**Проект NewsPostal**

Создаем директорий, где будет храниться проект

Создаем виртуальное окружение: python –m venv venv

И активируем его: venv\Scripts\activate

Устанавливаем django: pip install Django

Проверяем, что django действительно установлен: pip list

Создаем проект: django-admin startproject news

Переходим в созданный проект: cd news

Создаем приложение: python manage.py startapp newsapp

После работы с файлом models.py применяем миграции: python manage.py makemigrations и python manage.py migrate

Переходим в shell: python manage.py shell

Импортируем все данные из models.py: from newsapp.models import \*

**Задания:**

**1. Создать двух пользователей (с помощью метода**

**User.objects.create\_user('username')).**

**Выполнение:**

Создано 5 пользователей:

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

>>> u1

<User: Alex>

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

>>> u2

<User: Kim>

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

>>> u3

<User: Michael>

>>>u4=User.objects.create\_user(username='Chris')

>>> u4

<User: Chris>

>>>u5=User.objects.create\_user(username='Natalie')

>>> u5

<User: Natalie>

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

**Выполнение:**

>>> Author.objects.create(authorUser=u1)

<Author: Author object (1)>

>>> Author.objects.create(authorUser=u2)

<Author: Author object (2)>

>>> Author.objects.create(authorUser=u3)

<Author: Author object (3)>

>>> Author.objects.create(authorUser=u4)

<Author: Author object (4)>

>>> Author.objects.create(authorUser=u5)

<Author: Author object (5)>

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

**Выполнение:**

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

<Category: Category object (1)>

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

<Category: Category object (2)>

>>> Category.objects.create(name='Health)

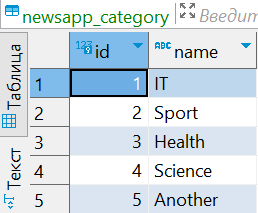
<Category: Category object (2)>

>>> Category.objects.create(name='Science')

<Category: Category object (2)>

>>> Category.objects.create(name='Another')

<Category: Category object (5)>



**4. Добавить 2 статьи и 1 новость.**

**Выполнение:**

***Статьи и новости первого автора(id=1):***

***>>> author=Author.objects.get(id=1)***

***>>> author***

***<Author: Author object (1)>***

>>> Post.objects.create(author=author, categoryType='NW', title='A new discovery in science', text='A scientist from the University and Technology of China has made an incredible discovery of a new species of butterfly')

<Post: Post object (1)>

>>> Post.objects.create(author=author, categoryType='AR', title='The US Open has a court where it always smells like marijuana. And the organizers can do nothing', text='At the US Open on Monday, the eighth racket of the world Maria Sakkari lost to Rebecca Masharova with a score of 4:6, 4:6. For the third time in a row, she was eliminated from the Grand Slam after the first match and said at a press conference that she might take a break from tennis - because she is tormented on the court.')

<Post: Post object (2)>

>>> Post.objects.create(author=author, categoryType='AR', title='Messi was stopped - in the 10th match for Inter. For the first time without victory and effective action', text="Messi spent the entire match on the field, but did not score or give an assist - for the first time in 10 games (or 88 days). During this time, the Argentine collected 11 + 3. Messi was not marked by effective actions - Inter's winning streak was also interrupted. Prior to that, the team won all nine matches with Leo, scoring 27 goals. Three of the nine wins were in penalty shootouts.")

<Post: Post object (3)>

>>> Post.objects.create(author=author, categoryType='AR', title='Apple is going to print cases for its devices on a 3D printer', text=''Apple has begun testing the use of 3D printers in the production of steel cases for its Apple Watch smartwatch. This was reported by Bloomberg, citing knowledgeable sources in the industry who wished to remain anonymous. 3D printing of cases should save manufacturers from the need to cut elements of the desired shape from a large piece of metal, reduce the time required to assemble the device and reduce waste.'')

<Post: Post object (4)>

>>> Post.objects.create(author=author, categoryType='NW', title='Indian rover "Pragyan" frolics on the moon', text=''The Indian Space Research Organization ISRO shared fresh news on how the Chandrayaan-3 the moon exploration mission is going. This time, ISRO shared a video showing the maneuvers of the moon rover Pragyan on the surface of the Earth's natural satellite. '')

<Post: Post object (5)>

***Статья четвертого автора(id=4):***

***>>> author=Author.objects.get(id=4)***

***>>> author***

***<Author: Author object (4)>***

>>> Post.objects.create(author=author, categoryType='AR', title='Fruit and Vegetable "Prescriptions" Tied to Food Security and Well-Being', text='Scientists say a greater possibility of avoiding stress and anxiety exists for Americans whose doctors advise consuming more fruits and vegetables through produce prescription. Through produce prescription programs, which permit doctors to prescribe fruits and vegetables alongside traditional medications and have those foods covered by a patient's health insurance or grants from a community program, researchers from Tufts University studied nearly 4,000 participants across a dozen states.')

<Post: Post object (6)>

***Новость второго автора(id=2):***

***>>> author=Author.objects.get(id=2)***

***>>> author***

***<Author: Author object (2)>***

>>> Post.objects.create(author=author, categoryType='NW', title=''Study Finds Links Between Children's Allergies and the Gut'', text=''Researchers found that childhood allergies, including eczema, hay fever, and food allergies, may develop because of imbalances in the gut biome during infancy. The gut biome is a complex collection of microorganisms that live in the digestive tract. Moreover, the gut microbiome begins to develop in infancy and can be influenced by many factors. These include diet, environmental exposures, and antibiotic use. Research suggests that gut microbiome disturbances may lead to neurodevelopmental disorders. For example, recent studies found altered gut biomes in children with attention deficit hyperactivity disorder (ADHD) and specific gut biome signatures in autistic individuals. In addition, other reports indicate that changes in an infant's gut microbiome may lead to childhood allergies.'')

<Post: Post object (7)>

***Новость третьего автора(id=3):***

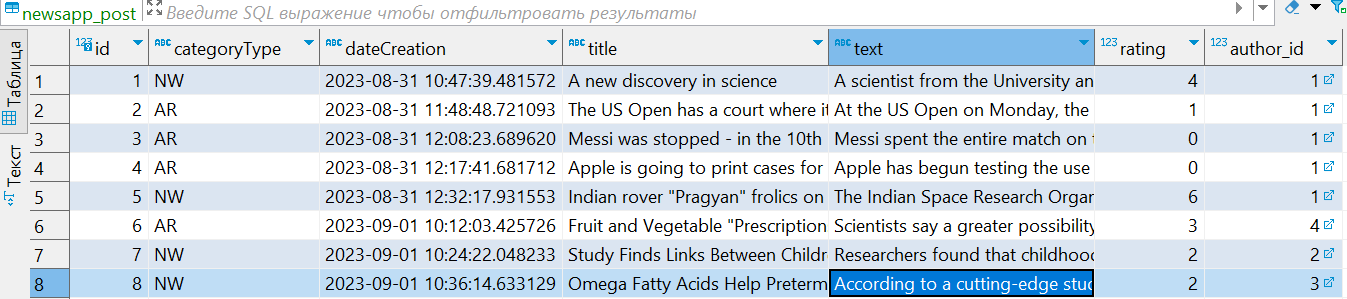
***>>> author=Author.objects.get(id=3)***

***>>> author***

***<Author: Author object (3)>***

>>> Post.objects.create(author=author, categoryType='NW', title='Omega Fatty Acids Help Preterm Babies With Vision', text='According to a cutting-edge study, preterm infants who receive an omega-3 and omega-6 fatty acid supplement before age two and a half have dramatically improved visual performance. The study, conducted between 2016 and 2019, featured 178 severely preterm infants who were delivered before the 28th week of pregnancy. The supplement improved how well the brain processed visual cues and lowered the risk of sight-threatening eye illness. By two and a half years old, preterm infants who received an omega-3 and omega-6 fatty acid supplements had a superior visual function. These positive results could affect upcoming recommendations for newborn care. ')

<Post: Post object (8)>



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

Примечание. Возможно здесь ошибка? И имеется в виду, что в каждой категории должно быть не менее двух статей/новостей, так как зачастую определенная статья/новость относится к определенной тематике (категории).

**Выполнение:**

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

>>> Post.objects.get(id=2).postCategory.add(Category.objects.get(id=2))

>>> Post.objects.get(id=3).postCategory.add(Category.objects.get(id=2))

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

>>> Post.objects.get(id=5).postCategory.add(Category.objects.get(id=1))

>>> Post.objects.get(id=5).postCategory.add(Category.objects.get(id=4))

>>> Post.objects.get(id=6).postCategory.add(Category.objects.get(id=3))

>>> Post.objects.get(id=6).postCategory.add(Category.objects.get(id=4))

>>> Post.objects.get(id=7).postCategory.add(Category.objects.get(id=3))

>>> Post.objects.get(id=7).postCategory.add(Category.objects.get(id=4))

>>> Post.objects.get(id=7).postCategory.add(Category.objects.get(id=3))

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

**Выполнение:**

>>>Comment.objects.create(commentPost=Post.objects.get(id=1), commentUser= Author.objects.get(id=1).authorUser, text='WOW')

<Comment: Comment object (1)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=1), commentUser= Author.objects.get(id=3).authorUser, text='What an amazing news!')

<Comment: Comment object (2)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=1), commentUser= Author.objects.get(id=2).authorUser, text='Ahh... China is ahead of the planet as usual...')

<Comment: Comment object (3)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=2), commentUser= Author.objects.get(id=1).authorUser, text=''OMG... It's US Open!'')

<Comment: Comment object (4)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=2), commentUser= Author.objects.get(id=2).authorUser, text=''What's happening to people...'')

<Comment: Comment object (5)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=2), commentUser= Author.objects.get(id=3).authorUser, text=''What did you expect? It's the USA!'')

<Comment: Comment object (6)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=2), commentUser= Author.objects.get(id=4).authorUser, text=''At least there is a good excuse for defeat'')

<Comment: Comment object (7)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=3), commentUser= Author.objects.get(id=1).authorUser, text=''Sometimes happens...'')

<Comment: Comment object (8)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=4), commentUser= Author.objects.get(id=3).authorUser, text=''Hm.. Cool idea, but it will work?'')

<Comment: Comment object (9)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=4), commentUser= Author.objects.get(id=4).authorUser, text=''They'll find a way.. Trust my intuition!!!'')

<Comment: Comment object (10)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=5), commentUser= Author.objects.get(id=2).authorUser, text='Who would have thought that India would do wonders in space?!')

<Comment: Comment object (11)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=5), commentUser= Author.objects.get(id=1).authorUser, text='Ahahahhaha')

<Comment: Comment object (12)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=6), commentUser= Author.objects.get(id=1).authorUser, text='Healthy food only!!!')

<Comment: Comment object (13)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=6), commentUser= Author.objects.get(id=1).authorUser, text='How about pizza and burgers?????!!!!!!')

<Comment: Comment object (14)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=7), commentUser= Author.objects.get(id=5).authorUser, text='Hate ma allergy((((')

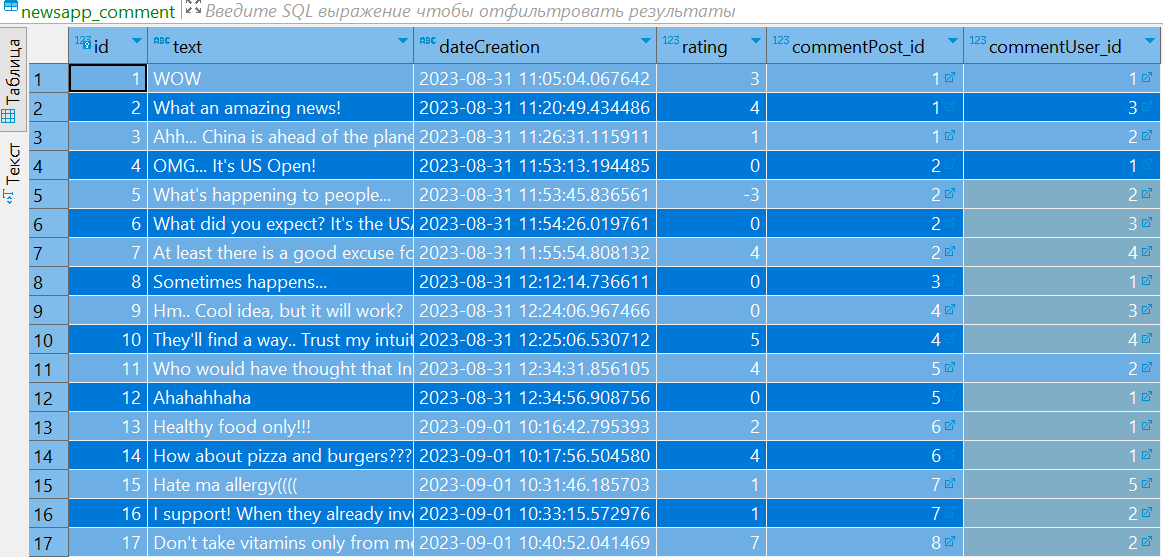
<Comment: Comment object (15)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=7), commentUser= Author.objects.get(id=2).authorUser, text='I support! When they already invent a 100% vaccine...')

<Comment: Comment object (16)>

>>>Comment.objects.create(commentPost=Post.objects.get(id=8), commentUser= Author.objects.get(id=2).authorUser, text=''Don't take vitamins only from medicals! Eat more fruits and vegetables! You'll be healthier!'')

<Comment: Comment object (17)>



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

**Выполнение:**

Рейтинг статей/новостей:

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

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

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

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

>>> Post.objects.get(id=2).like()

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

>>> Post.objects.get(id=1).dislike()

>>> Post.objects.get(id=1).rating

4

>>> Post.objects.get(id=2).like()

>>> Post.objects.get(id=2).rating

1

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

>>> Post.objects.get(id=3).dislike()

>>> Post.objects.get(id=3).rating

0

>>> Post.objects.get(id=4).like()

>>> Post.objects.get(id=4).like()

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

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

>>> Post.objects.get(id=4).rating

0

>>> Post.objects.get(id=5).like()

>>> Post.objects.get(id=5).like()

>>> Post.objects.get(id=5).like()

>>> Post.objects.get(id=5).like()

>>> Post.objects.get(id=5).like()

>>> Post.objects.get(id=5).like()

>>> Post.objects.get(id=5).rating

6

>>> Post.objects.get(id=6).like()

>>> Post.objects.get(id=6).like()

>>> Post.objects.get(id=6).like()

>>> Post.objects.get(id=6).rating

3

>>> Post.objects.get(id=7).like()

>>> Post.objects.get(id=7).like()

>>> Post.objects.get(id=7).dislike()

>>> Post.objects.get(id=7).like()

>>> Post.objects.get(id=7).rating

2

>>> Post.objects.get(id=8).like()

>>> Post.objects.get(id=8).like()

>>> Post.objects.get(id=8).like()

>>> Post.objects.get(id=8).like()

>>> Post.objects.get(id=8).dislike()

>>> Post.objects.get(id=8).dislike()

>>> Post.objects.get(id=8).rating

>>> Post.objects.get(id=8).rating

2

Рейтинг комментариев:

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

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

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

>>> Comment.objects.get(id=1).rating

3

>>> Comment.objects.get(id=2).like()

>>> Comment.objects.get(id=2).like()

>>> Comment.objects.get(id=2).like()

>>> Comment.objects.get(id=2).like()

>>> Comment.objects.get(id=2).rating

4

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

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

>>> Comment.objects.get(id=3).dislike()

>>> Comment.objects.get(id=3).rating

1

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

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

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

>>> Comment.objects.get(id=5).rating

-3

>>> Comment.objects.get(id=7).like()

>>> Comment.objects.get(id=7).like()

>>> Comment.objects.get(id=7).like()

>>> Comment.objects.get(id=7).dislike()

>>> Comment.objects.get(id=7).like()

>>> Comment.objects.get(id=7).like()

>>> Comment.objects.get(id=7).like()

>>> Comment.objects.get(id=7).dislike()

>>> Comment.objects.get(id=7).rating

4

>>> Comment.objects.get(id=10).like()

>>> Comment.objects.get(id=10).like()

>>> Comment.objects.get(id=10).like()

>>> Comment.objects.get(id=10).like()

>>> Comment.objects.get(id=10).like()

>>> Comment.objects.get(id=10).rating

5

>>> Comment.objects.get(id=11).like()

>>> Comment.objects.get(id=11).like()

>>> Comment.objects.get(id=11).like()

>>> Comment.objects.get(id=11).dislike()

>>> Comment.objects.get(id=11).like()

>>> Comment.objects.get(id=11).like()

>>> Comment.objects.get(id=11).rating

4

>>> Comment.objects.get(id=13).like()

>>> Comment.objects.get(id=13).like()

>>> Comment.objects.get(id=13).rating

2

>>> Comment.objects.get(id=14).like()

>>> Comment.objects.get(id=14).like()

>>> Comment.objects.get(id=14).like()

>>> Comment.objects.get(id=14).like()

>>> Comment.objects.get(id=14).rating

4

>>> Comment.objects.get(id=15).like()

>>> Comment.objects.get(id=15).rating

1

>>> Comment.objects.get(id=16).like()

>>> Comment.objects.get(id=16).rating

1

>>> Comment.objects.get(id=17).like()

>>> Comment.objects.get(id=17).like()

>>> Comment.objects.get(id=17).like()

>>> Comment.objects.get(id=17).like()

>>> Comment.objects.get(id=17).like()

>>> Comment.objects.get(id=17).like()

>>> Comment.objects.get(id=17).like()

>>> Comment.objects.get(id=17).rating

7

**8. Обновить рейтинги пользователей.**

**Выполнение:**

>>> Author.objects.get(id=1)

<Author: Author object (1)>

>>> a1=Author.objects.get(id=1)

>>> a1.update\_rating()

>>> a1.ratingAuthor

37

>>> Author.objects.get(id=2)

<Author: Author object (2)>

>>> a2=Author.objects.get(id=2)

>>> a2.update\_rating()

>>> a2.ratingAuthor

12

>>> Author.objects.get(id=3)

<Author: Author object (3)>

>>> a3=Author.objects.get(id=3)

>>> a3.update\_rating()

>>> a3.ratingAuthor

19

>>> Author.objects.get(id=4)

<Author: Author object (4)>

>>> a4=Author.objects.get(id=4)

>>> a4.update\_rating()

>>> a4.ratingAuthor

11

>>> Author.objects.get(id=5)

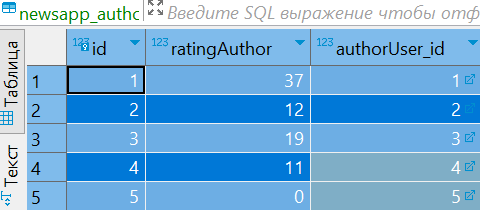
<Author: Author object (5)>

>>> a5=Author.objects.get(id=5)

>>> a5.update\_rating()

>>> a5.ratingAuthor

0



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

**Выполнение:**

a.values('authorUser\_\_username','ratingAuthor')

<QuerySet [{'authorUser\_\_username': 'Alex', 'ratingAuthor': 37}]>



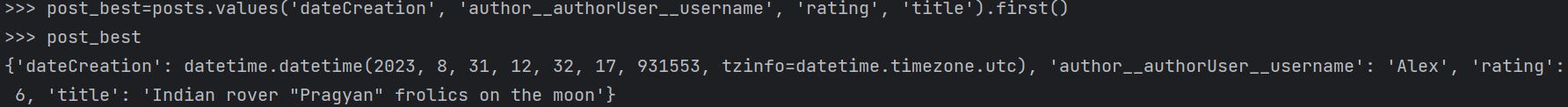
**10. Вывести дату добавления, username автора, рейтинг, заголовок и превью лучшей статьи/новости, основываясь на лайках/дислайках к этой статье/новости.**

**Выполнение:**

>>> post\_best = posts.values('dateCreation', 'author\_\_authorUser\_\_username', 'rating', 'title').first()

>>> post\_best

{'dateCreation': datetime.datetime(2023, 8, 31, 12, 32, 17, 931553, tzinfo=datetime.timezone.utc), 'author\_\_authorUser\_\_username': 'Alex', 'rating': 6, 'title': 'Indian rover "Pragyan" frolics on the moon'}





>>> posts.first().preview()

'The Indian Space Research Organization ISRO shared fresh news on how the Chandrayaan-3 the moon exploration mission is goin...'



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

**Выполнение:**

>>> post\_best = Post.objects.order\_by('-rating').first()

>>> post\_best

<Post: Post object (5)>

>>> post = Post.objects.get(id=5)

>>> post.comment\_set.all()

<QuerySet [<Comment: Comment object (11)>, <Comment: Comment object (12)>]>

>>> post.comment\_set.values('dateCreation', 'commentPost', 'commentUser', 'text', 'rating')

<QuerySet [{'dateCreation': datetime.datetime(2023, 8, 31, 12, 34, 31, 856105, tzinfo=datetime.timezone.utc), 'commentPost': 5, 'commentUser': 2, 't

ext': 'Who would have thought that India would do wonders in space?!', 'rating': 4}, {'dateCreation': datetime.datetime(2023, 8, 31, 12, 34, 56, 908756, tzinfo=datetime.timezone.utc), 'commentPost': 5, 'commentUser': 2, 'text': 'Ahahahhaha', 'rating': 0}]>

