(venv) C:\Users\artem\PycharmProjects\App\NewsPaper>py manage.py shell

Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from newapp.models import \*

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

>>> u1

<User: Petr>

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

Traceback (most recent call last):

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\utils.py", line 84, in \_execute

return self.cursor.execute(sql, params)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\sqlite3\base.py", line 423, in execute

return Database.Cursor.execute(self, query, params)

sqlite3.IntegrityError: UNIQUE constraint failed: auth\_user.username

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\contrib\auth\models.py", line 152, in create\_user

return self.\_create\_user(username, email, password, \*\*extra\_fields)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\contrib\auth\models.py", line 146, in \_create\_user

user.save(using=self.\_db)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\contrib\auth\base\_user.py", line 67, in save

super().save(\*args, \*\*kwargs)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\base.py", line 726, in save

self.save\_base(using=using, force\_insert=force\_insert,

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\base.py", line 763, in save\_base

updated = self.\_save\_table(

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\base.py", line 868, in \_save\_table

results = self.\_do\_insert(cls.\_base\_manager, using, fields, returning\_fields, raw)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\base.py", line 906, in \_do\_insert

return manager.\_insert(

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\manager.py", line 85, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\query.py", line 1270, in \_insert

return query.get\_compiler(using=using).execute\_sql(returning\_fields)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\sql\compiler.py", line 1416, in execute\_sql

cursor.execute(sql, params)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\utils.py", line 98, in execute

return super().execute(sql, params)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\utils.py", line 66, in execute

return self.\_execute\_with\_wrappers(sql, params, many=False, executor=self.\_execute)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\utils.py", line 75, in \_execute\_with\_wrappers

return executor(sql, params, many, context)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\utils.py", line 84, in \_execute

return self.cursor.execute(sql, params)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\utils.py", line 90, in \_\_exit\_\_

raise dj\_exc\_value.with\_traceback(traceback) from exc\_value

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\utils.py", line 84, in \_execute

return self.cursor.execute(sql, params)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\backends\sqlite3\base.py", line 423, in execute

return Database.Cursor.execute(self, query, params)

django.db.utils.IntegrityError: UNIQUE constraint failed: auth\_user.username

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

>>> u2

<User: Alex>

>>> u1

<User: Petr>

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

<Author: Author object (1)>

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

<Author: Author object (2)>

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

<Category: Category object (1)>

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

<Category: Category object (2)>

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

<Category: Category object (3)>

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

<Category: Category object (4)>

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

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'author' is not defined

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

>>> author

<Author: Author object (1)>

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

>>> author

<Author: Author object (2)>

>>> Post.objects.create(author=author, categoryType='NW', title='sometitle', text='somebigtext')

<Post: Post object (1)>

>>> Post.object.get(id=2)

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: type object 'Post' has no attribute 'object'

>>> Post.object.get(id=2)

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: type object 'Post' has no attribute 'object'

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

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\manager.py", line 85, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\query.py", line 435, in get

raise self.model.DoesNotExist(

newapp.models.Post.DoesNotExist: Post matching query does not exist.

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

'sometitle'

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

...

... )

>>> Post.objects.get(id=1).postCategory.add(Category.objects.get(id=3))

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

>>> Post.objects.create(author=author, categoryType='AR', title='funnytitle', text='bigtext')

<Post: Post object (2)>

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

... )

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

>>> Post.objects.create(author=author, categoryType='AR', title='smalltitle', text='anytext')

<Post: Post object (3)>

>>> Post.objects.get(id=3).postCategory.add(Category.objects.get(id=1))

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

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

<Comment: Comment object (1)>

>>> Comment.objects.create(commentPost=Post.objects.get(id=1), commentUser=Author.objects.get(id=2).authorUser, text='goodtext')

<Comment: Comment object (2)>

>>> Comment.objects.create(commentPost=Post.objects.get(id=2), commentUser=Author.objects.get(id=1).authorUser, text='notgoodtext')

<Comment: Comment object (3)>

>>> Comment.objects.create(commentPost=Post.objects.get(id=3), commentUser=Author.objects.get(id=2).authorUser, text='smiletext')

<Comment: Comment object (4)>

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

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

1

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

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

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

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

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

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

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

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

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

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

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

5

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

1

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

-3

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

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Aytgor' is not defined

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

Traceback (most recent call last):

File "<console>", line 1, in <module>

NameError: name 'Aythor' is not defined

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

<Author: Author object (1)>

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

>>> a.update

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: 'Author' object has no attribute 'update'

>>> a.update\_rating()

>>> a.ratengAuthor

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: 'Author' object has no attribute 'ratengAuthor'

>>> a.ratingAuthor

2

>>> Author.objects.get(id=2).update\_rating()

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

3

>>> Author.objects.get(id=3).update\_rating()

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\manager.py", line 85, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\query.py", line 435, in get

raise self.model.DoesNotExist(

newapp.models.Author.DoesNotExist: Author matching query does not exist.

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

2

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

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\manager.py", line 85, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\query.py", line 435, in get

raise self.model.DoesNotExist(

newapp.models.Post.DoesNotExist: Post matching query does not exist.

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

>>> Author.objects.get(id=3).update\_rating()

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\manager.py", line 85, in manager\_method

return getattr(self.get\_queryset(), name)(\*args, \*\*kwargs)

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\query.py", line 435, in get

raise self.model.DoesNotExist(

newapp.models.Author.DoesNotExist: Author matching query does not exist.

>>> Author.objects.get(id=2).update\_rating()

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

3

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

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

>>> Author.objects.get(id=2).update\_rating()

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

3

>>> Author.objects.get(id=1).update\_rating()

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

11

>>> a = Author.objects.order\_by('-ratingAuthor')[:1]

>>> a

<QuerySet [<Author: Author object (1)>]>

>>> a = Author.objects.order\_by('-ratingAuthor')[:-10]

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "C:\Users\artem\PycharmProjects\App\venv\lib\site-packages\django\db\models\query.py", line 294, in \_\_getitem\_\_

assert ((not isinstance(k, slice) and (k >= 0)) or

AssertionError: Negative indexing is not supported.

>>> a = Author.objects.order\_by('-ratingAuthor')[:0]

>>> a

<QuerySet []>

>>> a = Author.objects.order\_by('-ratingAuthor')[:1]

File "<console>", line 1

a = Author.objects.order\_by('-ratingAuthor')[:1]

IndentationError: unexpected indent

>>> a

<QuerySet []>

>>> a = Author.objects.order\_by('-ratingAuthor')[:1]

>>> a

<QuerySet [<Author: Author object (1)>]>

>>> a = Author.objects.order\_by('-ratingAuthor')

>>> a

<QuerySet [<Author: Author object (1)>, <Author: Author object (2)>]>

>>> for i in a:

... i.ratingAuthor

... i.authorUser.username

...

11

'Petr'

3

'Alex'