

**Московский государственный технический университет  
им. Н.Э. Баумана.**

**Факультет «Информатика и управление»**

**Кафедра ИУ5. Курс «Разработка интернет приложений»**

**Отчет по лабораторной работе №5**

## **«Работа с СУБД»**

**Выполнил:**

студент группы ИУ5-52  
Афанасьев Л. Е.

**Проверил:**

преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Москва, 2017 г.

## 1 Задание

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

## 2 Листинг

### urls.py

```
from django.conf.urls import url, include
from lab5App.views import home, ProductView, ProductsView
urlpatterns = [
    url(r'^$', home),
    url(r'^product$', ProductsView.as_view(), name='products_url'), ]
```

### views.py

```
from django.shortcuts import render from
django.views.generic import View
def
home(request):
    par = {
        'header': 'Home'
    }
    return render(request, 'home.html', context=par)
class
CustomerView(ListView):
    model = Customer      template_name =
'customer_list.html'
class
ProductView(ListView):
    model = Product
    template_name = 'products.html'
context_object_name = 'products_list'
class OrderView(ListView):
    model = Order      template_name =
'order_list.html'
```

### models.html

```
from django.db import models
from django.contrib.auth.models import User
class
Customer(models.Model):
```

```

        user = models.OneToOneField(User, on_delete=models.CASCADE)
customer_name = models.CharField(max_length=30)        email =
models.EmailField(max_length=75)        first_name =
models.CharField(max_length=30)        last_name =
models.CharField(max_length=30)        birthday =
models.DateField()        sex = models.CharField(max_length=1)

class
Product(models.Model):
    product_name = models.CharField(max_length=30)
    description = models.CharField(max_length=255, null=True)
    price = models.FloatField(max_length=10)

    objects = models.Manager()

class
Order(models.Model):
    user = models.ForeignKey(Customer, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    order_date = models.DateField()

```

## products.html

```

{% extends 'base.html' %}

{% block title %}Фильмы{% endblock %}

{% block content %}
    <div class="jumbotron">
        <div class="container">
            <div class="page-header">
                <h1>Фильмы</h1>
            </div>
            <ul class="list-group">
                {% for product in products_list %}
                    <li>
                        <div class="main_block_elem">
                            <h4>{{ product.product_name }}</h4>
                            Описание: {{ product.description }}<br>
                            Цена: {{ product.price }}
                        </div>
                    </li>
                {% empty %}
                    <li>пустой список</li>
                {% endfor %}
            </ul>
        </div>
    </div>
{% endblock %}

```

## ex1.py

```
import MySQLdb

#! Открытие соединения с базой данных
db = MySQLdb.connect(
    host="localhost",
    user="dbuser",
    passwd="123",
    db="lab_db"
)
db.set_character_set('utf8')
#! Получить курсор для работы с базой данных
c=db.cursor()

#! Выполнить вставку
c.execute("insert into films (name, description) VALUES (%s, %s);",
    ('Криминальное чтиво', 'Квентин Тарантино, 1994'))
#! Фиксирование изменений
db.commit()

#! Выполнить выборку
c.execute("select * from films;")

#! Забрать все полученные записи
entries = c.fetchall()

#! Распечатать записи
for e in entries:
    print(e)

#! Закрытие курсора
c.close()
#! Закрытие соединения
db.close()
```

## ex2.py

```
import MySQLdb

class Connection:
    def __init__(self, user, password, db, host='localhost'):
        self.user = user
        self.host = host
        self.password = password
        self.db = db
        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()

    #! Открытие соединения
    def connect(self):
        if not self._connection:
            self._connection = MySQLdb.connect(
                host=self.host,
                user=self.user,
```

```

        passwd=self.password,
        db=self.db
    )
    self._connection.set_character_set('utf8')

    #! Закрытие соединения
    def disconnect(self):
        if self._connection:
            self._connection.close()

class films:

    def __init__(self, db_connection, product_name, description):
        self.db_connection = db_connection.connection
        self.product_name = product_name
        self.description = description

    def save(self):
        c = self.db_connection.cursor()
        c.execute("insert into films (name, description) values(%s, %s);",
                  (self.product_name, self.description))
        self.db_connection.commit()
        c.close()

conn = Connection("dbuser", "123", "lab_db")

with conn:
    product = films(conn, 'Тёмный рыцарь', 'Кристофер Нолан, 2008')
    product.save()

```

## Результат ex1.py, ex2.py

```

"C:\Program Files\Python36\python.exe" C:/Users/Leonid/Desktop/lab_5/lab6/ex1.py
(1, 'Список Шиндлера', 'Стивен Спилберг, 1993')
(7, 'Криминальное чтиво', 'Квентин Тарантино, 1994')
(8, 'Тёмный рыцарь', 'Кристофер Нолан, 2008')

```

## Django

# Home

## Фильмы

Список Шиндлера

2 отзыва

Криминальное чтиво