**Список команд, запущенных в Django Shell**

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

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

*user1 = User.objects.create\_user('ivan.ivanov', 'ivanov@example.com', 'password1')*

*user2 = User.objects.create\_user('petr.petrov', 'petrov@example.com', 'password2')*

**2.** Создать два объекта модели *Author*, связанные с пользователями:

*from news.models import Author*

*author1 = Author.objects.create(user=user1)*

*author2 = Author.objects.create(user=user2)*

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

*from news.models import Category*

*cat1 = Category.objects.create(name='Technology')*

*cat2 = Category.objects.create(name='Science')*

*cat3 = Category.objects.create(name='Sport')*

*cat4 = Category.objects.create(name='Entertainment')*

**4.** Добавить 2 статьи и 1 новость:

*from news.models import Post*

*post1 = Post.objects.create(author=author1, post\_type='AR', title='Article 1', text='Text of article 1')*

*post2 = Post.objects.create(author=author2, post\_type='AR', title='Article 2', text='Text of article 2')*

*post3 = Post.objects.create(author=author1, post\_type='NW', title='News 1', text='Text of news 1')*

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

*post1.categories.add(cat1, cat2)*

*post2.categories.add(cat3)*

*post3.categories.add(cat2, cat4)*

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

*from news.models import Comment*

*comment1 = Comment.objects.create(post=post1, user=user1, text='Comment 1 on post 1')*

*comment2 = Comment.objects.create(post=post1, user=user2, text='Comment 2 on post 1')*

*comment3 = Comment.objects.create(post=post2, user=user1, text='Comment 1 on post 2')*

*comment4 = Comment.objects.create(post=post3, user=user2, text='Comment 1 on post 3')*

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

*post1.like()*

*post1.like()*

*post2.dislike()*

*post3.like()*

*comment1.like()*

*comment2.dislike()*

*comment3.like()*

*comment4.like()*

**8.** Обновить рейтинги пользователей:

*author1.update\_rating()*

*author2.update\_rating()*

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

*best\_author = Author.objects.all().order\_by('-rating').first()*

*print(f'Username: {best\_author.user.username}, Rating: {best\_author.rating}')*

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

*best\_post = Post.objects.all().order\_by('-rating').first()*

*if best\_post:*

*print(f"Date: {best\_post.creation\_date.strftime('%Y-%m-%d %H:%M:%S')}, "*

*f"Username: {best\_post.author.user.username}, "*

*f"Rating: {best\_post.rating}, "*

*f"Title: {best\_post.title}, "*

*f"Preview: {best\_post.preview()}")*

*else:*

*print("No posts available.")*

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

*comments = Comment.objects.filter(post=best\_post)*

*for comment in comments:*

*print(f"Date: {comment.creation\_date.strftime('%Y-%m-%d %H:%M:%S')}, "*

*f"User: {comment.user.username}, "*

*f"Rating: {comment.rating}, "*

*f"Text: {comment.text}")*