

Защищено:

Гапанюк Ю.Е.

"__" _____ 2016 г.

Демонстрация ЛР:

Гапанюк Ю.Е.

"__" _____ 2016 г.

Отчет по лабораторной работе №6
по курсу РИП

Вариант № <26>

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-52

(подпись)

Чекулина М.Ю.

"__" _____ 2016 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. Код

1. Обращение к БД из Python.

```
import MySQLdb

# Открываем соединение
db = MySQLdb.connect(
    host='localhost',
    user='Marina27',
    passwd='luck27',
    db='cars_db',
)

# без utf-8 не работает, что-то с кодировкой не то
db.set_character_set('utf8')
# Получаем курсор для работы с БД
c = db.cursor()
print("Вставить описание акции:")
# Выполняем вставку
c.execute('INSERT INTO lab_Sales(id, name, date, text) VALUES (%s, %s, %s, %s);',
          (4, 'Новая акция', 'с 15 декабря по 25 декабря', 'С наступающим Новым Годом!'))

# Фиксируем изменения
db.commit()

# Выполняем выборку
c.execute("SELECT * FROM lab_Sales;")

# Забираем все полученные записи
sales = c.fetchall()

# Печатаем их
for item in sales:
    print(item)

c.execute('DELETE FROM lab_Sales where id=4;')
db.commit()

c.execute("SELECT id, name, date, text FROM lab_Sales;")
print()
print("Результат после удаления акции с id=4:")
sales = c.fetchall()

# Печатаем их
for item in sales:
    print(item)
```

```
c.close() #Закрываем курсор
db.close() #Закрываем соединение
```

Результат работы:

```
C:\Python34\python.exe C:/Users/Марина/PycharmProjects/App/lab/first-db.py
Вставить описание акции:
(1, 'Зимние шины в подарок ', 'с 10 декабря 2016 по 15 февраля 2017', 'При покупке автомобиля от 1,5 млн. руб.')
(2, 'КАСКО без процентов', 'с 25 ноября 2016 по 20 декабря 2016', ' При покупке до 20 декабря 2016')
(3, 'Скидка 3% на автомобили', ' с 10 ноября 2016 ', ' Акция распространяется на модели 2016 года выпуска')
(4, 'Новая акция', 'с 15 декабря по 25 декабря', 'С наступающим Новым Годом!')

Результат после удаления акции с id=4:
(1, 'Зимние шины в подарок ', 'с 10 декабря 2016 по 15 февраля 2017', 'При покупке автомобиля от 1,5 млн. руб.')
(2, 'КАСКО без процентов', 'с 25 ноября 2016 по 20 декабря 2016', ' При покупке до 20 декабря 2016')
(3, 'Скидка 3% на автомобили', ' с 10 ноября 2016 ', ' Акция распространяется на модели 2016 года выпуска')

Process finished with exit code 0
```

2. Написание классов предметной области с соединением с БД

```
import MySQLdb
```

```
class Connection:
    def __init__(self, user, password, db, host="localhost", charset="utf8"):
        self.user = user
        self.host = host
        self.password = password
        self.db = db
        self.charset = charset
        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):
        #открытие соединения
        if not self._connection:
            self._connection = MySQLdb.connect(
                host=self.host,
                user=self.user,
                passwd=self.password,
                db=self.db,
                charset=self.charset
            )

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

class SalesClass:
    def __init__(self, db_connection, id, name, date, text):
        self.db_connection = db_connection.connection
        self.id = id
        self.name = name
        self.date = date
        self.text = text
```

```

def save(self):
    c = self.db_connection.cursor()
    c.execute("INSERT INTO lab_Sales(id, name, date, text) VALUES (%s, %s, %s, %s);", (self.id, self.name, self.date, self.text))
    self.db_connection.commit()
    c.close()

def print_all(self):
    c = self.db_connection.cursor()
    c.execute('SELECT id, name, date, text FROM lab_Sales;')
    items = c.fetchall()
    for item in items:
        print(item)
    c.close()
    return items

def del_str(self):
    c = self.db_connection.cursor()
    c.execute("DELETE FROM lab_Sales where id=5;")
    self.db_connection.commit()
    c.close()

con = Connection('Marina27', 'luck27', 'cars_db')

with con:
    new_sale = SalesClass(con, 5, 'Вторая новая акция', 'с 17 декабря по 25 декабря',
'S наступающим Новым Годом 2017!')
    new_sale.save()
    print("Добавили предложение:")
    list(new_sale.print_all())
    print()
    print("После удаления добавленного предложения:")
    new_sale.del_str()
    list(new_sale.print_all())

```

Результат работы:

C:\Python34\python.exe C:/Users/Марина/PycharmProjects/App/lab/classes_test.py

Добавили предложение:

```

(1, 'Зимние шины в подарок ', 'с 10 декабря 2016 по 15 февраля 2017', 'При покупке автомобиля от 1,5 млн. руб.')
(2, 'КАСКО без процентов', 'с 25 ноября 2016 по 20 декабря 2016', ' При покупке до 20 декабря 2016')
(3, 'Скидка 3% на автомобили', ' с 10 ноября 2016 ', ' Акция распространяется на модели 2016 года выпуска')
(5, 'Вторая новая акция', 'с 17 декабря по 25 декабря', 'С наступающим Новым Годом 2017!')

```

После удаления добавленного предложения:

```

(1, 'Зимние шины в подарок ', 'с 10 декабря 2016 по 15 февраля 2017', 'При покупке автомобиля от 1,5 млн. руб.')
(2, 'КАСКО без процентов', 'с 25 ноября 2016 по 20 декабря 2016', ' При покупке до 20 декабря 2016')
(3, 'Скидка 3% на автомобили', ' с 10 ноября 2016 ', ' Акция распространяется на модели 2016 года выпуска')

```

Process finished with `exit` code 0

3. Django ORM

models.py

```
from django.db import models
```

```
# Create your models here.
```

```

class Car(models.Model):
    name = models.CharField(max_length=255, verbose_name='Название модели автомобиля')
    price = models.FloatField(verbose_name='Цена')
    country = models.CharField(max_length=100, verbose_name='Страна-производитель')
    image_car = models.ImageField(upload_to='images', verbose_name='Изображение модели')
    length = models.CharField(max_length=20, verbose_name='Длина автомобиля')
    vehicle_clearance = models.CharField(max_length=15, verbose_name='Клиренс')

```

```

автомобиля')
    max_speed = models.CharField(max_length=15, verbose_name='Максимальная скорость
автомобиля')
    average_fuel_consumption = models.CharField(max_length=15, verbose_name='Средний
расход топлива в смешанном режиме')
    weight = models.CharField(max_length=15, verbose_name='Масса автомобиля')
    type_of_transmission = models.CharField(max_length=100, verbose_name='Коробка
передач')
    volume = models.CharField(max_length=20, verbose_name='Объем двигателя')
    description = models.CharField(max_length=1500, verbose_name='Описание')

class Address(models.Model):
    name = models.CharField(max_length=255, verbose_name='Название')
    adr = models.CharField(max_length=255, verbose_name='Адрес')
    time = models.CharField(max_length=255, verbose_name='Время работы')
    phone = models.CharField(max_length=255, verbose_name='Телефон')

class Sales(models.Model):
    name = models.CharField(max_length=255, verbose_name='Название')
    date = models.CharField(max_length=50, verbose_name='Сроки акции')
    text = models.CharField(max_length=255, verbose_name='Описание')

class User (models.Model):
    dop_id = models.IntegerField()
    first_name = models.CharField(max_length=255, verbose_name='Имя')
    last_name = models.CharField(max_length=255, verbose_name='Фамилия')
    age = models.IntegerField(verbose_name='Возраст')
    phone = models.CharField(max_length=20, verbose_name='Контактный телефон')
    email = models.EmailField(verbose_name='Электронная почта')

```

4. Class Based Views

lab\urls.py

```

from django.conf.urls import url
from . import views
from lab.views import CarView, AddressView, SalesView, UserView

```

```

urlpatterns = [
    url(r'^$', views.main, name='main'),
    url(r'^model_row$', CarView.as_view(), name='model_row'),
    url(r'^sales$', SalesView.as_view(), name='sales'),
    url(r'^users$', UserView.as_view(), name='users'),
    url(r'^address$', AddressView.as_view(), name='address'),
]

```

views.py

```

from django.shortcuts import render
from lab.models import Car, Address, Sales, User
from django.views.generic import View

# Create your views here.
# def base(request):
#     return render(request, 'lab/base.html', {})

def main(request):
    return render(request, 'lab/main.html', {})

class CarView(View):
    def get(self, request):
        model_row = Car.objects.all()

```

```

    return render(request, 'lab/model_row.html', {'model_row': model_row})

class AddressView(View):
    def get(self, request):
        address = Address.objects.all()
        return render(request, 'lab/address.html', {'address': address})

class SalesView(View):
    def get(self, request):
        sales = Sales.objects.all()
        return render(request, 'lab/sales.html', {'sales': sales})

class UserView(View):
    def get(self, request):
        users = User.objects.all()
        return render(request, 'lab/users.html', {'users': users})

```

base.html

```

{% load staticfiles %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}Nissan Россия{% endblock %}</title>
    <link rel="stylesheet" type="text/css" href="{% static 'lab/css/bootstrap.css' %}">
    {# <link rel="stylesheet" type="text/css" href="/static/css/bootstrap-theme.css">#}
    {# <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">#}
    <link rel="stylesheet" href="{% static 'lab/css/style.css' %}">
    {# <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">#}
    {# <link
href="https://fonts.googleapis.com/css?family=Lobster&subset=latin,cyrillic"
rel="stylesheet" type="text/css">#}

</head>

<body>

    <div class="content container">
        <div class="page-header">
            <h1 class="rows_main"><a href="/"> Автомобили Nissan</a></h1>
        </div>
        <div class="row">
            <div class="col-md-8">
                {% block content %}
                {% endblock %}
            </div>
        </div>
    </div>

    <div class="container">
        <header class="ttl"> {% block header %} {% endblock %}</header>
        <div>
            {% block body %}

                <style>

body {
background:url({% static 'lab/images/logotip.png' %} ) no-repeat center center

```

```

fixed;
    -webkit-background-size: cover;
    -moz-background-size: cover;
    -o-background-size: cover;
    background-size: cover;
}
h3{
    font-family: 'Lobster';
    font-size:xx-large;
}
</style>
    {% endblock %}
</div>
<div>

    <div class="container">
        <div class="footer">
            <p>&copy; Чекулина Марина, 2016</p>
        </div>
    </div>

</body>
</html>

```

main.html

```

{% extends 'lab/base.html' %}
{% block title %} Главная страница {% endblock %}
{% block content %}
    <br>
    <ul>
        <li>
            <a href="{% url 'model_row' %}" class="rows">
                Модельный ряд
            </a>
        </li>
        <li>
            <a href="{% url 'sales' %}" class="rows">
                Выгодные предложения
            </a>
        </li>
        <li>
            <a href="{% url 'address' %}" class="rows">
                Официальные дилеры
            </a>
        </li>
        <li>
            <a href="{% url 'users' %}" class="rows">
                Пользователи
            </a>
        </li>
    </ul>
</br>

{% endblock %}

```

address.html

```

{% extends 'lab/base.html' %}
{% load staticfiles %}
{% block title %}Официальные дилеры{% endblock %}

{% block header %}Официальные дилеры{% endblock %}

{% block body %}

    <style type="text/css">
body {

```

```
{# background: url({% static 'lab/images/engine.png' %}) lightgray; /* Добавляем фон */#}
background:lightgray;
background-position: right bottom; /* Положение фона */
background-repeat: repeat-y; /* Повторяем фон по горизонтали */
}
</style>
<ul>
{% for adr in address %}

    {% include 'lab/address_item.html' %}

{% endfor %}
</ul>

<input type="button" value="Назад" onclick="history.back()" >
{% endblock %}
```

address_item.html

```
{% load staticfiles %}

{% block body %}

<ul>
<li>

    <p class="adr">{{ adr.name }}</p>
    <p>Адрес:{{ adr.adr }}</p>
    <p>Время работы:{{ adr.time }}</p>
    <p>Телефон:{{ adr.phone }}</p>
    <p>Схема проезда:</p>
</li>
</ul>

{% endblock %}
```

model_row.html

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

{% block title %}Модельный ряд {% endblock %}

{% block header %}Модельный ряд{% endblock %}

{% block body %}
    <style type="text/css">
    div {
        background: powderblue;
    }
    </style>

    <ul>
    {% for car in model_row %}
        {% include 'lab/model_row_item.html' %}

    {% endfor %}
    </ul>
{% endblock %}
```

model_row_item.html

```
{% load staticfiles %}
<div>
<li>

    <p class="rows">Модель: {{ car.name }}</p>
```



```


<div class="card-block">
    <p>Уникальный номер автомобиля:{{ car.id }}</p>
    <p>Стоимость: от {{ car.price }} {{ "рублей" }}</p>
    <p>Страна-производитель:{{ car.country }}</p>
    <p>Описание:{{ car.description }}</p>
    <p>Габариты атомобиля (Д*Ш*Ш):{{ car.length }}</p>
    <p>Максимальная скорость:{{ car.max_speed }}</p>

</div>
</li>
</div>

```

sales.html

```

{% extends 'lab/base.html' %}
{% load staticfiles %}
{% block title %}Выгодные предложения{% endblock %}

{% block header %}Выгодные предложения{% endblock %}

{% block body %}
<style type="text/css">
body {
    background: url({% static 'lab/images/sales.png' %}) 100% 100% no-repeat; /*
Добавляем фон */
    background-size: cover; /* Масштабируем фон */
}
</style>

<ul>
    {% for model in sales %}
        <li>
            <p class="sal">{{ model.name }}</p>
            <p>Срок проведения: {{ model.date }}</p>
            <p>Комментарий:{{ model.text }}</p>
            {#
                #}
            </li>
        {% endfor %}
    </ul>
<input type="button" value="Назад" onclick="history.back()">
{% endblock %}

```

users.html

```

{% extends 'lab/base.html' %}
{% load staticfiles %}
{% block title %}Пользователи{% endblock %}

{% block header %}Пользователи{% endblock %}

{% block body %}
<style type="text/css">
body {
    background: url({% static 'lab/images/sales.png' %}) 100% 100% no-repeat; /*
Добавляем фон */
    background-size: cover; /* Масштабируем фон */
}
</style>

<ul>
    {% for model in users %}
        <li>
            <p class="sal">{{ model.first_name }}{{ " " }}{{ model.last_name }}</p>
            {#
                <p>Фамилия: {{ model.last_name }}</p>#}

```

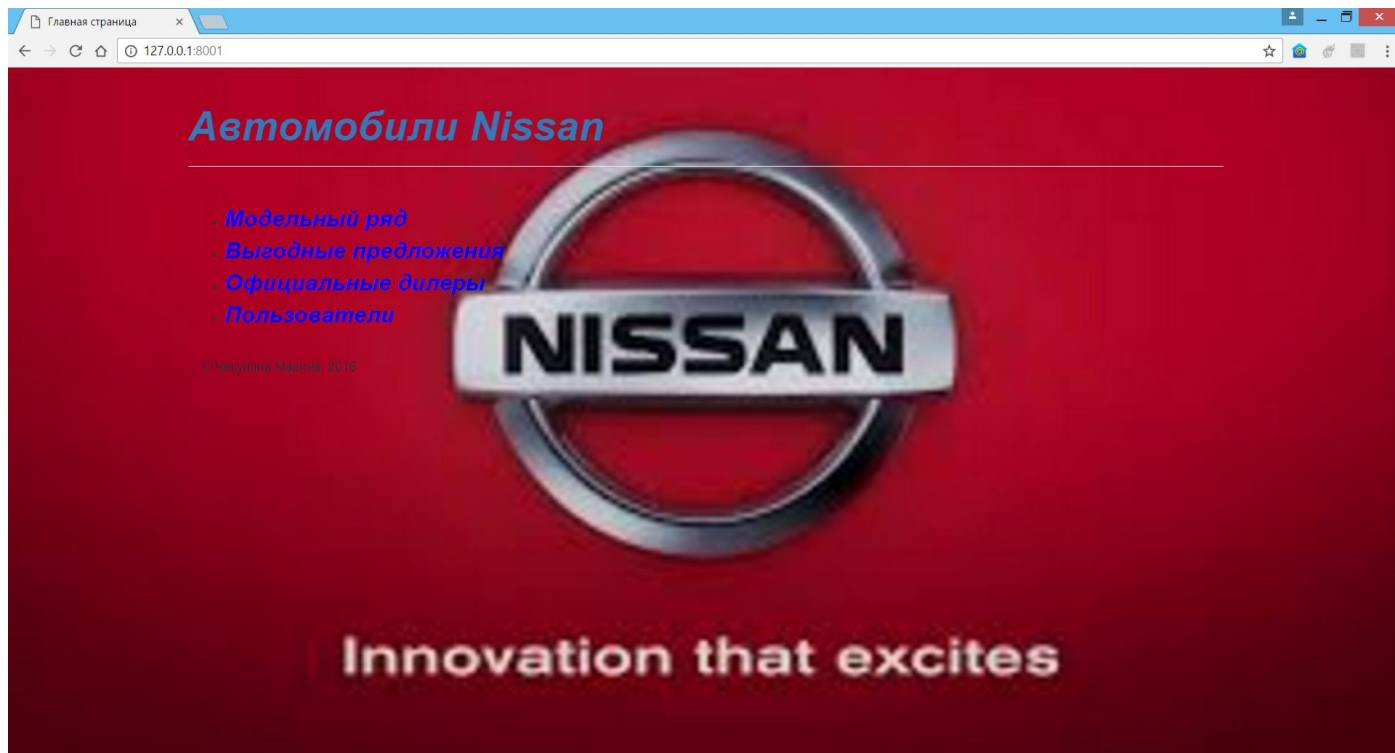
```

<p>Возраст: {{ model.age }}</p>
<p>Контактный телефон: {{ model.phone }}</p>
<p>Электронная почта: {{ model.email }}</p>
</li>
{% endfor %}
</ul>
<input type="button" value="Назад" onclick="history.back()" ">
{% endblock %}

```

3. Результаты работы

Главная страница:



Возможные переходы из главного меню:

