

一、数据库表的创建

1、创建表的代码

```
from django.db import models
```

```
class Nav (models.Model):
```

```
    name=models.CharField(max_length=10,unique=True,verbose_name="导航名称") #int可以不加长度，字符串类型必须加长度
```

```
    url=models.CharField(max_length=100,unique=True,verbose_name='导航链接')
```

```
    classname=models.CharField(max_length=50,unique=True,verbose_name='导航图标')
```

```
    is_delete=models.SmallIntegerField(default=1,verbose_name='是否被删除')
```

```
    create_time=models.DateTimeField(verbose_name='创建时间',auto_now_add=True) #auto_now_add插入数据自动取当前时间
```

```
    update_time=models.DateTimeField(verbose_name='修改时间',auto_now=True)#auto_now 修改时间自动取当前时间
```

```
    def __str__(self):
```

```
        return self.name
```

```
class Meta:
```

```
    verbose_name='导航表'
```

```
    verbose_name_plural=verbose_name
```

```
    db_table='nav' #指定表名
```

```
    ordering=['update_time'] #查询数据的时候，用来排序的
```

2、运行命令，生成表

```
python manage.py makemigrations 表结构
```

```
python manage.py migrate 创建数据库表
```

二、表数据增加

#新增，批量插入，可通过循环控制

```
models.test.objects.create(name='首页 8',url='index8') #新增第一种方法
```

```
models.test(name='首页 9',url='index9').save() #第二种新增数据方法
```

```
for i in range(10,100):
```

```
    models.test(name='首页%s'%i, url='index%s'%i).save()
```

三、表数据删除

#删除

```
models.test.objects.all().delete() #删除全部
```

```
models.test.objects.filter(id=3).delete() #指定删除
```

四、表数据修改

#修改

```
n=models.test.objects.get(id=100) #第一种修改方式
```

```
n.name="wode"
```

```
n.save()
```

#

```
models.test.objects.all().update(is_delete=0) #第二种修改方式
```

五、表数据查询

查询

get方法查询的话，必须的保证这个数据查出来的只有1条，才可以

```
result = models.test.objects.get(name="wode")
```

```
result2 = models.test.objects.get(id=109)
```

```
print(result)
```

```
print(result2)
```

filter 可以查询多条数据

```
result = models.test.objects.filter(is_delete=0)
```

```
print(result)
```

模糊查询

```
result = models.test.objects.filter(name__contains='w', id__gte=1) # gt大于, lt 小于, gte大于等于, lte小于等于, range范围之间, in在什么内
```

```
result = models.test.objects.exclude(id=100) # exclude排除, 不等于
```

```
all = models.test.objects.all()
```

```
print(result)
```

```
print(all)
```