**Внимание: Писал я это в первую очередь для себя, поэтому есть в тексте комментарии(в shell они конечно не вводились)**

Работа в shell:

* + - * 1. Запуск

py manage.py shell

* + - * 1. Импорт моделей

from news.models import \*

* + - * 1. Выход ctrl+z

Если менялись методы таблиц, то надо выйти и зайти иначе будет работать старая версия метода. Если менялись данные не через shell, то тоже.

* + - * 1. Создать двух пользователей

u1 = User.objects.create\_user(username='Andy')

u2 = User.objects.create\_user(username='Bob')

u3 = User.objects.create\_user(username='Curt')

# просмотр

u1

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

Author.objects.create(user=u1)

Author.objects.create(user=u3)

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

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

Category.objects.create(name='IT')

Category.objects.create(name='Car')

Category.objects.create(name='TV')

* + - * 1. Добавить 2 статьи и 1 новость.

a = Author.objects.get(user\_id=1)

Post.objects.create(author=a, property='A', heading='Article 1', content='Первая статья')

b = Author.objects.get(user\_id=3)

Post.objects.create(author=b, property='A', heading='Article 2', content='Вторая статья')

Post.objects.create(author=a, property='N', heading='News', content='Новость')

Post.objects.create(author=b, heading='News 2', content='Новость другая')

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

Post.objects.get(id=1).category.add(Category.objects.get(id=2))

Post.objects.get(id=2).category.add(Category.objects.get(id=1))

Post.objects.get(id=2).category.add(Category.objects.get(id=4))

Post.objects.get(id=3).category.add(Category.objects.get(id=4))

Post.objects.get(id=4).category.add(Category.objects.get(id=1))

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

Comment.objects.create(post=Post.objects.get(id=1), user=Author.objects.get(user\_id=3).user, content='good')

Comment.objects.create(post=Post.objects.get(id=2), user=Author.objects.get(user\_id=1).user, content='bad')

Comment.objects.create(post=Post.objects.get(id=3), user=Author.objects.get(user\_id=3).user, content='best')

Comment.objects.create(post=Post.objects.get(id=4), user=Author.objects.get(user\_id=1).user, content='worst')

Comment.objects.create(post=Post.objects.get(id=2), user=User.objects.get(id=2), content="user's comment (not author)")

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

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

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

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

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

Comment.objects.get(id=5).dislike()

Comment.objects.get(id=5).dislike()

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

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

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

Post.objects.get(id=4).dislike()

* + - * 1. Обновить рейтинги пользователей. (имхо: правильнее было бы авторов)

a.update\_rating()

b.update\_rating()

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

# вариант 1

# q здесь список, возвращаю данные из первого элемента

q = Author.objects.order\_by('-rate')

str(q[0].rate)+ ' - '+q[0].user.username

# еще один вариант

# q1 = Author.objects.order\_by('-rate')[:1]

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

#Т.к статья была короткая (длина) я меняю содержание

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

tmp.content = 'Первая статья. После просмотра записи вебинара вам предоставляется возможность оценить его и поделиться своими впечатлениями/комментариями в [форме обратной связи](https://skillfactoryschool.typeform.com/to/E9YAIWSL#course=FPW&webinar=%D0%92%D0%B5%D0%B1%D0%B8%D0%BD%D0%B0%D1%80%20%C2%AB%D0%A0%D0%B0%D0%B7%D0%B1%D0%BE%D1%80%20%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0%20%D0%BF%D0%BE%20Django%C2%BB&link=https%3A//lms.skillfactory.ru/courses/course-v1%3ASkillFactory%2BFPW-2.0_D%2B2022/jump_to/block-v1%3ASkillFactory%2BFPW-2.0_D%2B2022%2Btype%40html%2Bblock%4026db96ec4a39432ea2848bef60e60547). Для нас это очень важно. Просим оставить детальный комментарий. Это поможет нам скорректировать/улучшить планы и форматы вебинаров, что позволит сделать ваш процесс обучения более эффективным и интересным.'

tmp.save(update\_fields=['content'])

#отбор только СТАТЬИ и сортировка по рейтингу

ar = Post.objects.filter(property='A').order\_by('-rate')

ar[0].time\_create

# в Post нет имени автора, поэтому схема post.author.user и имя поля

ar[0].author.user.username

ar[0].rate

ar[0].heading

ar[0].preview()

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

qc = Comment.objects.filter(post=ar[0])

for i in qc:

i.time\_create

i.user.username

i.rate

i.content

15) выход ctrl+z