**Работа в Shell.**

from News.models import \*

* 1. Создать двух пользователей

User.objects.create(username = "Artem")

User.objects.create(username = "Anton")

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

ar = User.objects.all()[0]

an = User.objects.all()[1]

Author.objects.create(user\_com = ar)

Author.objects.create(user\_com = an)

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

Category.objects.create(category = 'политика')

Category.objects.create(category = 'спорт')

Category.objects.create(category = 'образование')

Category.objects.create(category = 'культура')

* 1. Добавить 2 статьи и 1 новость.

ar = Author.objects.all()[0]

an = Author.objects.all()[1]

Post.objects.create(author=ar) - новость

Post.objects.create(author=ar, news\_article=Post.article, title\_news='Политики расматривают закон об образовании', text\_news='Предметом регулирования настоящего Федерального закона являются общественные отношения, возникающие в сфере образования в связи с реализацией права на образование, обеспечением государственных гарантий прав и свобод человека в сфере образования и созданием условий для реализации права на образование (далее - отношения в сфере образования)')

Post.objects.create(author=an, news\_article=Post.article, title\_news='«Локомотив» объявил об уходе Керка', text\_news='«Локомотив» объявил об уходе полузащитника Джирано Керка. Игрок покинул расположение команды на сборах в Абу-Даби (ОАЭ). Стороны достигли соглашения, при котором игрок на условиях аренды будет играть за другую команду до лета 2023 года.')

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

post1 = Post.objects.all()[0]

post2 = Post.objects.all()[1]

post3 = Post.objects.all()[2]

post1.category\_com.add(1)

post2.category\_com.add(1)

post2.category\_com.add(3)

post3.category\_com.add(2)

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

Comment.objects.create(post\_com=Post.objects.get(pk=1), user\_com=User.objects.get(username='Anton'))

Comment.objects.create(post\_com=Post.objects.get(pk=1), user\_com=User.objects.get(username='Anton'), comment='Мир всему миру')

Comment.objects.create(post\_com=Post.objects.get(pk=2), user\_com=User.objects.get(username='Anton'), comment='Нужно вернуться к советской программе образования')

Comment.objects.create(post\_com=Post.objects.get(pk=3), user\_com=User.objects.get(username='Artem'), comment='Быстрее, выше, сильнее')

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

Post.objects.get(pk=1).like()

Post.objects.get(pk=2).dislike()

Post.objects.get(pk=3).like()

Comment.objects.get(pk=1).like()

Comment.objects.get(pk=2).dislike()

Comment.objects.get(pk=3).like()

Comment.objects.get(pk=4).dislike()

* 1. Обновить рейтинги пользователей.

ar = Author.objects.get(pk=1)

ar.update\_rating()

ar.rating

an = Author.objects.get(pk=2)

an.update\_rating()

an.rating

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

Author.objects.all().values('user\_com\_\_username','rating').order\_by('-rating').first()

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

Post.objects.all().values('time\_in', 'author\_\_user\_com\_\_username', 'rating', 'title\_news').order\_by('-rating').first()

Post.objects.all().order\_by('-rating').first().preview()

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

Comment.objects.all().values('time\_in', 'user\_com', 'rating', 'comment')