>>> from News.models import \*

1. Создать двух пользователей

>>> from django.contrib.auth.models import User

>>> User.objects.create\_user(username='Michael Scott')

>>User.objects.create\_user(username='THE ART NEWSPAPER RUSSIA')

//перс чисто для кринж комментов, не берем в расчет

>>> User.objects.create\_user(username='Dwight Kurt Schrute III')

1. Создать два объекта модели Author, связанные с пользователями.

author1=Author.objects.create(author=User.objects.get(username='THE ART NEWSPAPER RUSSIA'))

author2=Author.objects.create(author=User.objects.get(username='Michael Scott'))

1. Добавить 4 категории в модель Category.

>>> Category.objects.create(category\_name='Наука')

>>> Category.objects.create(category\_name='Искусство')

>>> Category.objects.create(category\_name='Социология')

>>> Category.objects.create(category\_name='Экономика')

1. Добавить 2 статьи и 1 новость.

>>> post1 = Post.objects.create(author=author1, news\_type='AR',title='Искусственный интеллект отличил подлинник от подделки',text='some text')

>>> post2= Post.objects.create(author=author2, news\_type='AR',title='Идеальный ли

дер. Как сочетать в себе диктатора, формалиста, либерала, демократа и создать эффективную команду разработки',text='some text'

>>> post3 = Post.objects.create(author=author2, news\_type='NE', title = 'Samsung и TSMC хотят построить заводы в ОАЭ на сумму $100 млрд', text='some text')

1. Присвоить им категории (как минимум в одной статье/новости должно быть не меньше 2 категорий).

>>> science = Category.objects.get(category\_name='Наука')

>>> art = Category.objects.get(category\_name='Искусство')

>>> sociology = Category.objects.get(category\_name='Социология')

>>> economics = Category.objects.get(category\_name='Экономика')

>>>post1.category.add(science,art)

Проверка:

>>> post1.category.all()

<QuerySet [<Category: Category object (1)>, <Category: Category object (2)>]>

>>post2.category.add(sociology)

>>> post3.category.add(economics)

1. Создать как минимум 4 комментария к разным объектам модели Post (в каждом объекте должен быть как минимум один комментарий).

>>> user1 = User.objects.get(username='Michael Scott')

>>> comment1 = Comment.objects.create(post=post1,user=user1,text="'THAT'S WHAT SHE SAID")

>>> user2 = User.objects.get(pk=3)

>>> comment2 = Comment.objects.create(post=post2, user=user2,text='Есть над чем подумать...')

>>> comment3 = Comment.objects.create(post=post3, user=user2, text='Логичнее было бы выбрать Скрэнтон.')

>>> comment4 = Comment.objects.create(post=post1, user=user2, text='Война человечества против ИИ неизбежна...Я готовлюсь.')

1. Применяя функции like() и dislike() к статьям/новостям и комментариям, скорректировать рейтинги этих объектов.

>>> comment1.like()

>>> comment1.like()

>>> comment1.like()

>>> comment1.like()

>>> comment1.like()

>>> comment2.like()

>>> comment2.like()

>>> comment3.like()

>>> comment3.dislike()

>>> comment3.dislike()

>>> comment3.dislike()

>>> comment4.like()

>>> comment4.like()

>>> comment4.like()

>>> comment4.like()

>>> comment4.dislike()

>>> comment4.dislike()

>>> comment4.dislike()

>>> post1.like()

>>> post1.like()

>>> post1.like()

>>> post1.dislike()

>>> post2.like()

>>> post2.like()

>>> post2.dislike()

>>> post3.dislike()

>>> post3.dislike()

>>> post3.like()

Проверка рейтинга командой

print(f"Рейтинг post/comment: {post/comment rating}")

1. ‘Обновить рейтинги пользователей.

>>> author1.update\_rating()

Проверка

>>> print({author1.rating})

>>> author2.update\_rating()

1. Вывести username и рейтинг лучшего пользователя (применяя сортировку и возвращая поля первого объекта).

>>> top\_user=Author.objects.order\_by('-rating').first()

>>> print(f'Лучший пользователь: {top\_user.author.username}, рейтинг: {top\_user.rating}')

1. Вывести дату добавления, username автора, рейтинг, заголовок и превью лучшей статьи, основываясь на лайках/дислайках к этой статье.

>>> top\_article=Post.objects.order\_by('-rating').first()

print(f'Дата публикации лучшей статьи:{top\_article.publication\_date}, автор и рейтинг лучшей статьи:{top\_article.author.author.username}, {top\_article.rating}. Статья “{top\_article.title}”. Превью: {top\_article.preview()}')

1. Вывести все комментарии (дата, пользователь, рейтинг, текст) к этой статье.

>>> for comment in top\_article.comment\_set.all():

... print(f"Дата: {comment.comment\_date}; Пользователь: {comment.user.username}; Рейтинг: {comment.rating};Текст: {comment.text}")