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

## Разработка интернет-приложений Лабораторная работа № 6

"Работа с СУБД"

Выполнил: студент группы ИУ5-53 Степанов Д. Г. Подпись: Дата:

### Задание и порядок выполнения

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.
- 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
- View для отображения списка ваших сущностей

#### Файл dbclass.py

import MySQLdb

```
class Connection:
    def init (self, user, passwd, db, host='localhost'):
       self.host = host
       self.user = user
       self.passwd = passwd
       self.db = db
       self.use unicode = True
       self.charset = "utf8"
       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.passwd,
               db=self.db,
               use_unicode=self.use_unicode,
               charset=self.charset
    def disconnect(self):
       if self._connection:
           self. connection.close()
```

```
class Computer:
    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 `my_app_computer` (`name`, `description`) VALUES (%s, %s);", (self.name, self.description))
        self.db_connection.commit()
        c.close()

con = Connection('dbuser', '12345', 'first_db', 'localhost')
with con:
        computer = Computer(con, 'Компутер', 'Описание компутера')
        computer.save()
```

#### Файл dbconn.py

```
import MySQLdb
db = MySQLdb.connect(
    host='localhost',
    user='dbuser',
passwd='12345',
     db='first_db',
    use_unicode=True,
     charset='utf8'
c = db.cursor()
#c.execute("INSERT INTO computer (name, description) VALUES (%s, %s);", ('компутер', 'описание'))
db.commit()
c.execute("SELECT * FROM my_app_computer;")
entries = c.fetchall()
for el in entries:
    print(el)
c.close()
db.close()
```

```
from django.db import models
class ComputerModel(models.Model):
   class Meta:
       db_table = 'my_app_computer'
   name = models.CharField(max_length=30)
   description = models.CharField(max_length=255)
   def __str__(self):
      return "'name':{}, 'description':{}".format(self.name, self.description)
class CustomerModel(models.Model):
   class Meta:
       db_table = 'my_app_customer'
    name = models.CharField(max_length=64, default='')
   email = models.CharField(max_length=64)
   password = models.CharField(max_length=64)
    computers = models.ManyToManyField(ComputerModel, through='OrderModel')
   def __str__(self):
        return "'name':{}, 'email':{}, 'password':{}".format(self.name, self.email, self.password)
class OrderModel(models.Model):
   class Meta:
       db_table = 'my_app_order'
    customer = models.ForeignKey(CustomerModel, on_delete=models.CASCADE)
   computer = models.ForeignKey(ComputerModel, on_delete=models.CASCADE)
    date_received = models.DateTimeField()
   date_completed = models.DateTimeField()
```

## Результат выполнения программы

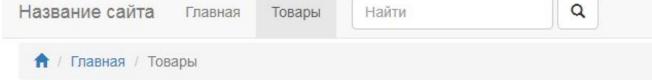
dbconn

C:\Users\0\AppData\Local\Programs\Python\Python36-32\python.exe C:/GIT/python\_labs/lab6/dbconn.py
(1, 'Asus n550jk', 'Asus n550jk', 15,6 1920x1080 ips display, intel core i5-4600 2,8GHz, 8Gb RAM, 128Gb SSD, wifi 2,4GHz bgn+ac, bluetooth 4.
(2, 'Компутер 1', 'Описание компутера 1')
(3, 'Asus n751jv', '17,3 1920x1080 ips display, intel core i7-4800 3,2GHz, 16Gb RAM, 256Gb SSD, wifi 2,4GHz bgn+ac, bluetooth 4.0')

Ргосезя finished with exit code 0

Название сайта главная Товары Найти

Q



- Asus n550jk
- Компутер 1
- Asus n751jv

Название сайта Главная Товары Найти Q

↑ Главная / Главная / Asus n550jk

Unit number: 1

Unit name: Asus n550jk

Unit description: Asus n550jk, 15,6 1920x1080 ips display, intel core i5-

4600 2,8GHz, 8Gb RAM, 128Gb SSD, wifi 2,4GHz

bgn+ac, bluetooth 4.0