

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 рубежному контролю №2

Выполнил:  
студент группы ИУ5-53Б

Баркалова Ирина

Дата: 19.12.2021

Москва. 2021 г.

### Вариант РК:

### Вариант В.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с минимальной зарплатой сотрудников в каждом отделе, отсортированный по минимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.

4	Компьютер	Дисплейный класс
---	-----------	------------------

Код программы:

*models.py*

```
from django.db import models

class PC(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=20)

    class Meta:
        db_table = 'pc'

class Display(models.Model):
    id = models.IntegerField(primary_key=True)
    matrix_type = models.TextField()
    diagonal = models.FloatField()
    pc = models.ForeignKey(PC, on_delete=models.PROTECT)

    class Meta:
        db_table = 'display'
```

*serializers.py*

```
from django.db import models

class PC(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=20)

    class Meta:
        db_table = 'pc'

class Display(models.Model):
    id = models.IntegerField(primary_key=True)
    matrix_type = models.TextField()
    diagonal = models.FloatField()
    pc = models.ForeignKey(PC, on_delete=models.PROTECT)

    class Meta:
        db_table = 'display'
```

*views.py*

```

from django.shortcuts import render
from rest_framework import viewsets

from computers.models import PC, Display
from computers.serializers import PCSerializer, DisplaySerializer

class PCViewSet(viewsets.ModelViewSet):
    queryset = PC.objects.all()
    serializer_class = PCSerializer

class DisplayViewSet(viewsets.ModelViewSet):
    queryset = Display.objects.all()
    serializer_class = DisplaySerializer

def report(request):
    return render(request, 'report.html', {'data': {
        'display': Display.objects.select_related('pc')
    }})

```

*settings.py*

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'rk2_db',
        'USER': 'iris',
        'PASSWORD': '000000',
        'HOST': 'localhost',
        'PORT': 5432,
    }
}

```

### *urls.py*

```
from rest_framework import routers
from django.urls import path, include
from django.contrib import admin
from django.urls import path
from computers import views as views

router = routers.DefaultRouter()
router.register('display', views.PCViewSet)
router.register('pc', views.DisplayViewSet)

urlpatterns = [

    path('', include(router.urls)),
    path('report/', views.report),
    path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
    path('admin/', admin.site.urls),
]
```

### *Base.html*

```
<!doctype html>
<html lang="en" class="h-100">
<head>
    <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    {% block content %}{% endblock %}
</body>
</html>
```

### *details.html*

```
<!doctype html>
<html lang="en" class="h-100">
<head>
    <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    {% block content %}{% endblock %}
</body>
</html>
```

Привет запросов:

GET

http://127.0.0.1:8000/pc

Send

ParamsAuthorizationHeaders (8)Body●Pre-request ScriptTestsSettingsCookies

BodyCookiesHeaders (10)Test Results

200 OK24 ms845 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   {
3     "id": 1,
4     "matrix_type": "IPS",
5     "diagonal": 13.0,
6     "pc": 8
7   },
8   {
9     "id": 2,
10    "matrix_type": "TN",
11    "diagonal": 14.0,
12    "pc": 7
13  },
14  {
15    "id": 3,
16    "matrix_type": "VA",
17    "diagonal": 15.6,
18    "pc": 6
19  },
20  {
21    "id": 4,
22    "matrix_type": "VA",
23    "diagonal": 24.0,
24    "pc": 5
25  }
26 }
```

GET

http://127.0.0.1:8000/display

Send

GET

http://127.0.0.1:8000/display

Send

ParamsAuthorizationHeaders (8)Body●Pre-request ScriptTestsSettingsCookies

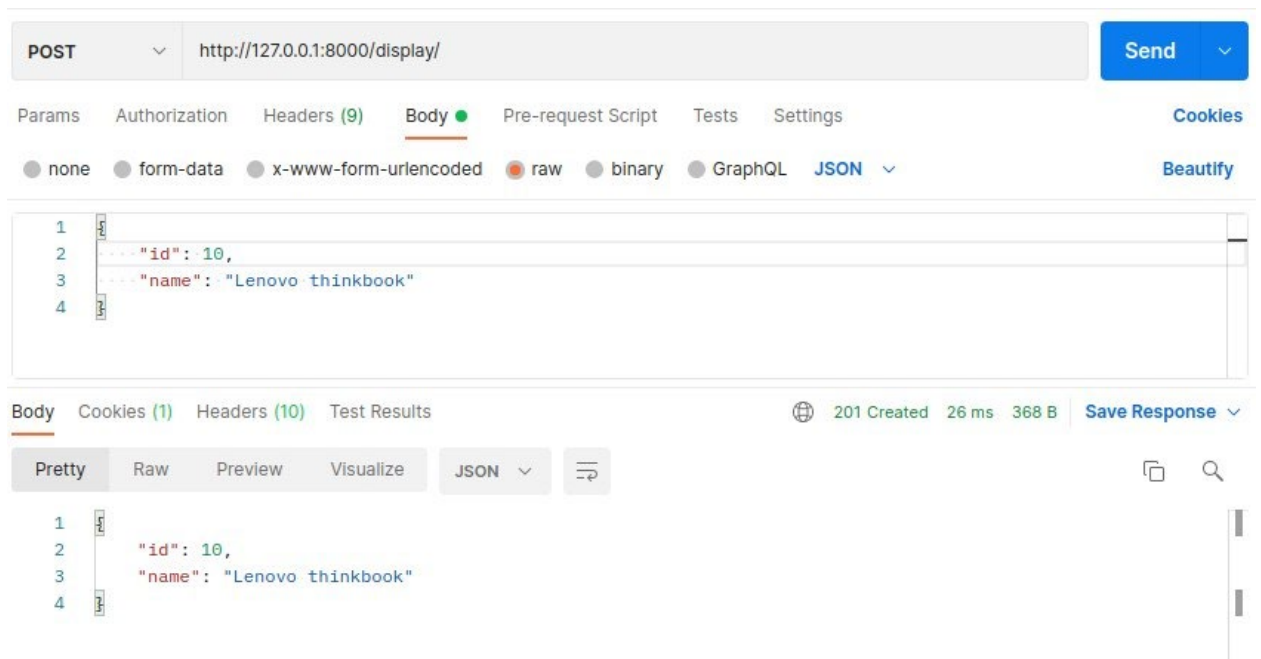
BodyCookiesHeaders (10)Test Results

200 OK26 ms606 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   {
3     "id": 1,
4     "name": "Lenovo IdeaCentre G5"
5   },
6   {
7     "id": 2,
8     "name": "HP Pavilion Gaming"
9   },
10  {
11    "id": 3,
12    "name": "ZOTAC MAGNUS ONE"
13  },
14  {
15    "id": 4,
16    "name": "Apple Mac mini 2020"
17  },
18  {
19    "id": 5,
20    "name": "Acer Aspire TC-895"
21  },
22  {
23    "id": 6,
24    "name": "Apple iMac 24"
25  }
26 }
```





Результат работы программы:

## Дисплей:

- Матрица: *IPS*
- Диагональ: *13.0*
- Название компьютера: *HP M01*
- Матрица: *TN*
- Диагональ: *14.0*
- Название компьютера: *Gigabyte GB-BR*
- Матрица: *VA*
- Диагональ: *15.6*
- Название компьютера: *Apple iMac 24*
- Матрица: *VA*
- Диагональ: *24.0*
- Название компьютера: *Acer Aspire TC-895*
- Матрица: *IPS*
- Диагональ: *12.0*
- Название компьютера: *Apple Mac mini 2020*
- Матрица: *TN*
- Диагональ: *13.0*
- Название компьютера: *ZOTAC MAGNUS ONE*
- Матрица: *TN*
- Диагональ: *17.0*
- Название компьютера: *HP Pavilion Gaming*
- Матрица: *VA*
- Диагональ: *15.6*
- Название компьютера: *Lenovo IdeaCentre G5*
- Матрица: *IPS*
- Диагональ: *24.0*
- Название компьютера: *Lenovo IdeaCentre G5*
- Матрица: *IPS*
- Диагональ: *17.0*
- Название компьютера: *Gigabyte GB-BR*