Django存在关联关系的反向查询

在《Django数据表关联关系映射（一对一、一对多、多对多）》一节中，我们介绍过 Model 之间存在三种关系模型用来维护表与表之间的关联。同时，Django 也为此提供了非常强度大关联关系查询，在实际工作中，大多情况下数据表之间都会存在关联，所以学习关联关系查询就显的尤为重要。在本节我们将逐一进行介绍，希望小伙伴可以学会所得。

1. Model的反向查询

Django 中的每一中关联关系都可以是实现反向查询，我们对三种关系的关联反向查询依次进行介绍，其实只要掌握了其中一种，大家就可以领悟反向的查询是如何进行应用的。

1) 多对一关系的反向查询

```
#通过多查一创建书籍书籍实例对象（正向查询）
In [12]: from index.models import PubName,Book
In [13]: book1=Book.objects.get(id=1)
In [14]: print(book1.title,"的出版社是",book1.pub.pubname)#建立外键关联的字段
#输出结果：Python Django 的出版社是 清华出版社
#通过一查多创建出版社实例对象（反向查询）
In [16]: pubname1=PubName.objects.get(pubname="C语言中文网出版")
In [17]: books=pubname1.book_set.all()
In [18]: for book in books:
   ...:     print(book.title)
#输出结果：Django Flask
```

通过上述的反向查询可以得知，对于每一个 PubName 的实例对象都自动地会有一个管理器可以用来查询与它关联的 Book 实例对象。在默认情况下，管理器的名称是“小写模型名_set”。

在上例中管理器就是 book_set，我们利用它就可以实现一对多的反向查询。book_set 管理器同查询管理器 objects 一样可以调用相应的 API 接口。如下所示，不过它都对应的使用场景也不能乱用哦。

```
xxxfiled.book_set.all() #查询所有数据
xxxfiled.book_set.filter() #查询满足特定条件的数据
author.book_set.create(book) # 创建新书并联作用author
author.book_set.add(book)  #添加已有的书为当前作者author
author.book_set.clear()  #删除author所有并联的书
```

2) 多对多关系的反向查询

我们知道一个作者可以出版多本图书，同时一本图书可以被多名作者同时编写。这种关系就是多对多的关系，那么它的多对多关系又是如何实现的呢？

```
#通过Author(Luncy)查询对应的所有的Books
In [5]: from index.models import Book,Author
In [6]: author=Author.objects.get(id=1)
In [7]: author.books.all()
Out[7]: <QuerySet [<Book: Book object (1)>, <Book: Book object (2)>, <Book: Book object (3)>, <Book: Book object (4)>, <Book: Book object (5)>]>
#通过book(Python Django)的书籍查询其对应的所有的 Authors
In [8]: book=Book.objects.get(id=1)
In [9]: book.author_set.all()
Out[9]: <QuerySet [<Author: 作家：Luncy>, <Author: 作家：Tom>]>
In [10]: book.author_set.add(Author.objects.get(name="Xiaolong"))#为id=1书籍添加作者Xionglong
```

3) 一对一关系的反向查询

一对一的反向查询比较特殊，它的管理器是一个单一的对象么不是对象的集合密，且名称变成了小写的 Model 名。下面我们通过举例进行说明：

```
In [1]: from index.models import UserInfo,ExtendUserinfo
In [2]: user=UserInfo.objects.get(id=2)
In [3]: user.extenduserinfo
Out[3]: <ExtendUserinfo: ExtendUserinfo object (1)>
```

我们知道 ExtendUserinfo 的 user 字段与 UserInfo 存在一对一的关联关系，所以反向查询是利用 UserInfo 的实例对象来查询 ExtendUserinfo Model 中与其匹配的对象。

提示：所谓反向查询可以简单理解为在建立关联关系的两个 Model 之间，利用没有关联关系字段的 Model 来查询另一个有关联关系字段的 Model。一般的操作时首先实例化对象，然后根据不同的关联关系选用不同的方法，从而实现查询。

2. 跨关联关系查询

那我们考虑一下，有没有这种场景，在存在关联关系的两个 Model 之间进行查询的时候，不能只考虑其中一个 Model，而需要满足两个 Model 的要求呢。比如，要查询“c语言中网”出版了哪几本图书，我们应该如何通过 Django 查询呢？Django 为我们提供了一种简单的方法，也就是跨关联关系的查询方式，这种方式使用双下画线的与关联的 Model 的字段名称组合在一起，并给出合适的条件就可以完成查询。实例如下：

```
#“c语言中网”出版了哪几本图书
from index.models import PubName,Book
In [1]: Book.objects.filter(pub__pubname__contains="c语言中文网出版")#注意是双下画线哦
Out[1]: <QuerySet [<Book: Book object (4)>, <Book: Book object (5)>]>
```

同时 Django 也支持反向的关联查询，只需要使用关联 Model 的小写名称即可，实例如下：

```
#查询价格大于等于30的书籍
In [2]: PubName.objects.filter(book__price__gte=60)
Out[2]: <QuerySet [<PubName: PubName object (9)>]>
```

本节详细介绍了如何使用多表关联关系查询，这在实际的开发工作中很重要，所以大家要重点掌握本节知识。