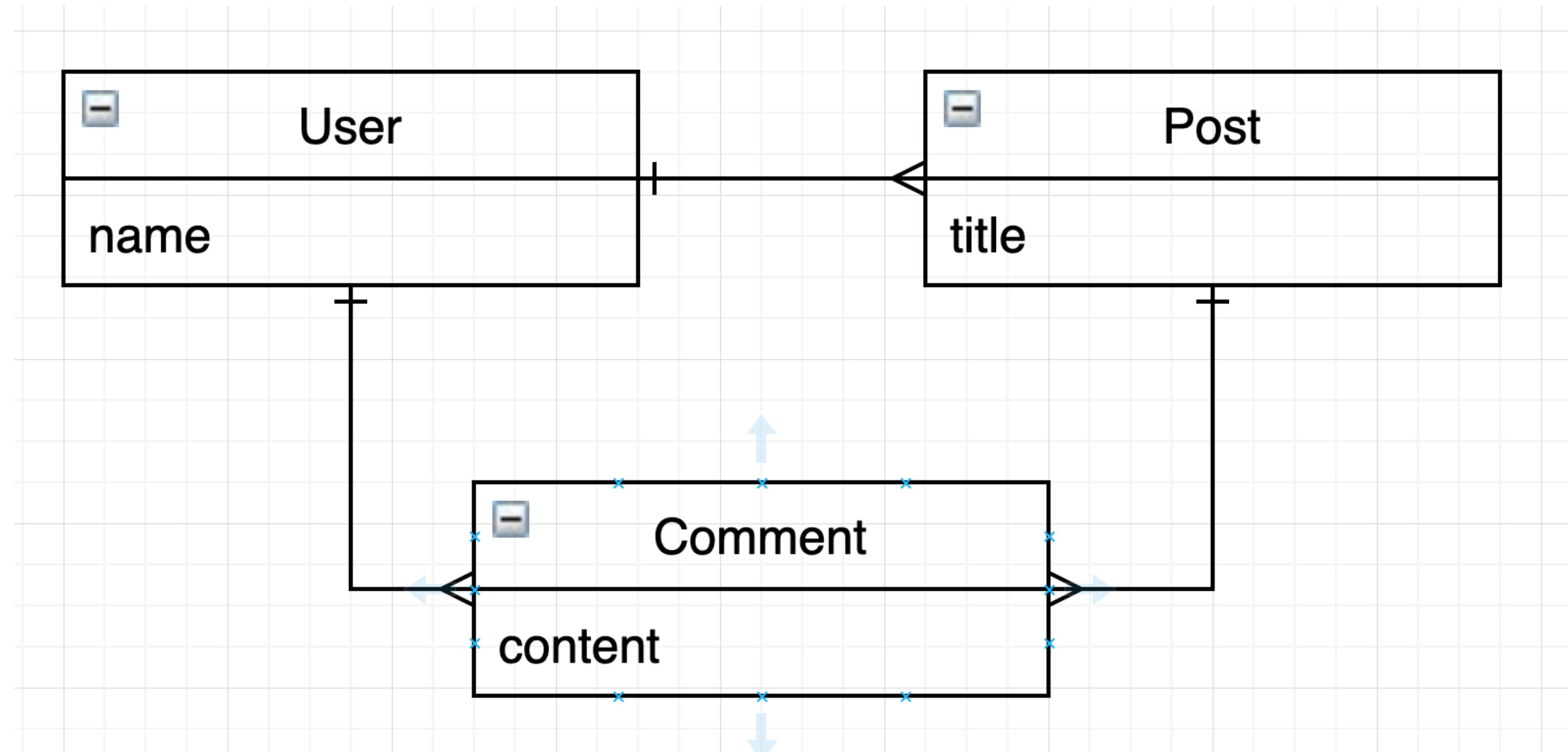


Relationship

startproject
startapp onetomany

settings.py => INSTALLED_APPS

models.py



models.py

```
class User(models.Model):  
    name = models.CharField(max_length=20)  
  
class Post(models.Model):  
    title = models.CharField(max_length=20)  
    user = models.ForeignKey(User, on_delete=models.CASCADE)  
  
class Comment(models.Model):  
    content = models.CharField(max_length=20)  
    post = models.ForeignKey(Post, on_delete=models.CASCADE)  
    user = models.ForeignKey(User, on_delete=models.CASCADE)
```

migration

pip install django_extensions

INSTALLED_APPS

python manage.py shell_plus

데이터를 추가해보자!

이름이 싸피인 User를 생성하여 변수 user에 저장

**user가 “안녕하세요” 라는 제목의
Post 생성하여 변수 post에 저장**

**user가 “반갑습니다” 라는 내용의
Comment 생성하여 변수 comment에 저장**

**db.sqlite3 삭제
migrate**

```
user1 = User.objects.create(name='Kim')
user2 = User.objects.create(name='Lee')
post1 = Post.objects.create(title='1글', user=user1)
post2 = Post.objects.create(title='2글', user=user1)
post3 = Post.objects.create(title='3글', user=user2)
c1 = Comment.objects.create(content='1글1댓글', user=user1, post=post1)
c2 = Comment.objects.create(content='1글2댓글', user=user2, post=post1)
c3 = Comment.objects.create(content='1글3댓글', user=user1, post=post1)
c4 = Comment.objects.create(content='1글4댓글', user=user2, post=post1)
c5 = Comment.objects.create(content='2글1댓글', user=user1, post=post2)
c6 = Comment.objects.create(content='!1글5댓글', user=user2, post=post1)
c7 = Comment.objects.create(content='!2글2댓글', user=user2, post=post2)
```

https://zzu.li/user_post_comment

ID가 1인 유저

User.objects.get(pk=1)

User.objects.get(id=1)

1번 사람이 작성한 게시글은?


```
user1.post_set.all()
```

1번 사람의 모든 게시글에 달린
모든 댓글의 내용을 출력

```
for post in user1.post_set.all():  
    for comment in post.comment_set.all():  
        print(comment.content)
```

2번 댓글을 쓴 사람의 이름

c2.user.name

2번 댓글을 쓴 사람의 모든 게시물

```
c2.user.post_set.all()
```

1번 글의 첫번째 댓글을 쓴사람의 이름

`post1.comment_set.first().user.name`

`post1.comment_set.all()[0].user.name`

1번 글의 2번째부터 4번째까지 댓글

```
post1.comment_set.all()[1:4]
```

1번 글의 두번째 댓글을 쓴 사람의
첫번째 게시물의 작성자의 이름

Post1.comment_set.all()[1].user.post_set.all()[0].user.name

1번 댓글의 user 정보만?

Comment.objects.values('user').get(pk=1)

2번 사람이 작성한 댓글을
content의 내림차순으로


```
user2.comment_set.order_by('-content')
```

제목이 '1글'인 게시물

Post.objects.filter(title='1글')

제목에 ‘글’이 들어간 게시물

```
Post.objects.filter(title__contains='글')
```

댓글 중에 해당 글의 제목이 '1글'인 댓글

```
Comment.objects.filter(post__title='1글')
```

댓글 중 해당 글의 제목에
10이 들어가 있는 댓글


```
Comment.objects.filter(post__title__contains='1')
```

020

```
class Profile(models.Model):  
    user = models.OneToOneField(User, on_delete=models.CASCADE)  
    nickname = models.CharField(max_length=20, blank=True)
```

M2M

startapp manytomany

settings.py => INSTALLED_APPS

병원에 오는 사람들을 기록하는 시스템

1:N으로 구현 할 수 있을까?

추가 테이블이 있다면 가능!


```
class Doctor(models.Model):  
    name = models.CharField(max_length=20)  
  
class Patient(models.Model):  
    name = models.CharField(max_length=20)  
  
class Reservation(models.Model):  
    doctor = models.ForeignKey(Doctor, on_delete=models.CASCADE)  
    patient = models.ForeignKey(Patient, on_delete=models.CASCADE)
```

의사의 환자에 바로 접근을 하려면?

장고에게 알려주자

```
class Doctor(models.Model):
    name = models.CharField(max_length=20)

class Patient(models.Model):
    name = models.CharField(max_length=20)
    doctors = models.ManyToManyField(Doctor, through='Reservation')

class Reservation(models.Model):
    doctor = models.ForeignKey(Doctor, on_delete=models.CASCADE)
    patient = models.ForeignKey(Patient, on_delete=models.CASCADE)
```

반대의 경우 불편 π

```
class Doctor(models.Model):
    name = models.CharField(max_length=20)

class Patient(models.Model):
    name = models.CharField(max_length=20)
    doctors = models.ManyToManyField(Doctor, through='Reservation', related_name='patients')

class Reservation(models.Model):
    doctor = models.ForeignKey(Doctor, on_delete=models.CASCADE)
    patient = models.ForeignKey(Patient, on_delete=models.CASCADE)
```

복잡한 구조를 간단하게 재구성!

```
class Doctor(models.Model):
    name = models.CharField(max_length=20)

class Patient(models.Model):
    name = models.CharField(max_length=20)
    doctors = models.ManyToManyField(Doctor, related_name='patients')

# class Reservation(models.Model):
#     doctor = models.ForeignKey(Doctor, on_delete=models.CASCADE)
#     patient = models.ForeignKey(Patient, on_delete=models.CASCADE)
```