

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

**Разработка интернет-приложений
Лабораторная работа № 6
“Работа с СУБД”**

Выполнил:
студент группы ИУ5-53
Кузнецов В.И.

Подпись:
Дата:

Москва 2017г.

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.
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

Файлsql.py

```
import MySQLdb

db = MySQLdb.connect(
    host="localhost",
    user="dbuser",
    passwd="123",
    db="first_db",
    charset="utf8"
)

c = db.cursor()

# c.execute("INSERT INTO banks (name, address) VALUES (%s, %s);", ('Название', 'адрес'))
# db.commit()

c.execute("SELECT *FROM banks;")

entries = c.fetchall()

for e in entries:
    print(e)

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

Файл next.py

```
#!/usr/bin/env python
# -*- coding:utf8 -*-
import MySQLdb

class Connection:
    def __init__(self, user, password, db, host='localhost', charset='utf8'):
        # параметры соединения
        self.user = user
        self.host = host
        self.password = password
        self.db = db
        self._connection = None
        self.use_unicode = True
        self.charset = charset

    @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 соедин.
        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 соедин.
        if self._connection:
            self._connection.close()

class Bank:
    def __init__(self, db_connection, name, address):
        # соедин. и данные о банке
        self.db_connection = db_connection.connection
        self.name = name
        self.address = address

    def save(self):
        # запись данных из объекта в БД
        c = self.db_connection.cursor()
        c.execute("INSERT INTO banks (name, address) VALUES (%s, %s);", (self.name,
                                                                           self.address))
        self.db_connection.commit()
        c.close()

if __name__ == '__main__':
    # sfd
    con = Connection("dbuser", "123", "first_db")

    with con:
        bank = Bank(con, "вапв1", "fdg3")
        bank.save()
```

Файл models.py

```
class BankModel(models.Model):
    class Meta:
        db_table = 'banks'
    idBanks = models.AutoField(primary_key=True)
    name = models.CharField(max_length=100)
    address = models.CharField(max_length=200)
    phone = models.CharField(max_length=16)
    comments = models.CharField(max_length=1000)

    def __str__(self):
        return "name: {}, address: {}, phone number: {}, comments: {}".format(self.name, self.address, self.phone,
                                                                                self.comments)

class CustomerModel(models.Model):
    class Meta:
        db_table = 'customer'
    idCustomer = models.AutoField(primary_key=True)
    fio = models.CharField(max_length=200)
    login = models.CharField(max_length=45, unique=True)
    password = models.CharField(max_length=45)
    phone = models.CharField(max_length=16, unique=True)

    def __str__(self):
        return "id: {}, fio: {}, login: {}, password: {}, phone: {}".format(self.idCustomer, self.fio,
                                                                              self.login, self.password, self.phone)

class AccountModel(models.Model):
    class Meta:
        db_table = 'accounts'
    idAccounts = models.AutoField(primary_key=True)
    customerId_FK = models.ForeignKey(
        CustomerModel,
        db_column='customerId_FK'
    )
```

```

        db_column='customerId_FK'
    )
    bankId_FK = models.ForeignKey(
        BankModel,
        db_column='bankId_FK'
    )
    type = models.CharField(max_length=30)
    money = models.IntegerField()
    currency = models.CharField(max_length=10)

class TransactionsModel(models.Model):
    class Meta:
        db_table = 'transactions'
    idTransactions = models.AutoField(primary_key=True)

    customerId_from = models.ForeignKey(
        CustomerModel,
        related_name='customerId_from',
        db_column='CustomerId_from'
    )
    customerId_to = models.ForeignKey(
        CustomerModel,
        related_name='customerId_to',
        db_column='CustomerId_to'
    )

    accountId_to = models.ForeignKey(
        AccountModel,
        related_name='accountId_to',
        db_column='AccountId_to'
    )
    accountId_from = models.ForeignKey(
        AccountModel,
        related_name='accountId_from',
        db_column='AccountId_from'

```

```

    )
    money = models.IntegerField()
    currency = models.CharField(max_length=10)
    comment = models.CharField(max_length=100)
    time = models.DateField()

```

Views.py

```

class Home(View):
    def get(self, request):
        return render(request, 'main.html')

class Bank(View):
    def get(self, request):
        data = BankModel.objects.all()
        return render(request, 'banks_main.html', {'banks': data})

class BankInDetail(View):
    def get(self, request, id):
        data = BankModel.objects.get(idBanks=int(id))
        '''...'''
        return render(request, 'bank_detail.html', {'banks': data})

class Customer(View):
    def get(self, request):
        data_cus = CustomerModel.objects.all()
        return render(request, 'customer_main.html', {'customers': data_cus})

class CustomerInDetail(View):
    def get(self, request, id):
        data = CustomerModel.objects.get(idCustomer=int(id))
        data1 = AccountModel.objects.filter(customerId_FK=int(id))
        data2 = TransactionsModel.objects.filter(customerId_from=int(id))
        return render(request, 'customer_detail.html', {'customers': data, 'accts': data1, 'trans': data2})

class CustomerAccounts(View):
    def get(self, request):
        data = AccountModel.objects.all()

```

```

class CustomerAccounts(View):
    def get(self, request):
        data = AccountModel.objects.all()
        return render(request, 'accounts_main.html', {'accts': data})

class CustomerTransactions(View):
    def get(self, request):
        data = TransactionsModel.objects.all()
        return render(request, 'transactions_main.html', {'trans': data})

```

Urls.py

```

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^home/$', Home.as_view(), name='home'),

    url(r'^bank/$', Bank.as_view(), name='banks_main'),
    url(r'^bank/(?P<id>\d+)/$', BankInDetail.as_view(), name='banks_detail'),
    url(r'^customers/$', Customer.as_view(), name='customers_main'),
    url(r'^customers/(?P<id>\d+)/$', CustomerInDetail.as_view(), name='customers_detail'),
    url(r'^accounts/$', CustomerAccounts.as_view(), name='accts'),
    url(r'^transactions/$', CustomerTransactions.as_view(), name='trans'),
    url(r'^reg/$', registration, name='reg'),
    url(r'^login/$', vlog.as_view(), name='login')]

```

Итог

[В начало](#) [Customers](#) [Banks](#) [Transactions](#)

ФИО: Сметанкин К.И.

Login: smet

Password: 123

Phone: +7 999 888 77 66

Ваши счета:

id Счета: 1
id Банка: 9 Название: Сбербанк России
Тип: debit
Сумма: 1000 rub

id Счета: 2
id Банка: 10 Название: Банк ВТБ
Тип: credit
Сумма: 100 usd

Ваши переводы:

id Транзакции: 1
id From: 1 ФИО: Сметанкин К.И.
id Счета (на какой): 1
Сумма: 500 rub
Комменты: test
Дата: Nov. 30, 2017, 12:57 p.m.

[В начало](#) [Customers](#) [Banks](#) [Transactions](#) [Accounts](#)

Название банка: Сбербанк России
Адрес: Россия, Москва, 117997, ул. Вавилова, д. 19
Телефон: +7 800 555-55-50



Коротко о банке:

Акционерный коммерческий Сберегательный банк Российской Федерации (Сбербанк России) создан в форме акционерного общества открытого типа в соответствии с Законом РСФСР «О банках и банковской деятельности в РСФСР». Учредителем и основным акционером Сбербанка России является Центральный банк Российской Федерации (свыше 60% акций уставного капитала). Его акционерами являются более 200 тысяч юридических и физических лиц.

В начало

Customers

Banks

Transactions

Accounts

Все переводы:

id Транзакции: 1

id From: 1 ФИО: Сметанкин К.И.

id Счета (на какой): 1

Сумма: 500 rub

Комменты: test

Дата: Nov. 30, 2017, 12:57 p.m.

В начало

Customers

Banks

Transactions

Accounts

Все счета:

id Счета: 1

id Банка: 9 Название: Сбербанк России

Тип: debit

Сумма: 1000 rub

id Счета: 2

id Банка: 10 Название: Банк ВТБ

Тип: credit

Сумма: 100 usd