Django F对象和Q对象查询详解

F对象查询与Q对象查询，刚看到大家一定会感到很陌生，其实它们也是 Django 提供的查询方法，而且非常的简单的高效，对于一些特殊的场景需求应用起来非常的合适，在本节我们将对这两种查询方法进行讲解，帮助大家掌握它们的使用方法以及适合应用的场景。

1. F对象查询

F对象主要用于模型类的 A 字段属性与 B 字段属性两者的比较，即操作数据库中某一列的值。通常是对数据库中的字段值在不获取的情况下进行操作。F 对象内置在数据包django.db.models中，所以使用时需要提前导入。如下所示：

from django.db.models import F

它的语法格式如下所示：

```
from django.db.models import F
F('字段名')
```

在使用F对象进行查询的时候需要注意：一个 F() 对象代表了一个 Model 的字段的值；F 对象可以在没有实际访问数据库获取数据值的情况下对字段的值进行引用。

Django 支持对 F对象引用字段的算术运算操作，并且运算符两边可以是具体的数值或者是另一个 F 对象，下面我们通过实例进一步认识 F 对象。

```
from django.db.models import F
from index.models import Book
#给Book所有实例价格 (retail_price) 涨价20元
Book.objects.all().update(retail_price=F('retail_price')+20) #获取该列所有值并加20
#利用传统的方法实现涨价20元
books = models.Book.objects.all()
for book in books:
    book.update(retail_price=book.retail_price+20)
    book.save()
```

通过上述实例可以看出，使用 F 对象相对传统的方法要简单的多。那么如何通过 F 对象实现两个字段值（列）之间的比较呢？实例如下所示：

```
#对数据库中两个字段的值进行比较, 列出哪儿些书的零售价高于定价
books = Book.objects.filter(retail_price__gt=F('price'))
for book in books:
    print(book.title, '定价:', book.price, '现价:', book.retail_price)
```

2. Q对象查询

Q 对象相比 F 对象更加复杂一点，它主要应用于包含逻辑运算的复杂查询。Q 对象把关键字参数封装在一起，并传递给 filter、exclude、get 等查询的方法。多个 Q 对象之间可以使用&或者|运算符组合（符号分别表示与和或的关系），从而产生一个新的 Q 对象。当然也可以使用~（非）运算符来取反，从而实现NOT查询。Q 对象的导入方式如下所示：

from django.db.models import Q

它和 Q 对象位于一个数据包里面。常用语法格式如下：

```
from django.db.models import Q
Q(条件1)|Q(条件2)  # 条件1成立或条件2成立
Q(条件1)&Q(条件2)  # 条件1和条件2同时成立
Q(条件1)&~Q(条件2)  # 条件1成立且条件2不成立
#...等
```

最简单的 Q 对象的使用方法是将单个字段类属性作为参数进行查询，实例如下：

```
#查询 书籍的title中包含有字母P的
In [1]: from index.models import Book
In [2]: from django.db.models import Q
In [4]: Book.objects.filter(Q(title__contains="P"))
Out[4]: <QuerySet [<Book: Book object (1)>]>
```

但时 Q 对象在实际的应用中往往是较为复杂的，和常和逻辑运算符一起使用，如下所示：

```
#多个Q对象组合
from index.models import Book
from django.db.models import Q
#查找c语言中文网出版的书或价格低于35的书
Book.objects.filter(Q(retail_price__lt=35)|Q(pub_id='2'))#两个Q对象是或者的逻辑关系
#查找不是c语言中文出版的书且价格低于45的书
Book.objects.filter(Q(retail_price__lt=45)&~Q(pub_id='2'))#条件1成立条件2不成立
```

注意此时的的字段为pub_id 因为此此段建立一对多的关联关系，不可以直接使用C语言中文网，不然会发生如下报错：

ValueError: invalid literal for int() with base 10: 'c语言中文网出版'

报错原因是：字段的条件约束为int类型，但是给了一个字符串类型。

Q 对象也可以与类属性的字段名组合在一起使用，但是在这种情况下，Django 规定，Q 对象必须放在前面，示例如下：

```
Book.objects.filter(Q(price__lte=100),title__icontains="p")#组合使用
<QuerySet [<Book: Book object (1)>]>
```

本节详解介绍了 F 与 Q 对象的查询方法以及是如何进行应用的，熟练掌握它们会给我们在开发过程中带来事半功倍的效果，能够大量简化我们的工作量。所以我们要尽可能的学习并掌握它们。