**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕНЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.Шухова»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №7

По дисциплине: Базы данных

Тема: «Использование ORM при работе с базой данных»

Выполнила: ст. группы ПВ-31

Донцов А.А.

Проверил: Панченко М.В.

Белгород 2019

**Цель работы:** изучить особенности работы с базами данных с использованием ORM.

Models.py

from django.db import models

class AuthGroup(models.Model):

name = models.CharField(unique=True, max\_length=150)

class Meta:

managed = False

db\_table = 'auth\_group'

class AuthGroupPermissions(models.Model):

group = models.ForeignKey(AuthGroup, models.DO\_NOTHING)

permission = models.ForeignKey('AuthPermission', models.DO\_NOTHING)

class Meta:

managed = False

db\_table = 'auth\_group\_permissions'

unique\_together = (('group', 'permission'),)

class AuthPermission(models.Model):

name = models.CharField(max\_length=255)

content\_type = models.ForeignKey('DjangoContentType', models.DO\_NOTHING)

codename = models.CharField(max\_length=100)

class Meta:

managed = False

db\_table = 'auth\_permission'

unique\_together = (('content\_type', 'codename'),)

class AuthUser(models.Model):

password = models.CharField(max\_length=128)

last\_login = models.DateTimeField(blank=True, null=True)

is\_superuser = models.BooleanField()

username = models.CharField(unique=True, max\_length=150)

first\_name = models.CharField(max\_length=30)

last\_name = models.CharField(max\_length=150)

email = models.CharField(max\_length=254)

is\_staff = models.BooleanField()

is\_active = models.BooleanField()

date\_joined = models.DateTimeField()

class Meta:

managed = False

db\_table = 'auth\_user'

class AuthUserGroups(models.Model):

user = models.ForeignKey(AuthUser, models.DO\_NOTHING)

group = models.ForeignKey(AuthGroup, models.DO\_NOTHING)

class Meta:

managed = False

db\_table = 'auth\_user\_groups'

unique\_together = (('user', 'group'),)

class AuthUserUserPermissions(models.Model):

user = models.ForeignKey(AuthUser, models.DO\_NOTHING)

permission = models.ForeignKey(AuthPermission, models.DO\_NOTHING)

class Meta:

managed = False

db\_table = 'auth\_user\_user\_permissions'

unique\_together = (('user', 'permission'),)

class Confederacy(models.Model):

confederacy\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=50)

class Meta:

managed = False

db\_table = 'confederacy'

class DjangoAdminLog(models.Model):

action\_time = models.DateTimeField()

object\_id = models.TextField(blank=True, null=True)

object\_repr = models.CharField(max\_length=200)

action\_flag = models.SmallIntegerField()

change\_message = models.TextField()

content\_type = models.ForeignKey('DjangoContentType', models.DO\_NOTHING, blank=True, null=True)

user = models.ForeignKey(AuthUser, models.DO\_NOTHING)

class Meta:

managed = False

db\_table = 'django\_admin\_log'

class DjangoContentType(models.Model):

app\_label = models.CharField(max\_length=100)

model = models.CharField(max\_length=100)

class Meta:

managed = False

db\_table = 'django\_content\_type'

unique\_together = (('app\_label', 'model'),)

class DjangoMigrations(models.Model):

app = models.CharField(max\_length=255)

name = models.CharField(max\_length=255)

applied = models.DateTimeField()

class Meta:

managed = False

db\_table = 'django\_migrations'

class DjangoSession(models.Model):

session\_key = models.CharField(primary\_key=True, max\_length=40)

session\_data = models.TextField()

expire\_date = models.DateTimeField()

class Meta:

managed = False

db\_table = 'django\_session'

class Sponsor(models.Model):

sponsor\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=50)

class Meta:

managed = False

db\_table = 'sponsor'

class Club(models.Model):

club\_id = models.AutoField(primary\_key=True)

name = models.CharField(max\_length=50)

rating = models.IntegerField(blank=True, null=True)

confederacy = models.ForeignKey(Confederacy, models.DO\_NOTHING, blank=True, null=True)

sponsor = models.ForeignKey(Sponsor, models.DO\_NOTHING, blank=True, null=True)

class Meta:

managed = False

db\_table = 'club'

class Match(models.Model):

match\_id = models.AutoField(primary\_key=True)

playdata = models.DateTimeField(blank=True, null=True)

count1 = models.IntegerField()

club\_id1 = models.ForeignKey(Club, models.DO\_NOTHING, related\_name = 'club\_id1', db\_column='club\_id1', blank=True, null=True)

count2 = models.IntegerField()

club\_id2 = models.ForeignKey(Club, models.DO\_NOTHING, related\_name = 'club\_id2', db\_column='club\_id2', blank=True, null=True)

sold = models.IntegerField()

class Meta:

managed = False

db\_table = 'match'

class Coach(models.Model):

coach\_id = models.AutoField(primary\_key=True)

club = models.ForeignKey(Club, models.DO\_NOTHING, blank=True, null=True)

name = models.CharField(max\_length=50)

lastname = models.CharField(max\_length=50)

class Meta:

managed = False

db\_table = 'coach'

class Player(models.Model):

player\_id = models.AutoField(primary\_key=True)

club = models.ForeignKey(Club, models.DO\_NOTHING, blank=True, null=True)

name = models.CharField(max\_length=50)

lastname = models.CharField(max\_length=50)

efficiency = models.IntegerField(blank=True, null=True)

class Meta:

managed = False

db\_table = 'player'

views.py

from django.shortcuts import render

from database.models import Sponsor, Confederacy, Club, Coach, Match, Player

def home(request):

return render(request, 'database/home.html', {})

def view(request):

return render(request, 'database/view.html', {})

def club(request):

club=Club.objects.all()

return render(request, 'database/club.html', {'clubs':club})

def coach(request):

coach=Coach.objects.all()

return render(request, 'database/coach.html', {'coachs':coach})

def confederacy(request):

confederacy=Confederacy.objects.all()

return render(request, 'database/confederacy.html', {'confederacys':confederacy})

def match(request):

match=Match.objects.all()

return render(request, 'database/match.html', {'matchs':match})

def player(request):

player=Player.objects.all()

return render(request, 'database/player.html', {'players':player})

def sponsor(request):

sponsor = Sponsor.objects.all()

return render(request, 'database/sponsor.html', {'sponsors':sponsor})

urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.home, name='home'),

path('view', views.view, name='view'),

path('club', views.club, name='club'),

path('coach', views.coach, name='coach'),

path('confederacy', views.confederacy, name='confederacy'),

path('match', views.match, name='match'),

path('player', views.player, name='player'),

path('sponsor', views.sponsor, name='sponsor'),

]









