

Day03

1. 增

```
BookInfo.objects.create(  
    name = '射雕英雄后传'  
    pub_date = '2020-2-2'  
)
```

2. 删

```
BookInfo.objects.filter(id=1).delete()  
BookInfo.objects.get(id=1).delete()
```

3. 改

```
BookInfo.objects.filter(id=1).update(  
    name = '射雕英雄前传'  
    pub_date = '1990-2-2'  
)
```

4. 查

- 基本查询

1. `all` 查询全部结果

```
BookInfo.objects.all()
```

2. `count` 查询结果数量

```
BookInfo.objects.all().count() ==> BookInfo.objects.count()
```

- 过滤查询

- 1. `filter` 过滤出多个结果 如果不存在模型类, 会返回空列表

```
BookInfo.objects.filter(id=1)
```

2. `exclude` 排除掉符合条件剩下的结果

```
BookInfo.objects.exclude(id=1)
```

3. `get` 查询单个结果 如果不存在模型类, 会抛出异常

```
BookInfo.objects.get(id=1)
```

- 条件:

1. `exact`: 表示相等
`BookInfo.objects.filter(id__exact=1) ==>`
`BookInfo.objects.filter(id=1)`
2. `contains`: 表示包含
`BookInfo.objects.filter(name__contains='传')`
3. `startswith`: 以指定值开头
`BookInfo.objects.filter(name__startswith='射')`
4. `endswith`: 以指定值结尾
`BookInfo.objects.filter(name__endswith='传')`
5. `isnull`: 判断是否为空
`BookInfo.objects.filter(name__isnull=True)`
6. `in`: 是否包含在范围内
`BookInfo.objects.filter(id__in=[1, 3, 5])`
7. 大小比较
`gt ==>` 大于
`gte ==>` 大于等于
`lt ==>` 小于
`lte ==>` 小于等于
`BookInfo.objects.filter(id__gt=1)`
8. 日期查询 对日期属性进行运算
`BookInfo.objects.filter(pub_date__year=2020)`

• F和Q对象

- `# 导入F和Q对象`
`from django.db.models import F, Q`

`# F: 两个属性的比较`
`BookInfo.objects.filter(readcount__gt=F('commentcount'))`

`BookInfo.objects.filter(readcount__gt=F('commentcount')*2)`

`# Q: 表示逻辑关系`
`& ==> and | ==> or ~Q ==> not`
`BookInfo.objects.filter(Q(id__gt=2) & Q(readcount__gt=20))`

• 聚合函数

- `# 导入聚合函数`
`from django.db.modles import Sum, Avg, Max, Min, Count`
`# BookInfo.objects.aggregate(聚合函数(条件))`

• 排序函数

- # order_by 默认是升序
 BookInfo.objects.all().order_by('readcountnt') # 升序排序
 BookInfo.objects.all().order_by('-readcountnt') # 降序排序