

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

**Факультет «Информатика и управление»**

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

**Отчет по лабораторной работе №6**

**«Работа с СУБД»**

Выполнил:  
студент группы ИУ5-53  
Егоров С.А.

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
from labApp import views

urlpatterns = [
    url(r'^$', views.home),
    url(r'^departments$', views.DepartmentsView.as_view()),
]
```

### views.py

```
from django.shortcuts import render
from django.views.generic import ListView
from .models import *

def home(request):
    par = {
        'header': 'Home'
    }
    return render(request, 'home.html', context=par)

class DepartmentsView(ListView):
    model = Departments
    template_name = 'departments.html'
    context_object_name = 'departments_list'
```

### models.html

```
from django.db import models
from django.contrib.auth.models import User

class Departments(models.Model):
    name = models.CharField(max_length=30)
    description = models.CharField(max_length=255, null=True)
    objects = models.Manager()
```

## base.html

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>{% block title %}{% endblock%}</title>
<link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.css'
%}">
</head>
<body>
<navclass="navbar navbar-expand-smbg-primary navbar-dark">
<ulclass="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="/">Главная</a>
</li>
<li class="nav-item">
<a class="nav-link" href="/departments">Отделы</a>
</li>
</ul>
</nav>
<div>
    {% block content %}{% endblock%}
</div>
</body>
</html>
```

## home.html

```
{% extends 'base.html' %}

{% block title %}home{% endblock%}

{% block content %}
<h1>{{ header }}</h1>
{% endblock%}
```

## departments.html

```
{% extends 'base.html' %}

{% block title %}Отделы{% endblock%}

{% block content %}
<div class="jumbotron">
<div class="container">
<div class="page-header">
<h1>Отделы</h1>
</div>
<ulclass="list-group">
    {% for department in departments_list%}
<li>
<div class="main_block_elem">
<h4>{{ department.name }}</h4>
Описание: {{ department.description }}<br>
</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="3298",
    db="first_db"
)
db.set_character_set('utf8')
#! Получить курсор для работы с базой данных
c=db.cursor()

#! Выполнить вставку
c.execute("insert into departments (name, description) VALUES (%s, %s);",
    ('Отдел кадров', 'Этаж 4'))
#! Фиксирование изменений
db.commit()

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

#! 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()

#!/ Открытисоединения
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')

#!/ Закрытисоединения
def disconnect(self):
    if self._connection:
        self._connection.close()

class Department:

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

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

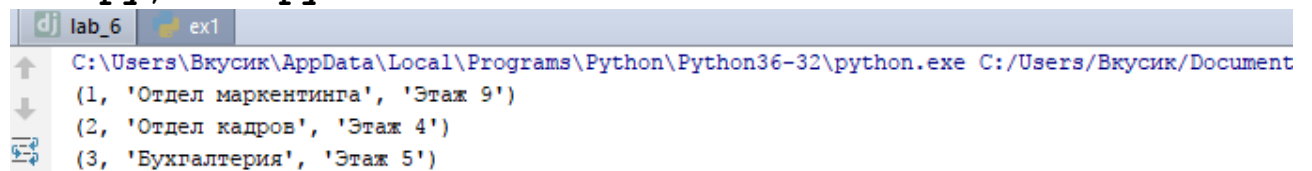
conn = Connection("dbuser", "3298", "first_db")

with conn:
    department = Department(conn, 'Бухгалтерия', "Этаж 5")
    department.save()

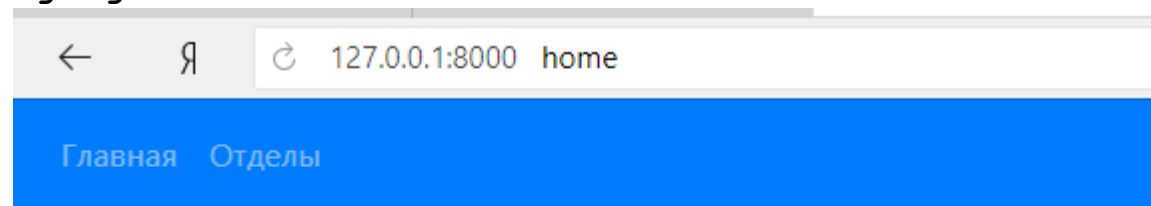
```

## Результат

ex1.py, ex2.py



Django



# Home

