

Лабораторная работа №6.

Тема: "Работа с СУБД"

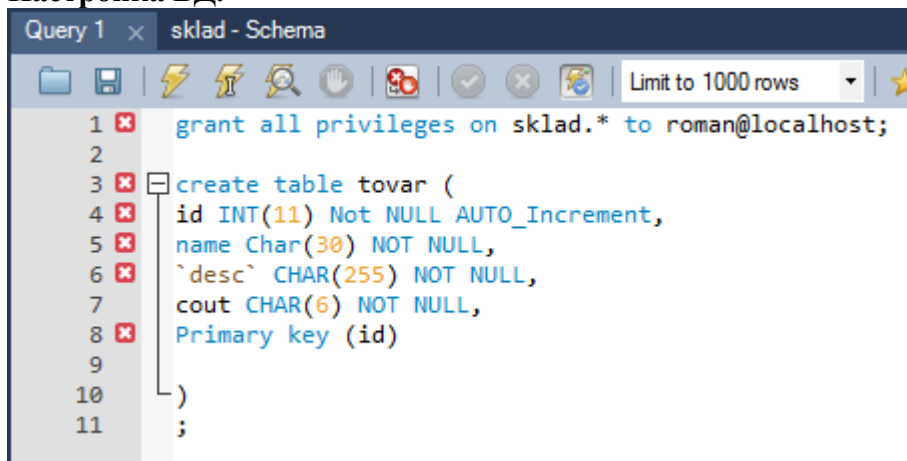
Студент: Щипицин Р.А.,
группа ИУ5-51.

Описание задания лабораторной работы

В этой лабораторной работе вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также 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.

Реализация.

Настройка БД.



```
1 grant all privileges on sklad.* to roman@localhost;
2
3 create table tovar (
4 id INT(11) Not NULL AUTO_Increment,
5 name Char(30) NOT NULL,
6 `desc` CHAR(255) NOT NULL,
7 cout CHAR(6) NOT NULL,
8 Primary key (id)
9
10 )
11 ;
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'django',
        'USER': 'dbuser',
        'PASSWORD': '123',
        'HOST': 'localhost',
        'PORT': '3306',
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8'
    }
}
```

Класс подключения к БД без django.

```

class Connection:
    def __init__(self, user, passwd, db, host='localhost'):
        self.user = user
        self.passwd = passwd
        self.db = db
        self.host = host
        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
            )

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

```

Класс Товары.

```
class Товар:
    def __init__(self, db_connection, name, desc, cout):
        self.db_connection = db_connection.connection
        self.name = name
        self.desc = desc
        self.cout = cout

    def save(self):
        c = self.db_connection.cursor()
        c.execute("INSERT INTO tovar (`name`, `desc`, cout) values (%s, %s, %s)",
                  (self.name, self.desc, self.cout))
        self.db_connection.commit()
        c.close()
```

Класс товары в django models

```
from django.db import models

class ТоварModel(models.Model):
    name = models.CharField(max_length=70)
    desc = models.CharField(max_length=300)
    cout = models.IntegerField()
```

Класс отображения товаров ТоварView

```
class ТоварView(View):
    def get(self, request):
        a = ТоварModel.objects.all()
        return render(request, 'order.html', {'a':a})
```

Urls.py

```
url(r'^test/$', ТоварView.as_view())
```

Шаблон

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

{% block title %}Товары{% endblock %}

{% block content %}

    Список товаров
    {% for i in a %}
        <br>{{ i.name }} : {{ i.desc }}
    {% empty %}
        <h1>Пустой список</h1>
    {% endfor %}
{% endblock %}
```

Скриншоты

Тестирование класса подключения без django

```
(2, 'Реймарк', 'Три товарища', '567.00')
(3, 'Толстой', 'Война и Мир', '567.00')
(4, 'Толстой', 'Война и Мир', '567.00')
(5, 'Толстой', 'Детство', '567.00')
(6, 'Толстой', 'Юность', '567.00')
(7, 'Толстой', 'Моложавость', '567.00')
(8, 'Толстой', 'Старость', '567.00')
(9, 'Толстой', 'Смерть', '567.00')
(10, 'Толстой', 'Смерть', '567.00')
(11, 'Толстой', 'Смерть', '567.00')
(12, 'Толстой', 'Смерть', '567.00')
```