# запускаем Shell

1. python manage.py shell

# импортируем все модели (так делать не стоит (импортировать все), но в данном случае терпимо)

1. from news\_db.models import \*

# создаем двух “юзуров”

1. u1 = User.objects.create\_user(username = 'Kirill')
2. u2 = User.objects.create\_user(username = 'Olga')

# создаём два объекта модели Author

1. a1 = Author.objects.create(user=u1)
2. a2 = Author.objects.create(user=u2)

# создаём четыре категории в модели Category

1. Category.objects.create(name='Спорт')
2. Category.objects.create(name='Политика')
3. Category.objects.create(name='Образование')
4. Category.objects.create(name='Досуг')

# добавляем две статьи и одну новость в модель Post

1. Post.objects.create(author=a1, title='Some title', text='Some text')
2. Post.objects.create(author=a2, title='Another article title', text='Text text text')
3. Post.objects.create(author=a1, position='NW', title='News title', text='Sports News')

# присваиваем к новостям и статьям категории

1. post1 = Post.objects.all()[0]
2. post2 = Post.objects.all()[1]
3. post3 = Post.objects.all()[2]
4. cat1 = Category.objects.get(name='Досуг')
5. cat2 = Category.objects.get(name='Спорт')
6. cat3 = Category.objects.get(name = 'Образование')
7. cat4 = Category.objects.get(name = 'Политика')
8. post1.category.add(cat1)
9. post1.category.add(cat3)
10. post2.category.add(cat4)
11. post3.category.add(cat2)

# создаем комментарии к постам

1. Comment.objects.create(post=post1, user=u1, text='Fire!!!')
2. Comment.objects.create(post=post2, user=u1, text='Bomb!!!')
3. Comment.objects.create(post=post3, user=u1, text='Awesome!!!')
4. Comment.objects.create(post=post3, user=u2, text='What a trash...')
5. Comment.objects.create(post=post1, user=u2, text='Rubbish')
6. Comment.objects.create(post=post2, user=u2, text='Hate this one')

# применяем функцию like()/dislike() к постам и комментариям

1. post1.like()
2. post2.dislike()
3. Comment.objects.get(text='Bomb!!!').like()
4. Comment.objects.get(text='Rubbish').dislike()

# смотрим рейтинг поста и комментария

1. post3.rate
2. Comment.objects.get(text='Rubbish').rate

# обновляем рейтинг пользователя

1. a1.update\_rating()

# выводим лучшего пользователя, используем annotate для создания Алиаса “username”

1. Author.objects.annotate(username=F('user\_\_username')).values('username', 'rate').order\_by('-rate').first()

# импортируем TruncDate для вывода только даты из поля DateTimeFeild

1. from django.db.models.functions import TruncDate

# получаем лучший по рейтингу пост, включая username автора, дату публикации, рейтинг и заголовок

1. Post.objects.annotate(date=TruncDate('create\_date'), username=F('author\_\_user\_\_username')).values('date', 'username', 'rate', 'title') .order\_by('-rate').first()

# получаем все комментарии с датой публикации, username, рейтингом и текстом к предыдущей статье

1. Post.objects.order\_by('-rate').first().comment\_set.annotate(date=TruncDate('create\_date'), username = F('user\_\_username')).values('date', 'username', 'rate', 'text')