

Отчет по лабораторной работе №6
По курсу: Разработка интернет-приложений

Выполнил
Студент гр.ИУ5-51
Дробышева Виктория

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

Test.py

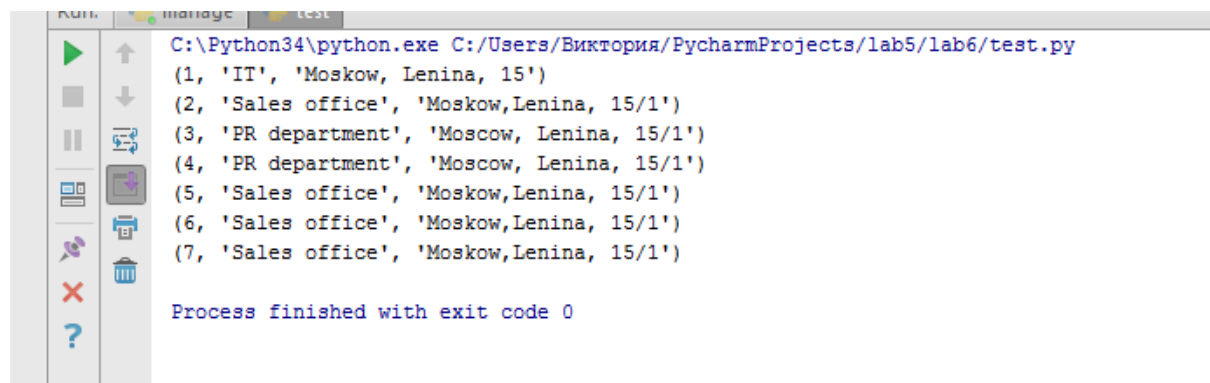
```
import MySQLdb

db = MySQLdb.connect (
    host='localhost',
    user='root',
    passwd='1234',
    db='first_db'
)

c = db.cursor()
c.execute('insert into office (named,location)VALUES (%s,%s);', ('Sales
office', 'Moscow,Lenina, 15/1'))
db.commit()

c.execute('select * from office;')
entries = c.fetchall()
for e in entries:
    print(e)

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



Connect.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
            )

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

class Offices:
    def __init__(self, db_connection, named, location):
        self.db_connection = db_connection.connection
        self.named = named
        self.location = location

    def save(self):
        c = self.db_connection.cursor()
        c.execute("insert into office (named, location) values (%s, %s);",
                  (self.named, self.location))
        self.db_connection.commit()
        c.close()

con = Connection('root', '1234', 'first_db')

with con:
    m = Offices(con, 'PR department', 'Moscow, Lenina, 15/1')
    m.save()

class Members:
    def __init__(self, db_connection, name, position):
        self.db_connection = db_connection.connection
        self.name = name
        self.position = position

    def save(self):
        c = self.db_connection.cursor()
        c.execute("insert into members (name, position) values (%s, %s);",
                  (self.name, self.position))
        self.db_connection.commit()
        c.close()

con = Connection('root', '1234', 'first_db')

with con:
    m = Members(con, 'Ivanov', 'lawyer')
    m.save()

```

Limit to 1000 rows

1 • `SELECT * FROM first_db.members;`

id	name	position
1	Лебедев Сергей Петрович	старший менеджер
2	Name	Position
9	Petrov	programmer
10	Ivanov	lawyer
11	Ivanov	lawyer
*	NULL	NULL

members 1 x

Apply Revert

odel rip_app_officesmodel rip_app_membersmodel_office rip_app_membersmodel auth_user members office

Limit to 1000 rows

1 • `SELECT * FROM first_db.office;`

id	named	location
5	Sales office	Moskow, Lenina, 15/1
6	Sales office	Moskow, Lenina, 15/1
7	Sales office	Moskow, Lenina, 15/1
8	PR department	Moscow, Lenina, 15/1
*	NULL	NULL

office 1 x

Apply Revert

Models.py

```
from __future__ import unicode_literals
from django.db import models
from django.contrib.auth.models import User
```

```
class OfficesModel(models.Model):
    named = models.CharField(max_length=30, verbose_name=u'Название',
blank=True, null=True)
    location = models.CharField(max_length=255, verbose_name=u'Адрес',
blank=True, null=True)
    picture = models.ImageField(upload_to="test/", blank=True, null=True,
verbose_name=u'')

    def __str__(self):
        return self.named
```

```
class MembersModel(models.Model):
    office = models.ManyToManyField(OfficesModel, related_name="members")
    f_name = models.CharField(max_length=100, verbose_name=u'Name',
```

```

null=True)
    l_name = models.CharField(max_length=100, verbose_name=u'L_name',
null=True)
    position = models.CharField(max_length=255, verbose_name=u'Position',
default='-----')
    user = models.OneToOneField(User, null=True, related_name='member')

    def __str__(self):
        return self.l_name

```

views.py

```

class OfficesListView(ListView):
    template_name = 'include/office.html'
    model = OfficesModel
    context_object_name = 'Offices'
    paginate_by = 10

class Offices(OfficesListView):
    template_name = 'index.html'

class OneOffice(DetailView):
    template_name = 'include/one_office.html'
    model = OfficesModel
    context_object_name = 'Offices'

    def get_object(self):
        object = super(OneOffice, self).get_object()
        if not self.request.user.is_authenticated():
            raise Http404
        return object

    def get_members(request, office_id):
        members = get_object_or_404(MembersModel, pk=office_id)
        return render(request, '/office/' + str(office_id) + '/', {'members':
members})

```

Query: `SELECT * FROM first_db.rip_app_membersmodel;`

id	position	user_id	f_name	l_name
1	Программист	9	Сергей Игоревич	Лебедев
2	Менеджер	8	Ася Владимировна	Андреева
3	Бухгалтер	7	Злата Андреевна	Внукова
4	Секретарь	3	Виктория Олеговна	Дробышева
7	-----	16	Валерия Сергеевна	Семенова
*	NULL	NULL	NULL	NULL

Model: rip_app_officesmodel rip_app_membersmodel_office rip_app_membersmodel auth_user members office

Limit to 1000 rows

1 • `SELECT * FROM first_db.rip_app_officesmodel;`

Result Grid

	id	named	location	picture
▶	1	IT отдел	Москва, ул.Ленина, 15	images/it_dept.png
	2	Отдел продаж	Москва, ул.Ленина, 15	images/avatar.jpeg
	3	Отдел финансов	Москва, ул.Ленина, 15/1	test/финансы.jpg
	13	Отдел маркетинга	Москва, ул.Ленина, 13	images/marketing.png
	17	Отдел кадров	Москва, ул.Ленина, 15/1	test/Кадры.jpg
	19	Архитектурный отдел	Москва, ул.Дуброва, 8	test/архитек.jpg
	31	Отдел телекоммуникаций	Москва, ул.Новая, 3	test/телекомм.jpg
	41	Отдел связи	Москва, ул.Никонорова, 7/5	test/связь.png
	45	Инженерный отдел	Москва, ул.Дуброва 8/1	test/инженер.png

rip_app_officesmodel 3 x

Apply Revert

Диалоги x DrViktoria/lab6_repo x lu5web/ЛР6. Работа с x ЛР6. Работа с СУБД.pdf x WorkSite x Пользователь 1


127.0.0.1:8000

WorkSite

Вы вошли как Drobysheva
[Выйти](#)


Офисы

[Создать офис](#)




Название офиса: IT отдел
Расположение: Москва, ул.Ленина, 15

[Подробнее](#)



Название офиса: Отдел продаж
Расположение: Москва, ул.Ленина, 15

[Подробнее](#)



Название офиса:
Расположение:

[Подробнее](#)

Виктория Дробышева