**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Отчет по лабораторной работе №5

«Работа с СУБД. Обработка данных с использованием Django ORM.»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Барышников Михаил |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2021 г.

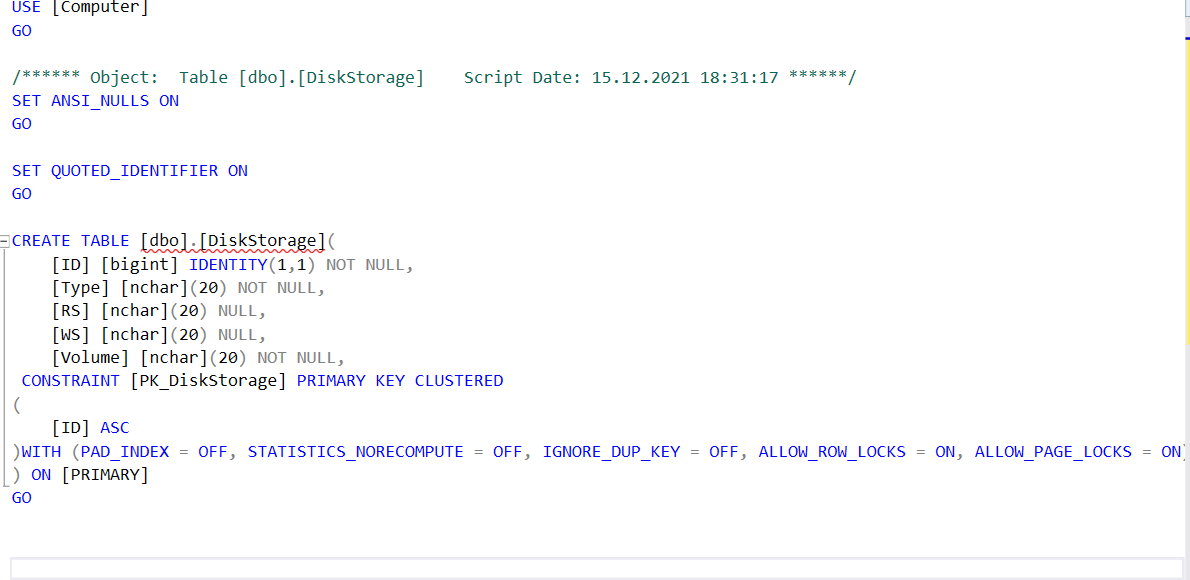
**Описание задания:**

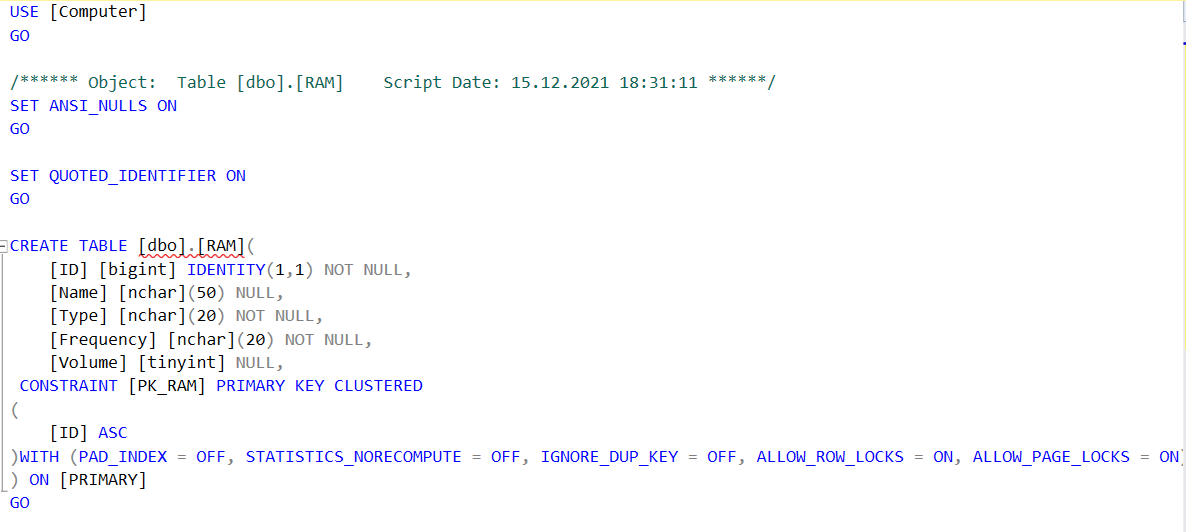
В этой лабораторной работе Вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также Вам нужно будет дополнить свои классы предметной области, связав их с созданной БД. После этого Вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей.

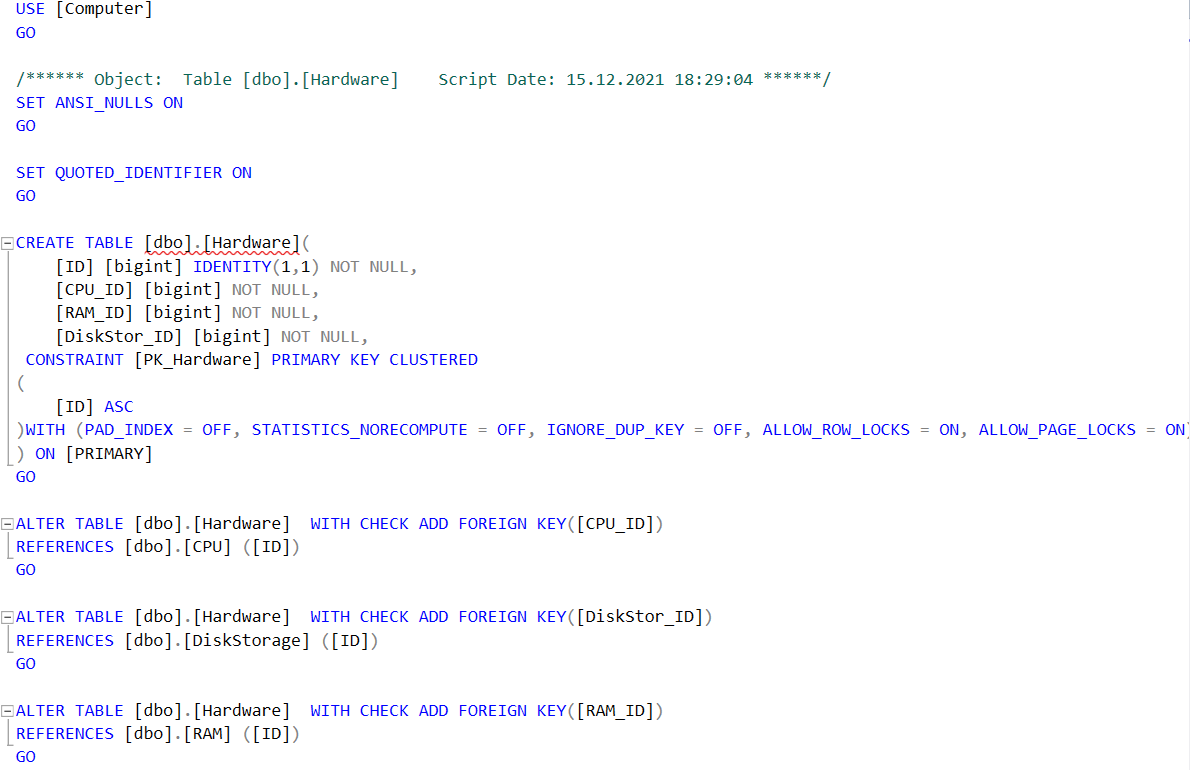
1. Создайте сценарий с подключением к БД и несколькими запросами, примеры рассмотрены в [методических указаниях.](https://github.com/iu5team/iu5web-fall-2021/blob/main/tutorials/lab5/lab5_tutorial.md)
2. Реализуйте модели Вашей предметной области из предыдущей ЛР (минимум две модели, т.е. две таблицы).
3. Создайте представления и шаблоны Django для отображения списка данных по каждой из сущностей.

**Создание таблиц базы данных:**









**Скрипт подключения к базе данных, добавления записей в таблицу и выборки данных:**

import pyodbc

connectionString = ("Driver={SQL Server Native Client 11.0};"

"Server=localhost\\MSSQLSERVERDEV;"

"Database=Computer;"

"Trusted\_Connection=yes;")

request1 = """INSERT INTO dbo.CPU (Name, Frequency, Architecture) VALUES ('Intel core i5', '5Ghz', 'x86');"""

request2 = "SELECT \* FROM dbo.CPU"

connection = pyodbc.connect(connectionString, autocommit=True)

dbCursor = connection.cursor()

#dbCursor.execute(request1)

dbCursor.execute(request2)

for row in dbCursor:

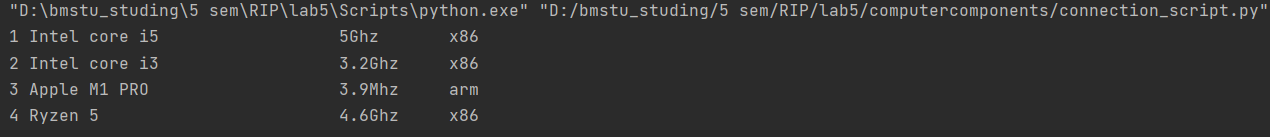
print(f"{row.ID} {row.Name} {row.Frequency} {row.Architecture}")

connection.commit()

dbCursor.close()

connection.close()

**Результат:**



**Файл main/models.py с созданием классов моделей таблиц на основе кода, сгенерированного с помощью команды «python manage.py inspectdb»:**

from django.db import models

class Cpu(models.Model):

id = models.BigAutoField(db\_column='ID', primary\_key=True) # Field name made lowercase.

name = models.CharField(db\_column='Name', max\_length=30) # Field name made lowercase.

frequency = models.CharField(db\_column='Frequency', max\_length=10, blank=True, null=True) # Field name made lowercase.

architecture = models.CharField(db\_column='Architecture', max\_length=20, blank=True, null=True) # Field name made lowercase.

class Meta:

db\_table = 'CPU'

class Diskstorage(models.Model):

id = models.BigAutoField(db\_column='ID', primary\_key=True) # Field name made lowercase.

type = models.CharField(db\_column='Type', max\_length=20) # Field name made lowercase.

rs = models.CharField(db\_column='RS', max\_length=20, blank=True, null=True) # Field name made lowercase.

ws = models.CharField(db\_column='WS', max\_length=20, blank=True, null=True) # Field name made lowercase.

volume = models.CharField(db\_column='Volume', max\_length=20) # Field name made lowercase.

class Meta:

db\_table = 'DiskStorage'

class Hardware(models.Model):

id = models.BigAutoField(db\_column='ID', primary\_key=True) # Field name made lowercase.

cpu = models.ForeignKey(Cpu, models.DO\_NOTHING, db\_column='CPU\_ID') # Field name made lowercase.

ram = models.ForeignKey('Ram', models.DO\_NOTHING, db\_column='RAM\_ID') # Field name made lowercase.

diskstor = models.ForeignKey(Diskstorage, models.DO\_NOTHING, db\_column='DiskStor\_ID') # Field name made lowercase.

class Meta:

db\_table = 'Hardware'

class Ram(models.Model):

id = models.BigAutoField(db\_column='ID', primary\_key=True) # Field name made lowercase.

name = models.CharField(db\_column='Name', max\_length=50, blank=True, null=True) # Field name made lowercase.

type = models.CharField(db\_column='Type', max\_length=20) # Field name made lowercase.

frequency = models.CharField(db\_column='Frequency', max\_length=20) # Field name made lowercase.

volume = models.SmallIntegerField(db\_column='Volume', blank=True, null=True) # Field name made lowercase.

class Meta:

db\_table = 'RAM'

**Файл computercomponents/urls.py:**

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('main.urls'))

]

**Файл main/urls.py:**

from django.urls import path

from . import views

urlpatterns = [

path('', views.index),

path('<str:model\_name>/', views.list, name='list')

]

**Файл main/views.py:**

from django.shortcuts import render

from .models import \*

from django.apps import apps

def index(request):

models = apps.get\_app\_config('main').get\_models()

models\_names = [model.\_meta.db\_table for model in models]

models\_dict = {model\_id: models\_name for model\_id, models\_name in zip(range(len(models\_names)), models\_names)}

params = {'models\_dict': models\_dict}

return render(request, 'main/index.html', params)

def list(request, model\_name):

models = apps.get\_app\_config('main').get\_models()

model = ''

for elem in models:

if elem.\_meta.db\_table == model\_name:

model = elem

params = {'model\_name': model.\_meta.db\_table, 'objects': model.objects.values()}

return render(request, 'main/list.html', params)

**Файл main/templates/main/base.html:**

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>{% block title %}{% endblock %}</title>

</head>

<body>

<a href='/'>Главная</a>

{% block content %}{% endblock %}

</body>

</html>

**Файл main/templates/main/index.html:**

{% extends 'main/base.html' %}

{% block title %}

Главная

{% endblock %}

{% block content %}

<h2>Подключенная база данных содержит следующие сущности:</h2>

<ol>

{% for model\_id, model\_name in models\_dict.items %}

<li><a href="{% url 'list' model\_name %}">{{ model\_name }}</a></li>

<br>

{% endfor %}

</ol>

{% endblock %}

**Файл main/templates/main/list.html:**

{% extends 'main/base.html' %}

{% block title %}

{{ model\_name }}

{% endblock %}

{% block content %}

<h2>Cущность <i>{{ model\_name }}</i></h2>

<ul>

{% for object in objects %}

<li>

{% for key, value in object.items %}

<i>{{key}}</i>: {{value}}

<br>

{% endfor %}

</li>

<br>

{% endfor %}

</ul>

{% endblock %}

**Прототип веб-приложения:**

