**Список всех команд, запускаемых в Django Shell**

from news.models import \*

1. **Создать двух пользователей (с помощью метода User.objects.create\_user).**

user01 = User.objects.create(username = 'Ivan')

user02 = User.objects.create(username = 'Anna')

user01 = User.objects.get(id=1)

user02 = User.objects.get(id=2)

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

author01=Author.objects.create(authorUser = user01)

author02=Author.objects.create(authorUser = user02)

Author.objects.get(id=1)

Author.objects.get(id=2)

1. **Добавить 4 категории в модель Category.**

category01 = Category.objects.create(category\_name='IT')

category02 = Category.objects.create(category\_name='Sport')

category03 = Category.objects.create(category\_name='Culture')

category04 = Category.objects.create(category\_name='Fashion')

category01 = Category.objects.get(id=1)

category02 = Category.objects.get(id=2)

category03 = Category.objects.get(id=3)

category04 = Category.objects.get(id=4)

1. **Добавить 2 статьи и 1 новость.**

article01 = Post.objects.create(author=author01, categoryType='AR', title='sometitle, text='somebigtext1')

article02 = Post.objects.create(author=author01, categoryType='AR', title='sometitle, text='somebigtext2')

news01 = Post.objects.create(author=author02, categoryType='NW', title='sometitle, text='somebigtext3')

article01 = Post.objects.get(id=1)

article02 = Post.objects.get(id=2)

news01 = Post.objects.get(id=3)

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

article01.postCategory.add(category01)

article01.postCategory.add(category02)

article02.postCategory.add(category03)

news01.postCategory.add(category03)

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

comment01 = Comment.objects.create(commentPost=article01, commentUser=user01, Text=’Commentstatistictext1')

comment02 = Comment.objects.create(commentPost=article02, commentUser=user02, Text=' Commentstatistictext2')

comment03 = Comment.objects.create(commentPost=article01, commentUser=user01, Text=' Commentstatistictext3')

comment04 = Comment.objects.create(commentPost=news01, commentUser=user02, Text=' Commentstatistictext4')

comment01 = Comment.objects.get(id=1)

comment02 = Comment.objects.get(id=2)

comment03 = Comment.objects.get(id=3)

comment04 = Comment.objects.get(id=4)

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

comment01.like()

comment02.like()

comment03.like()

comment04.like()

comment01.like() comment02.like() comment03.like() comment01.like() comment02.like() comment01.like()

comment01.dislike() comment04.dislike() comment03.dislike() comment01.dislike() article01.like() article01.like() article01.like() article01.like() article01.like() article02.like() article02.like() article02.like() news01.dislike()

1. **Обновить рейтинги пользователей.**

author01.update\_rating()

author02.update\_rating()

author01.authorRating

author02.authorRating

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

best = Author.objects.all().order\_by('-authorRating').values('authorUser', 'authorRating')[:1]

best

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

Post.objects.all().order\_by('-rating').values('dateCreation', 'postAuthor\_\_authorUser\_\_username', 'rating', 'preview\_name', 'text')[0]

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

Comment.objects.all().order\_by().values('dateCreation', 'commentUser\_\_username', 'commentPost', 'rating', 'commentText')[0]