

МГТУ им. Н. Э. Баумана, кафедра ИУ5  
Курс «Разработка интернет-приложений»

Лабораторная работа № 5  
Работа с СУБД

Выполнила:  
Нурлыева Дана Джалилевна  
Группа ИУ5-53

Москва 2018

## Задание и порядок выполнения

В этой лабораторной работе вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также вам нужно будет дополнить свои классы предметной области, связав их с созданной базой. После этого вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей и ClassBasedViews.

Для сдачи вы должны иметь:

- 1) Скрипт с подключением к БД и несколькими запросами.
- 2) Набор классов вашей предметной области с привязкой к СУБД (класс должен уметь хотя бы получать нужные записи из БД и преобразовывать их в объекты этого класса)
- 3) Модели вашей предметной области
- 4) View для отображения списка ваших сущностей

## Выполнение работы

### 1) Обращение к БД из Python

ex1.py

```
import pymysql
pymysql.install_as_MySQLdb()
db = pymysql.connect(
    host="localhost",
    user="root",
    passwd="123",
    db="first_db"
)
# курсор
c = db.cursor()
# Вставка
#c.execute("INSERT INTO product (product_name, cost) VALUES (%s, %s)", ('пироженка', '75'))
#db.commit()
c.execute('SELECT * FROM product;')
entries = c.fetchall()
for e in entries:
    print(e)
c.close()
db.close()
```

Результат работы:

```
/Users/user/.conda/envs/untitled/bin/python /Users/user/PycharmProjects/lab5_1/my_app/ex1.py
(1, 'макароны', 80)
(2, 'хлеб', 25)
(3, 'мороженка', 50)
(4, 'пироженка', 75)

Process finished with exit code 0
```

## 2) Написание классов предметной области с соединением с БД ex2.py

```
import pymysql
pymysql.install_as_MySQLdb()
class Connection:
    def __init__(self, user, password, db, host='localhost'):
        #Параметры соединения
        self.user = user
        self.password = password
        self.db = db
        self.host = host
        self._connection = None
    @property
    def connection(self):
        return self._connection
    def __enter__(self):
        self.connect()
    def __exit__(self, exc_type, exc_val, exc_tb):
        self.disconnect()
    def connect(self):
        if not self._connection:
            self._connection = pymysql.connect(
                host=self.host,
                user=self.user,
                passwd=self.password,
                db=self.db)
    def disconnect(self):
        if self._connection:
            self._connection.close()
class Product:
    def __init__(self, db_connection, product_name, cost):
        self.db_connection = db_connection.connection
        self.product_name = product_name
        self.cost = cost
    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO product (product_name, cost) VALUES
(%s, %s)", (self.product_name, self.cost))
        self.db_connection.commit()
        c.close()
con = Connection("root", "123", "first_db")
with con:
    product = Product(con, 'Бананчик', '15')
    product.save()
```

Результат работы:

id_product	product_name	cost
1	макароны	80
2	хлеб	25
3	мороженка	50
4	пироженка	75
11	Бананчик	15

### 3) Django ORM

#### settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'first_db',
        'USER': 'root',
        'PASSWORD': '123',
        'HOST': 'localhost',
        'PORT': 3306, # Стандартный порт MySQL
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}
```

#### models.py

```
from django.db import models
class ProductsModel(models.Model):
    product_name = models.CharField(max_length=30)
    cost = models.CharField(max_length=6)
    def __unicode__(self):
        dict = {}
        dict['product_name'] = self.product_name
        dict['cost'] = self.cost
        return dict
```

#### migrations.py

```
# Generated by Django 2.1.3 on 2018-12-03 16:38
from django.db import migrations, models
class Migration(migrations.Migration):
    initial = True
    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Products',
            fields=[
                ('id', models.AutoField(auto_created=True,
primary_key=True, serialize=False, verbose_name='ID')),
```

```

        ('product_name', models.CharField(max_length=30)),
        ('cost', models.CharField(max_length=6)),
    ],
),
]

```

#### 4) Class Based Views

##### views.py

```

from django.shortcuts import render
from django.views import View
from my_app.models import ProductsModel
class ProductView(View):
    def get(self, request):
        # product = ProductsModel(product_name="Хлеб", cost="25")
        # product.save()
        products = ProductsModel.objects.all()
        data = {
            'products': products
        }
        return render(request, 'products.html', data)

```

Результат работы:

- Название товара - Мандарины  
Цена - 100 Руб.
- Название товара - Апельсины  
Цена - 50 Руб.
- Название товара - Помидор  
Цена - 94 Руб.
- Название товара - Хлеб  
Цена - 25 Руб.