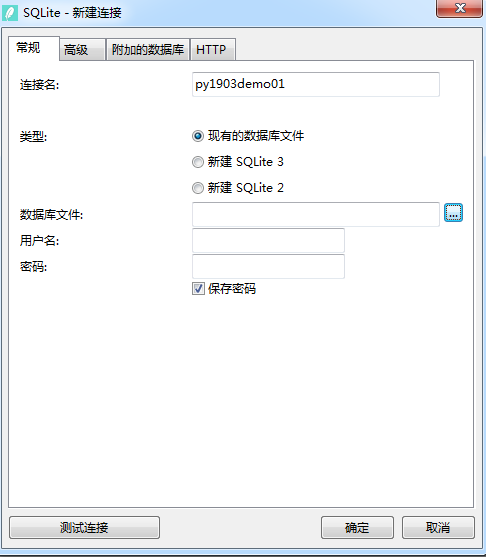
之后我们进入数据库点击连接



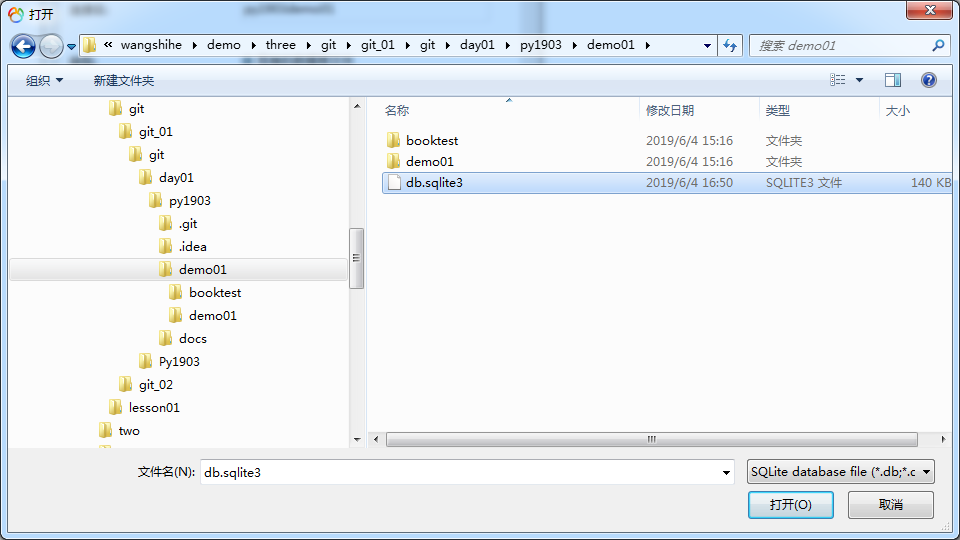
选择SQLite，因为我们python封装的转换是这个数据库的



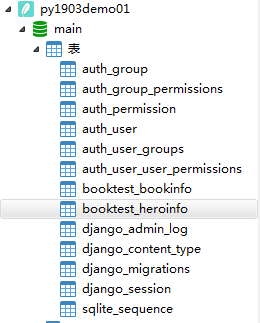
然后我们填写连接名，点击数据库文件的三个小点…



我们找到我们生成的db文件点击打开，然后完成



我们的表就生成完成了



其中booktest开头的两个表就是我们需要的表

1. 增删改查

“”

先进入shell模式

python manage.py shell

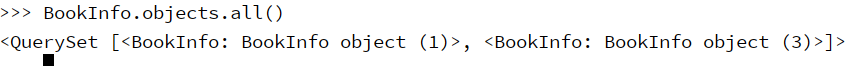
需要引入需要的包：

from booktest.models import BookInfo, HeroInfo

“”

查询查看所有的行：

类名.objects.all()



# 我们可以看到输出的是以列表的形式，而且是以对象的形式输出的，我们可以通过下标获取相应对象，通过对象.属性查看相应属性

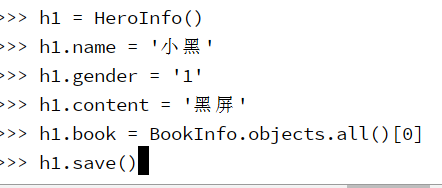
插入：

首先需要创建目标对象

然后对对象的每个属性一一赋值

最后对象.save()

注意，外键的值是目标对象



修改：

创建对象，修改对象数据然后提交

删除：

对象.delete()

关联查询

关系定义在多方

一对多查询：

一的实例.多的类名【小写】\_set.all()



查出的内容仍然是对象列表，可以通过对象获得数据

多对一查询：

多的实例.外键名.外键属性