

کتاب هایی را میخواهم که میزان تخفیف کتاب بیشتر از ۲۰۰۰ هزار تومن است..

```
Book.objects.filter(discount__gt=2000/F('price'))
```

چگونه یک instances ذخیره شده رو کپی کنیم

```
from copy import deepcopy

book1 = Book.objects.filter(id=1)

book1_copy = deepcopy(book1)

book1_copy.id = None

book1_copy.save()
```

چند مدل داشته باشید که دارای فیلدهای رابطه ای باشد پیشنهاد من همین مدل هایی که در کلاس کار شد.
۱. داده هایی به مدل اضافه کنید

```
from post.models import Post, Category, Tag, Comment

from django.contrib.auth.models import User


t1 = Tag.objects.create(title='tag1')

cat1 = Category.objects.create(title='cat1')

cat2 = Category.objects.create(title='cat2')

cat4 = Category.objects.create(title='cat4')

u1 = User.objects.create(username='Mohammad', password='1111')

p1 = Post.objects.create(title="Hello world", desc="I am new here in this world", owner=u1)
```

۲. جدولی مانند پست رابطه m2m دارد به آن ستون ها مقادیری جدید اضافه کنید

```
p1.tag.add(t1)
```

```
p1.category.add(cat1)
```

۳. می‌خواهیم این روابط پاک کنیم نمونه پاک نشه فقط اون رابطه پاک بشه

```
p1.tag.remove(t1)
```

۴. برای جدول هایی که روابط یک به چند دارند نیز داده ای اضافه کنید.

```
cm1 = Comment.objects.create(title="nice" , desc="I like this post", post=p1, owner=u1)
```

۵. related_name برای جاهایی که نیاز هست بذارید

```
t1.post_set.all()
```

```
post =models.ForeignKey(Post,on_delete=models.CASCADE,related_name="post_comment")
```

```
p1.post_comment.all()
```

۶. کوئری که تمرین میکنید بیشتر از جاهایی باشد که روابط یک به چند یا چند به چند دارند
مثال : تمام پست هایی را پیدا کنید که در موضوع دسته بندی آن ها حاوی حرف a باشد
تمام دسته بندی هایی را پیدا کنید که موضوع پست حاوی b باشد.
کوئری بنویسید که نشان دهد هر پست چند دسته بندی دارد
کوئری بنویسید که پست هایی که دارای تگ x هستند را پاک کنند.
یک مدل پست را انتخاب و برای آن یک دسته بندی جدید تعریف کنید (با استفاده از create)
کوئری بنویسید که shortdesc نال بود آن پست را پاک کند.

```
Post.objects.filter(category__title__contains='a')
```

```
Category.objects.filter(post__title__contains='b')
```

```
cnt = Post.objects.annotate(count_category=Count("category"))
```

```
cnt[0].count_category
```

```
Post.objects.all().aggregate(count_category=Count("category"))
```

```
Post.objects.filter(tag__title='tag4').delete()
```

```
(8, {'post.Post_tag': 2, 'post.Post_category': 3, 'post.Comment': 1, 'post.Post': 2})
```

```
Post.objects.create(title="post2", desc="This is the second post", owner=u1)
```

```
p2 = Post.objects.get(title='post2')
```

```
p2.category.create(title='catpost2')
```

```
Post.objects.filter(shortdesc__isnull=True).delete()
```