

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

**Кафедра ИУ5 «Системы  
обработки информации и управления»**

**Отчет по лабораторной работе №6**

**«Работа с СУБД»**

**по дисциплине «Разработка Интернет-приложений»**

**Выполнил: студент  
группы ИУ5-53  
Иванников Александр**

**Москва, 2016**

## 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

## Исходный код

Файл `lab/models.py`

```
from django.db import models

class User2(models.Model):
    name = models.CharField(max_length=50)
    description = models.CharField(max_length=200)

class Airlines(models.Model):
    name = models.CharField(max_length=50)
    flight_id = models.IntegerField()
    description = models.CharField(max_length=200)
```

Файл `lab/views.py`

```
from django.shortcuts import render
from .models import User2, Airlines2
from django.http import HttpResponse, HttpResponseRedirect
from django.contrib import auth
from .forms import LoginForm, SignupForm

def index(request):
    airlines = Airlines2.objects.all()
    return render(request, "objects_list.html", {'airlines': airlines})

def flight(request, id):
    data = {
        'flight': {
            'id': id
        }
    }
    return render(request, "object.html", data)
```

Файл `lab6/urls.py`

```
from django.conf.urls import url
from django.contrib import admin
from my_app.views import index, flight,
```

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^index/', index, name = 'index'),
    url(r'^flight/(?P<id>\d+)', flight, name='flight_url'),
]
```

## Файл templates/layout.html

```
{% load static %}
<!DOCTYPE html>
<html lang="ru">
    <head>
        <title>{% block title %}{% endblock %}</title>
        <meta name="description" content="The project for IU5 course 'Web
App Dev'">
        <meta charset="utf-8">
        <link rel="stylesheet" type="text/css" href="{% static
"css/styles.css" %}">
    </head>
    <body>
        <header>
            <div class="site-name">Web Booking</div>
            <div class="user-menu">
                <a href=""></a>
                <ul>
                    <li><a>Профиль</a></li>
                    <li><a>Мои Заказы</a></li>
                    <li><a href="{% url 'login' %}" style="text-
decoration: none; color: black;">Выход</a></li>
                </ul>
            </div>
        </header>
        <div class="wrapper">
            {% block body %}{% endblock %}
        </div>
        <footer>
            {% block footer %}{% endblock %}
        </footer>
    </body>
</html>
```

## Файл templates/object\_list.html

```
{% extends 'layout.html' %}
{% block title %}Flight list{% endblock %}
{% block body %}
    {% load staticfiles %}
    <div class="add-container"><a href=""></a></div>
    <div class="object-list-container">
        {% for airline in airlines %}
            <div class="object-container">
                <a href="{% url 'flight_url' airline.flight_id %}">
                    <div class="object-info">{{ airline.description
}}</div>
                </a>
            </div>
        {% endfor %}
    </div>
```

```

    </div>
{% endblock %}
{% block footer %}
    <ul class="pagination">
        <li><a href="#">1</a></li>
        <li><a class="active" href="#">2</a></li>
        <li><a href="#">3</a></li>
    </ul>
{% endblock %}

```

## Файл templates/object.html

```

{% extends 'layout.html' %}
{% block title %}Flight {{ flight.id }}{% endblock %}
{% block body %}
    <div class="add-container">
        <a href="{% url 'index' %}"></a>
    </div>
    <div class="big-object-container">
        
        <div class="big-object-info">
            <div class="big-object-row">
                <div class="flight flight-number">SU{{ flight.id
}}</div>
                <div class="flight flight-way">
                    <div class="flight depart">
                        <div class="station">DME</div>
                        <div class="flight-time">14:00</div>
                    </div>
                    <div class="stops">
                        <span class="flight-duration">6 часов</span>
                        <ul class="stop-line">
                            <li class="stop-line"></li>
                        </ul>
                    </div>
                    <div class="flight arrive">
                        <div class="station">GOE</div>
                        <div class="flight-time">20:00</div>
                    </div>
                </div>
            </div>
            <div class="big-object-row">
                <div class="flight flight-
airlines">Авиакомпания</div>
                <div class="flight flight-airplane">Самолёт</div>
                <div class="flight flight-price">Стоимость</div>
            </div>
            <div class="big-object-row">
                <div class="flight flight-info-form flight-
airlines">
                    <div class="small-row">Siberia</div>
                    <div class="small-row" id="#airlines-
description">Siberia</div>
                </div>
                <div class="flight flight-info-form flight-
airplane">
                    <div class="small-row">Boeing 777-200</div>

```

```

<div class="small-row" id="#airplane-
description">safassaaafsfsfsfassaf afsasfas fasfa</div>
</div>
<div class="flight flight-info-form flight-
price"><div class="content">999999 P</div></div>
</div>
</div>
</div>
{% endblock %}
{% block footer %}
    <a href="objects_list.html"></a>
{% endblock %}
</body>
</html>

```

### Файл mysql-test.py

```

import MySQLdb

db = MySQLdb.connect(
    host="localhost",
    user="alex",
    passwd="123",
    db="first_db",
    charset="utf8"
)

cursor = db.cursor()
cursor.execute("INSERT INTO books (name, description) VALUES (%s, %s);",
('Книга', 'Описание книги'))

db.commit()

cursor.execute("SELECT * FROM Books")

books = cursor.fetchall()

print(books)

for book in books:
    print(book)

cursor.close()
db.close()

```

### Файл mysql-class-test.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()

def connect(self):
    if not self._connection:
        self._connection = MySQLdb.connect(
            host=self.host,
            user=self.user,
            passwd=self.password,
            db=self.db,
            charset = "utf8"
        )

def disconnect(self):
    if self._connection:
        self._connection.close()

class User:

    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 user (name, description) VALUES (%s, %s);",
            (self.name, self.description))
        self.db_connection.commit()
        c.close()

    def get(self):
        c = self.db_connection.cursor()
        c.execute("SELECT * FROM user;")
        users = []
        for row in c.fetchall():
            u = User
            u.name = row[1]
            u.description = row[2]
            users.append(u)
        return users

class Airlines:

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

    def save(self):
        c = self.db_connection.cursor()

```

```

        c.execute("INSERT INTO my_app_airlines2 (name, description,
flight_id) VALUES (%s, %s, %s);",
                    (self.name, self.description, self.flight_id))
        self.db_connection.commit()
        c.close()

    def get(self):
        c = self.db_connection.cursor()
        c.execute("SELECT * FROM my_app_airlines2;")
        users = []
        for row in c.fetchall():
            u = User
            u.name = row[1]
            u.description = row[2]
            users.append(u)
        return users

con = Connection("alex", "123", "first_db")

with con:
    user = User(con, 'Санек'.encode('utf-8'), 'Крутой чел'.encode('utf-8'))
    user.save()
    airlines = Airlines(con, 'Аэрофлот', "Российские авиалинии. Москва -
Казань", 6634)
    airlines2 = Airlines(con, 'S7', 'Сибирские авиалинии. Москва -
Ставрополь ', 5640)
    airlines3 = Airlines(con, 'ПитерАвиа', 'Северные авиалинии. Питер -
Казань ', 7009)
    airlines2.save()
    airlines3.save()

```

## Результат



