

Начало работы с БД

Django ORM

Object-Relational Mapping

ORM - технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования

Грубо говоря, помогает структуру питона перевести в sql язык для работы с БД. По сути не будет необходимости писать на sql

Ранее уже говорили о том, какие данные должны храниться в таблице карточек товаров

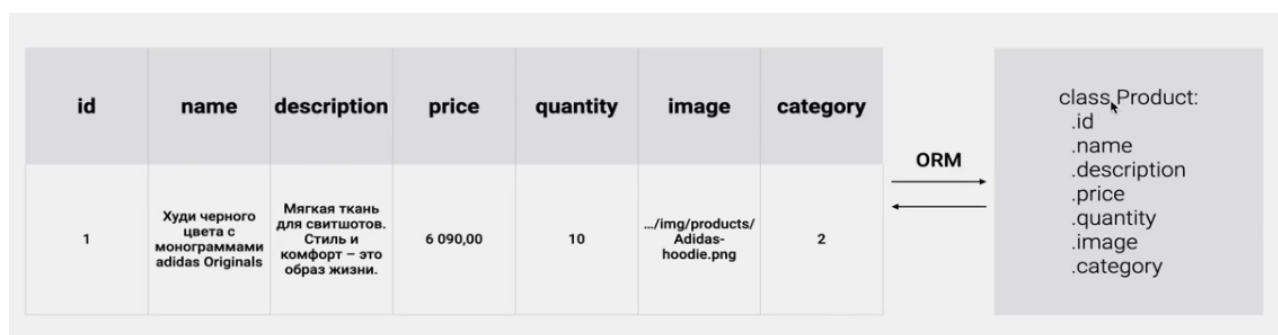
- id - primary key
- name - название товара
- description - описание товара
- price - цена товара
- quantity - кол-во товара
- image - путь до фото
- category - категория товара (связано с таблицей с категорий)

id	name	description	price	quantity	image	category
1	Худи черного цвета с монограммами adidas Originals	Мягкая ткань для свитшотов. Стиль и комфорт – это образ жизни.	6 090,00	10	.../img/products/Adidas-hoodie.png	2
2	Синяя куртка The North Face	Гладкая ткань. Водонепроницаемое покрытие. Легкий и теплый пуховый наполнитель.	23 725,00	7	.../img/products/Blue-jacket-The-North-Face.png	2

У каждого поля свой тип данных

Numeric (Int)	String (Varchar)	String (Text)	Numeric (Decimal)	Numeric (Int)	String (Varchar)	Numeric (Int)
id	name	description	price	quantity	image	category
1	Худи черного цвета с монограммами adidas Originals	Мягкая ткань для свитшотов. Стиль и комфорт – это образ жизни.	6 090,00	10	.../img/products/Adidas-hoodie.png	2

Будет создан класс Product с нужными переменными, который в дальнейшем поможет создать таблицу в БД



Создание таблицы БД

Очевидно необходимо избавиться от этого списка

```

def products(request):
    context = {
        'title': 'CandleFlame - Каталог',
        'products': [
            {
                'image': '/static/assets/img/products/ellie-ellien-HWqsW7FLWf4-unsplash.jpg',
                'name': 'Свечка-печка',
                'price': 6090,
                'description': 'Если хотите почувствовать себя как в сказке.'
            },
            {
                'image': '/static/assets/img/products/jocelyn-morales-KmL31BZp9qQ-unsplash.jpg',
                'name': 'Свечка-хлебопечка',
                'price': 6090,
                'description': 'Дома как в булочной.'
            },
            {
                'image': '/static/assets/img/products/stories-mFT0FzUKdL0-unsplash.jpg',
                'name': 'Свечка-сердечка',
                'price': 6090,
                'description': 'Для романтического настроения.'
            }
        ]
    }
    return render(request, 'products/products.html', context)

```

Тут только 3 товара и это было тяжело. А если 100? Необходимо перенести эти данные в таблицу БД, а перед этим создать ее.

Создание модели category

Для категорий будет меньше полей, поэтому с нее легче начать. Также в таблице товаров будет поле категория и нам будет, что связывать.

Какие поля нужны для этой таблицы:

- Название категории
- Описание категории
- id

Открываем файл models.py

<https://docs.djangoproject.com/en/4.2/topics/db/models/>

 Django • docs.djangoproject.com

Создаем класс `ProductCategory` и указываем, что он наследуется от встроенного в джанго класса с уже готовым набором функционала `models.Model`

```
3 # Create your models here.  
4 class ProductCategory(models.Model):  
5
```

Создаем первое поле `name` и с помощью пакета `models` задаем тип поля на понятном для sql языке. **Атрибут `max_length` обязателен для данного типа поля.** `Unique` - чтобы имена не повторялись

```
class ProductCategory(models.Model):  
    name = models.CharField(max_length=128, unique=True)
```

Следующее - поле описания категории. Здесь в качестве типа данных указан `textfield` - тот же текст, но неограничен по длине. *Поле может быть пустым благодаря указанным параметрам `null` и `blank` со значениями `True`*

```
class ProductCategory(models.Model):  
    name = models.CharField(max_length=128)  
    description = models.TextField(null=True, blank=True)
```

Все. Поле `id` автоматически предусмотрено в классе `Model`

Создание модели `products`

Объявляем класс

- поле `name` - ограничение по длине побольше + могут повторяться
- поле `description` - оставляем `textfield`, но оно должно быть обязательно заполнено (нет параметров `blank` и `null`)

- price - decimal, первый параметр - кол-во цифр до запятой (рандом), второй - после запятой
- quantity - PositiveIntegerField - целые числа больше нуля. Кол-во не может быть отрицательным, верно? В качестве атрибута значение по умолчанию.
- image - ImageField и в качестве атрибута папка, куда будут сохраняться изображения (чуть позже вернемся к этому)
- category - указываем foreignkey - это значит связь с другой таблицей и в скобках указываем нужную модель и метод удаления данных. Мы оставим каскад - удаляем категорию, значит удаляются все товары в ней.

```

10 class Product(models.Model):
11     name = models.CharField(max_length=256)
12     description = models.TextField()
13     price = models.DecimalField(max_digits=6, decimal_places=2)
14     quantity = models.PositiveIntegerField(default=0)
15     image = models.ImageField(upload_to='products_images')
16     category = models.ForeignKey(to=ProductCategory, on_delete=models.CASCADE)
17

```

Миграция

Миграции - перенос структуры модели на структуру БД

Команды:

- makemigrations - создание новых миграций
- migrate - применение миграций

Небольшое отступление. Откройте командную строку (или запустите сервер, если не запущен). Там будет ошибка о том, что не установлен нужный пакет.

```

ERRORS:
products.Product.image: (fields.E210) Cannot use ImageField because Pillow is not installed.
HINT: Get Pillow at https://pypi.org/project/Pillow/ or run command "python -m pip ins

```

Команда для установки

```
(venv) C:\Users\svtti\Desktop\store-server\store-server\store>python -m pip install Pillow
Collecting Pillow
```

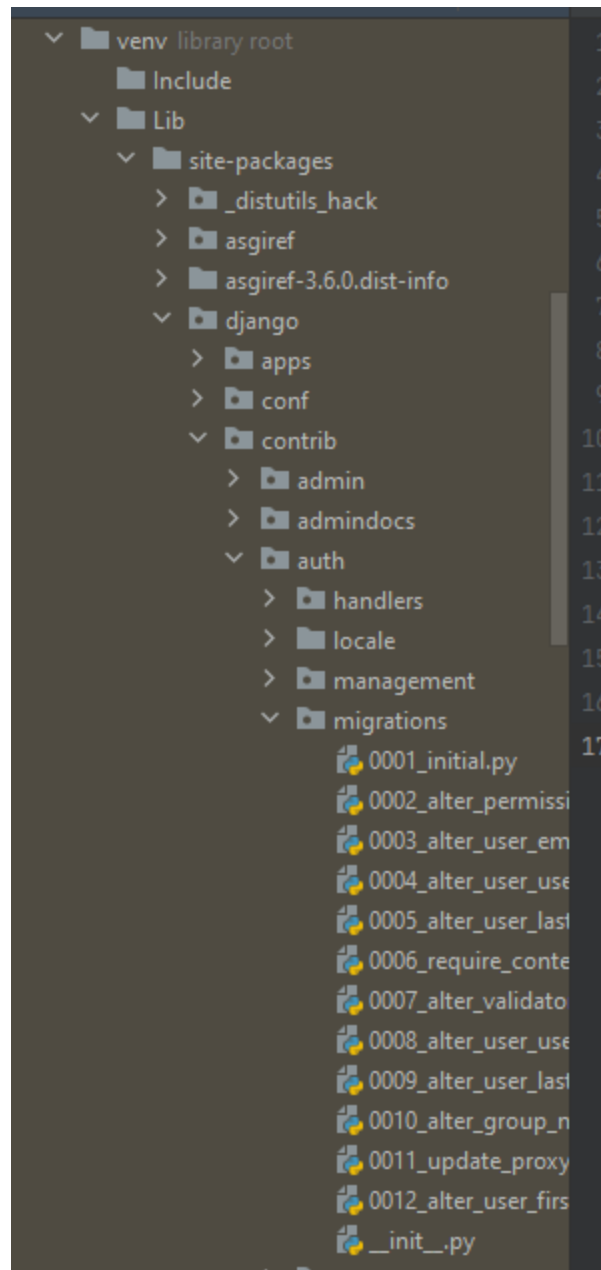
После успешной установке перезапустите сервер, не должно быть ошибок.
Зато

```
You have 18 unapplied migration(s). Your project may not work properly u
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 16, 2023 - 00:49:22
Django version 4.2.1, using settings 'store.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

18 непринятых миграций и нам советуют их применить. Но применение миграций это второй этап. А первый мы не делали, т.е. не создавали миграции. Но дальше в коде написано, какие приложения просят миграции

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
```

Джанго по умолчанию идет с этими приложениями. Но если открыть папку migration папки auth, то можно увидеть 12 файлов требующих миграцию. Это мы сделаем потом, вместе с нашими файлами.



В командной строке

```
(venv) C:\Users\svtti\Desktop\store-server\store-server\store>python manage.py makemigrations
Migrations for 'products':
  products\migrations\0001_initial.py
    - Create model ProductCategory
    - Create model Product
```

Нам сообщили, что создалась миграция с названием 0001_initial.py
Так выглядит сам файл

```
01_migrapy
views.py × models.py × 0001_initial.py × products.html × index.html ×
8
9 initial = True
10
11 dependencies = [
12 ]
13
14 operations = [
15     migrations.CreateModel(
16         name='ProductCategory',
17         fields=[
18             ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
19             ('name', models.CharField(max_length=128, unique=True)),
20             ('description', models.TextField(blank=True, null=True)),
21         ],
22     ),
23     migrations.CreateModel(
24         name='Product',
25         fields=[
26             ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
27             ('name', models.CharField(max_length=256)),
28             ('description', models.TextField()),
29             ('price', models.DecimalField(decimal_places=2, max_digits=6)),
30             ('quantity', models.PositiveIntegerField(default=0)),
31             ('image', models.ImageField(upload_to='products_images')),
32             ('category', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='productcategory')),
33         ],
34     ),
35 ]
```

Тут кстати появился id

Применяем миграции


```
(venv) C:\Users\svtti\Desktop\store-server\store-server\store>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, products, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying products.0001_initial... OK
  Applying sessions.0001_initial... OK
```

В нашей БД появились наши две таблички и остальные, которые требовали миграции

Имя	Тип	C
▼ Таблицы (13)		
> auth_group		C
> auth_group_permissions		C
> auth_permission		C
> auth_user		C
> auth_user_groups		C
> auth_user_user_permissions		C
> django_admin_log		C
> django_content_type		C
> django_migrations		C
> django_session		C
> products_product		C
> products_productcategory		C
> sqlite_sequence		C
▼ Индексы (16)		
> auth_group_permissions_group_id_b1...		C
> auth_group_permissions_group_id_pe...		C
> auth_group_permissions_permission_i...		C
> auth_permission_content_type_id_2f4...		C
> auth_permission_content_type_id_cod...		C
> auth_user_groups_group_id_97559544		C
> auth_user_groups_user_id_6a12ed8b		C
> auth_user_groups_user_id_group_id_...		C
> auth_user_user_permissions_permissi...		C