**关于django数据库的连接和使用：**

1. mysql数据库在jango中的使用，不需要单独创建对象，配置setting文件就可以了。
2. jango中的数据库有两种操作形式，其中之一就是使用原生的sql语句进行操作，在需要使用sql操作的界面里面使用以下的函数：原生界面使用以下的函数。

From django.db import connection

Cursor= connection.cursor() 获取游标

Cursor.execute(“select\*from book”) 得到游标后执行sql语句

Rows = cursor.fetchall() 获取到所有的数据

For row in rows: 遍历获取的数据

游标的常见的几种API使用：

Description: cursor.description得到的一个列表，和元祖

Rowcount:代表的是在执行了sql语句之后受到影响的行数

Close:关闭游标 execute:执行后面的sql语句

Fetchone：获取第一条数据

Fetchall:获取所有满足的数据 这两句话要在查询语句之后用

ORM模型在jango中的使用，也就是为了防止项目过大的时候，使用了过多的sql语句。

首先简单来说orm模型：对象映射关系，这个关系可以方便通过类的方法去操作数据库，即将数据库中的表映射成类，表中的行映射成实例，行中的字段映射成属性。

这个在django中使用的就是模型类,继承自models.Model，这个模型中有不同的字段，相对数据库会生成相应的行。

重要的是，定义的模型必须继承自django.db models

生成之后，需要进行迁移和更改。

Model中常用的字段，autofield,自动增长的属性

Datefield------传递下面连个参数 auto\_now 保存这个属性的时间， auto\_now\_add 对这个属性添加进行的时候，时间。

Datetimefield----日期时间类型，映射到数据库中是datetime属性。

URLfield------用来存储url格式的字符串

这些字段都可以设置的参数是最大长度

字段的常见属性：

1. null 即是在进行传递的参数时候，确定指定的是否为空，如果是则可以出现空值，会用一个空的字符串””存储进去。

如果变成true打开的话，会出现null或者“”两种情况。

1. Blank 和Null的区别就是一个是数据库存储级的，另一个就是表单的验证级。
2. Var\_name: 数据库中的名字---这个字段
3. Default: 默认值，既可以是函数，又可以是一个普通的值。

更多的属性可以从网上查询。

meta类的设置：

主要目的就是为了，让这个属性起到控制模型的作用，比如说我们想要指定自己想要的表名，那么可以在这个类中添加一个db\_table的属性。

外键：Foreignkey(to,on\_delete, options)

里面包含的三个参数，to表示引用的是哪个模型，to是一个函数，需要自己进行设定外键的值。

On\_delete 就是在外键引用的模型被删除后，我们怎么处理

即是在对方的模型删除之后，我们进行什么样的操作：

On\_delete=

1. CASCADE ，级联操作，如果对应的那条数据删除，那么这条数据也会被删除
2. PROTECT： 收到保护的数据，即如果这条数据引用了外键的数据，那么不能删除被引用的数据
3. Set\_null : 设置的空，如果外键那条数据删除了，那么这条数据会自动变成空的
4. Set\_default: 设置默认值，如果外键的那条数据被删除了，那么本条数据就将这条数据设置为默认值，不过前提需要的设定这条数据的默认值。
5. SET 如果外键的值被删除了，那么就会自动获取set函数中包含的值来作为外键的值，set()可以接受一个函数或者可以调用的对象。
6. 在使用二级评论之类的功能的时候，需要将外键设定外函数的本身，即函数的重复调用。

表单之间的关系，包括一对多，一对一或者多对多

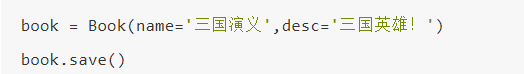
一对多就是简单来说，那些比较容易引用的。

OnetoOnefiled 模型或者 one to many field

Related\_name 如果想要一个文章想要访问文章的话，可以通过user.article\_set来进行访问，命名规则是模型的名字小写\_set。

添加数据到数据库中，使用的是模型方法：

直接创建一个模型对象，将数据直接写入就可以。



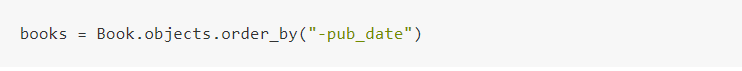
上面的BOOK是一个模型类，book一个BOOK的对象将内容写入

查找对象使用的都是object方法，例如：object.all object.filter...

在使用过滤方法的时候，需要把条件写在filter后面的（）中，

只获得一个条件的对象使用的就是get方法

排序：object.order\_by



因为上面使用的-号，所以排序方式使用的将pub\_date进行从小到大排序。

将查找到的数据属性进行修改后，就可以直接进行保存。

于.save()方法对立的方法就是.delete()方法-----直接将找到的数据进行删除。

查询使用的api不单单是filter, exclude, get三个方法。

精确查找的条件 art=Art.object.get(id\_exact=14)

找到id是14的数据，xx\_exact 精确寻找。

Iexact 指的就是模糊查找

contains包含某项数据 i+contains就是不包含

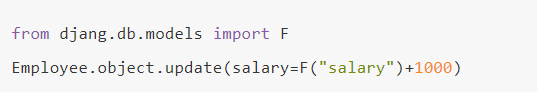
In 指的相对的对象是否在指定的值中

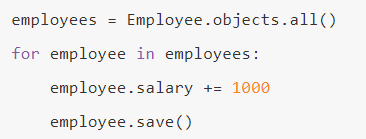
聚合函数，类似于求取满足条件的整体的函数

Sum max count都是可以获取的聚合函数

F和Q表达式：

F表达式直接将数据保存到数据库中，不需要提取出在修改，再保存





就像上面两个方式的对比，就是使用f和不适用f的区别。

Q表达式就是简单的复杂条件filter，即多个筛选条件，

通过模型名字进行查询，那么.object其实是一个manager从queryset类中拷贝过来的，

其中all filter 都是queryset类的方法

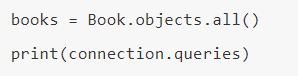
而且这些方法是可以进行链式调用的，其中需要满足的条件就是，会返回新的queryset对象：

Filter exclude 都属于可以返回新的对象的方法。

除此之外还有许多满足条件的函数方法，这些都是对数据库能够进行操作的函数。

queryset对象生成的时候，并不会立刻进行执行。

需要在一定条件的情况下，才会执行：



如上，这样并不会执行，（查询类）

需要执行的条件就是：

1. 遍历，迭代 2
2. 切片，这个执行之后不可以执行filter方法
3. len函数，长度
4. list函数，将函数转化成列表也会执行
5. 判断，对获得的queryset 对象执行

ORM进行迁移：

Makemigrations migrate

Showmigrations 查看app下的迁移文件

Sqlmigrate 查看迁移文件到数据库时，转移的sql语句