前置准备

```
1. 本地安装django项目
django-admin startproject drf

2. 创建子应用目录booktest,在项目根目录下执行以下命令
python manage.py startapp booktest

创建一个数据库,book
create database book charset=utf8;
grant all privileges on book.* to root@localhost identified by '123456';
flush privileges;

3. 配置数据库信息
   3.1 在主应用drf目录下的drf/__init__.py添加以下代码:
   import pymysql
   pymysql.install_as_MySQLdb()

   3.2 在项目配置文件中，settings中修改数据链接信息。

   DATABASES = {
    # 'default': {
    #     'ENGINE': 'django.db.backends.sqlite3',
    #     'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    # }
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'drf',
        'USER': 'root',
        'PASSWORD': '123456',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    },
    }
```

1 定义

创建应用booktest，在models.py 文件中定义模型类。

```python
from django.db import models

#定义图书模型类BookInfo
class BookInfo(models.Model):
    btitle = models.CharField(max_length=20, verbose_name='图书标题')
    bpub_date = models.DateField(verbose_name='出版时间')
```

```
    bread = models.IntegerField(default=0, verbose_name='阅读量')
    bcomment = models.IntegerField(default=0, verbose_name='评论量')
    is_delete = models.BooleanField(default=False, verbose_name='逻辑删除')

    class Meta:
        db_table = 'tb_books'  # 指明数据库表名
        verbose_name = '图书'  # 在admin站点中显示的名称
        verbose_name_plural = verbose_name  # 显示的复数名称

    def __str__(self):
        """定义每个数据对象的显示信息"""
        return "图书:《"+self.btitle+"》"

#定义英雄模型类HeroInfo
class HeroInfo(models.Model):
    GENDER_CHOICES = (
        (0, 'female'),
        (1, 'male')
    )
    hname = models.CharField(max_length=20, verbose_name='名称')
    hgender = models.SmallIntegerField(choices=GENDER_CHOICES, default=0,
verbose_name='性别')
    hcomment = models.CharField(max_length=200, null=True, verbose_name='描述信息')
    hbook = models.ForeignKey(BookInfo, on_delete=models.CASCADE, verbose_name='图书')
 # 外键
    is_delete = models.BooleanField(default=False, verbose_name='逻辑删除')

    class Meta:
        db_table = 'tb_heros'
        verbose_name = '英雄'
        verbose_name_plural = verbose_name

    def __str__(self):
        return self.hname
```

1） 数据库表名

模型类如果未指明表名，Django默认以 **小写app应用名_小写模型类名** 为数据库表名。

可通过**db_table** 指明数据库表名。

2） 关于主键

django会为表创建自动增长的主键列，每个模型只能有一个主键列，如果使用选项设置某属性为主键列后django不会再创建自动增长的主键列。

默认创建的主键列属性为id，可以使用pk代替，pk全拼为primary key。

3） 属性命名限制

- 不能是python的保留关键字。
- 不允许使用连续的下划线，这是由django的查询方式决定的。
- 定义属性时需要指定字段类型，通过字段类型的参数指定选项，语法如下：

```
属性=models.字段类型(选项)
```

4）字段类型

类型	说明
AutoField	自动增长的IntegerField，通常不用指定，不指定时Django会自动创建属性名为id的自动增长属性
BooleanField	布尔字段，值为True或False
NullBooleanField	支持Null、True、False三种值
CharField	字符串，参数max_length表示最大字符个数
TextField	大文本字段，一般超过4000个字符时使用
IntegerField	整数
DecimalField	十进制浮点数， 参数max_digits表示总位数， 参数decimal_places表示小数位数
FloatField	浮点数
DateField	日期， 参数auto_now表示每次保存对象时，自动设置该字段为当前时间，用于"最后一次修改"的时间戳，它总是使用当前日期，默认为False； 参数auto_now_add表示当对象第一次被创建时自动设置当前时间，用于创建的时间戳，它总是使用当前日期，默认为False; 参数auto_now_add和auto_now是相互排斥的，组合将会发生错误
TimeField	时间，参数同DateField
DateTimeField	日期时间，参数同DateField
FileField	上传文件字段
ImageField	继承于FileField，对上传的内容进行校验，确保是有效的图片

5） 选项

选项	说明
null	如果为True，表示允许为空，默认值是False
blank	如果为True，则该字段允许为空白，默认值是False
db_column	字段的名称，如果未指定，则使用属性的名称
db_index	若值为True, 则在表中会为此字段创建索引，默认值是False
default	默认
primary_key	若为True，则该字段会成为模型的主键字段，默认值是False，一般作为AutoField的选项使用
unique	如果为True, 这个字段在表中必须有唯一值，默认值是False

null是数据库范畴的概念，blank是表单验证范畴的

6） 外键

在设置外键时，需要通过**on_delete**选项指明主表删除数据时，对于外键引用表数据如何处理，在django.db.models中包含了可选常量：

- **CASCADE** 级联，删除主表数据时连通一起删除外键表中数据
- **PROTECT** 保护，通过抛出**ProtectedError**异常，来阻止删除主表中被外键应用的数据
- **SET_NULL** 设置为NULL，仅在该字段null=True允许为null时可用
- **SET_DEFAULT** 设置为默认值，仅在该字段设置了默认值时可用
- **SET()** 设置为特定值或者调用特定方法，如

```
from django.conf import settings
from django.contrib.auth import get_user_model
from django.db import models

def get_sentinel_user():
    return get_user_model().objects.get_or_create(username='deleted')[0]

class MyModel(models.Model):
    user = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.SET(get_sentinel_user),
    )
```

- **DO_NOTHING** 不做任何操作，如果数据库前置指明级联性，此选项会抛出**IntegrityError**异常

2 迁移

将模型类同步到数据库中。

1）生成迁移文件

```
python manage.py makemigrations
```

2）同步到数据库中

```
python manage.py migrate
```

3 添加测试数据

建议把下面添加测试数据的sql语句写在pycharm提供的数据库工具中，而不是cmd命令行里面。

原因是错误了，不好修改。

```
insert into tb_books(btitle,bpub_date,bread,bcomment,is_delete) values
('射雕英雄传','1980-5-1',12,34,0),
('天龙八部','1986-7-24',36,40,0),
('笑傲江湖','1995-12-24',20,80,0),
('雪山飞狐','1987-11-11',58,24,0);


insert into tb_heros(hname,hgender,hbook_id,hcomment,is_delete) values
('郭靖',1,1,'降龙十八掌',0),
('黄蓉',0,1,'打狗棍法',0),
('黄药师',1,1,'弹指神通',0),
('欧阳锋',1,1,'蛤蟆功',0),
('梅超风',0,1,'九阴白骨爪',0),
('乔峰',1,2,'降龙十八掌',0),
('段誉',1,2,'六脉神剑',0),
('虚竹',1,2,'天山六阳掌',0),
('王语嫣',0,2,'神仙姐姐',0),
('令狐冲',1,3,'独孤九剑',0),
('任盈盈',0,3,'弹琴',0),
('岳不群',1,3,'华山剑法',0),
('东方不败',0,3,'葵花宝典',0),
('胡斐',1,4,'胡家刀法',0),
('苗若兰',0,4,'黄衣',0),
('程灵素',0,4,'医术',0),
('袁紫衣',0,4,'六合拳',0);
```