

Защищено:
Гапанюк Ю.Е.

"__" _____ 2016 г.

Демонстрация ЛР:
Гапанюк Ю.Е.

"__" _____ 2016 г.

**Отчет по лабораторной работе № 6 по курсу
Разработка интернет приложений**

**"Лабораторная работа №6.
Работа с СУБД."**

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53 _____
(подпись)

Бирюкова Д. К.

"__" _____ 2016 г.

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

Текст программы:

Example.py:

```
import MySQLdb

db = MySQLdb.connect( host='localhost',

    user='dbuser', passwd='123',

    db='first_db', use_unicode=True,

    charset='utf8'

)

c = db.cursor()

c.execute("INSERT INTO team (name, description) VALUES (%s, %s);", ('Зенит',
'Российский футбольный клуб')) db.commit()

c.execute("SELECT * FROM team;") teams = c.fetchall()
for team in teams:
    print(team)

c.execute("DELETE FROM team;") db.commit()

c.execute("SELECT * FROM team;") teams = c.fetchall()
print('БД после удаления:')
for team in teams:
    print(team)

c.close()
db.close()
)
```

example2.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,
                use_unicode=True,
                charset='utf8'
            )

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

class Team:
    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 team (name, description) VALUES
(%s, %s);", (self.name, self.description))
        self.db_connection.commit()
        c.close()

class Teams:
    def __init__(self, db_connection):
        self.db_connection = db_connection.connection

    def select_all(self):
        c = self.db_connection.cursor()
        c.execute("SELECT * FROM team;")
        teams = c.fetchall()
        c.close()
        return teams

    def delete_all(self):
        c = self.db_connection.cursor()
        c.execute("TRUNCATE table team;")
        self.db_connection.commit()
        c.close()

```

```

con = Connection('dbuser', '123',
'first_db') with con:
    team = Team(con, 'ЦСКА', 'Российский
футбольный клуб') team.save()
    team = Team(con, 'Зенит',
'РФ клуб') team.save()
    teams = Teams(con)
    select_teams = teams.select_all()
    print(select_teams)
    teams.delete_all() print('-----
-----') select_teams
    = teams.select_all()
    print(select_teams)

```

models.py:

```

from django.db import models

class User(models.Model):
    def __str__(self):
        return self.first_name
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    age = models.IntegerField()
    email = models.EmailField()

class Team(models.Model):
    def __str__(self):
        return self.name
    name = models.CharField(max_length=30)
    description = models.TextField()

```

views.py:

```

from django.shortcuts import render
from django.views import View

from DataBaseApp.models import Team, User

def main_page(request):
    return render(request, 'index.html')

class TeamView(View):
    def get(self, request):
        teams = Team.objects.all()
        return render(request, 'teams.html', {'teams': teams})

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

```

Результаты работы программы:

