

**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»

Выполнила
Калугина Дарья

Проверил
Гапанюк Ю.Е.

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
from django.contrib import admin
from bd2 import views
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', views.OrdersView.as_view(), name='orders_url')
]
```

views.py

```
class OrdersView(ListView):
    model = Orders
    template_name = 'templ.html'
    context_object_name = 'orders_list'
```

models.html

```
from django.db import models

class Orders(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30)
    description = models.CharField(max_length=255)
```

base.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
```

```

    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">

</head>
<body>

{% block content %}{% endblock %}

</body>

</html>

```

temp.html

```

{% extends 'base.html' %}

{% block title %}Товары{% endblock %}

{% block content %}
    <div class="jumbotron">
        <div class="container">
            <div class="page-header">
                <h1>Товары</h1>
            </div>
            <ul class="list-group">
                {% for order in orders_list %}

                    <li>
                        <div class="main_block_elem">
                            <h4>{{ order.name }}</h4>
                            Описание: {{ order.description }}<br>

                        </div>
                    </li>
                    {% empty %}
                        <li>пустой список</li>
                    {% endfor %}
                </ul>
            </div>
        </div>
    {% endblock %}

```

ex1.py

```

import MySQLdb

db = MySQLdb.connect(
    host='localhost',
    user='dbuser',
    passwd='123',
    db='my_db'
)
db.set_character_set('utf8')
c = db.cursor()
c.execute('SET NAMES utf8;')
c.execute('SET CHARACTER SET utf8;')
c.execute('SET character_set_connection=utf8;')
c.execute("INSERT INTO orders (name, description) VALUES (%s, %s)", ('заказ 3', "какой то заказ 3"))

db.commit()

```

```
c.execute("SELECT * FROM orders")

entries = c.fetchall()

for e in entries:
    print(e)

c.close()
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 Orders:

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

    def save(self):
        c = self.db_connection.cursor()
        c.execute("insert into orders (name, description) values(%s, %s);",
            (self.zakaz, self.description))

        self.db_connection.commit()
```

```
c.close()

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

with conn:
    ord = Orders(conn, 'заказ 2 ', " какой то заказ 2")
    ord.save()
```

3. Результат

Товары

- Заказ 1

Описание: какой то заказ

- Заказ 2

Описание: какой то заказ 2

- Заказ 3

Описание: какой то заказ 3

- Заказ 4

Описание: какой то заказ 4

- Заказ 5

Описание: какой то заказ 5