链滴

Django 之奇技淫巧 Model

作者：zonghua

原文链接：https://ld246.com/article/1459263179627

来源网站：链滴

Django Model

1. RuntimeWarning: DateTimeField received a naive d tetime

USE_TZ = False

禁用掉时区设置，通通使用系统时间，所以要保证设置了正确的系统时间

TIME_ZONE = 'Asia/Shanghai

如果启用了时区设置，可以设置应用独立的时间

2. 级联删除

```
wechat_user = models.ForeignKey(Wechat_User,verbose_name=u'微信',blank=True,null=True
on_delete=models.SET_NULL)
```

CASCADE: 默认级联删除

PROTECT: 通过抛出django.db.models.ProtectedErrordjango.db.models.ProtectedError错误来止删除关联的对象

SET_NULL: 前置条件需要设定null=True，设置ForeignKey 为 Null

SET_DEFAULT: 前置条件需要先设定默认值，设置 ForeignKey 为默认值

SET: 设置为某个方法返回的值

DO_NOTHING: 保留原样，如果再次把外键的对象添加回来，会再次显示。如果数据库设置必须关则会报IntegrityError错。

```
||13    ||哈哈哈哈哈                    ||2016-04-02 23:50:51    ||2 || 1||
|| 14    ||哈哈哈哈哈哈              || 2016-04-02 23:56:19     ||2 ||    1||
||15    ||恍恍惚惚恍恍惚惚机会    ||2016-04-03 11:47:21  ||2  ||
```

上表中13 14 是设定了DO_NOTHING；15 是设定了SET_NULL

3. 关联查询

models.py

```
class School(models.Model):
    name = models.CharField(max_length=30, verbose_name='school')

class Deparment(models.Model):
    name = models.CharField(max_length=30, verbose_name='department')
    school = models.ForeignKey(School, verbose_name='school')

class StudentClass(models.Model):
    name = models.CharField(max_length=30, verbose_name='class')
```

```
    department = models.ForeignKey(Deparment, verbose_name='department')

class StudentYear(models.Model):
    year = models.CharField(max_length=30, verbose_name='year')

class Student(models.Model):
    name = models.CharField(max_length=30, verbose_name='student')
    studentClass = models.ForeignKey(StudentClass, verbose_name='class')
    year = models.ForeignKey(StudentYear, verbose_name='year')
```

查询的使用用双下划线去关联外键属性，只能过滤id的值

```
>>>school_ids = School.objects.filter(name='EE').values('id')
>>>year_ids = StudentYear.objects.filter(year='2016').values('id')
>>>student = Student.objects.filter(studentClass__department__school_id__in=school_ids, ye
r_id__in=year_ids).first()
        <Student object>
```

查询是懒惰的，只会在最终序列的时候执行拼接后的语句

```
utils-89 DEBUG : (0.001) QUERY
```