

**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 №1

Вариант №18

Выполнил:

студент группы ИУ5-53
Наказной Никита

Подпись и дата:
19.01.22

Проверил:

Юрий Евгеньевич Гапанюк

Подпись и дата:

Москва, 2022 г.

Задание:

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.

Код программы:

```
class Composition:

    def __init__(self, IDcom, Title, CompositorName, Age, IDorc):
        self.IDcom = IDcom
        self.Title = Title
        self.CompositorName = CompositorName
        self.Age = Age
        self.IDorc = IDorc

class Orchestra:

    def __init__(self, IDorch, Name, MemberNum):
        self.IDorch = IDorch
        self.Name = Name
        self.MemberNum = MemberNum

class OrchestraCompositions:

    def __init__(self, IDcom, IDorc):
        self.IDcom = IDcom
        self.IDorc = IDorc

class OrcheNamesMinAges:

    def __init__(self, Name, Age):
        self.Name = Name
        self.Age = Age

def composersAgeGet(OrcheNamesMinAges):
    return OrcheNamesMinAges.Age

Compositions = [Composition(1, "BAD HABITS", "ED SHEERAN", 10, 1),
                 Composition(2, "GOOD 4 U", "OLIVIA RODRIGO", 33, 1),
                 Composition(3, "PEACHES", "JUSTIN BIEBER/CAESAR/GIVEON", 5,
2),
                 Composition(4, "SAVE YOUR TEARS", "WEEKND", 9, 3),
```

```

        Composition(5, "MONTERO (CALL ME BY YOUR NAME)", "LIL NAS X",
12, 3),
        Composition(6, "LEVITATING", "DUA LIPA", 10, 4),
        Composition(7, "STAY", "KID LAROI & JUSTIN BIEBER", 8, 5),
        Composition(8, "HEAT WAVES", "GLASS ANIMALS", 3, 6),
        Composition(9, "EASY ON ME", "ADELE", 18, 6)
    ]

Orchestras = [Orchestra(1, "Simple music ensemble", 20),
    Orchestra(2, "Neorchestra", 13),
    Orchestra(3, "Imperialis Orchestra ", 28),
    Orchestra(4, "TCHAIKOVSKY ORCHESTRA ", 18),
    Orchestra(5, "Acapella-Sakartvelo", 12),
    Orchestra(6, "Russian Folk Orchestra Moscow", 80)
]

OrchestraCompositionsList = [
    OrchestraCompositions(1, 1),
    OrchestraCompositions(2, 1),
    OrchestraCompositions(3, 2),
    OrchestraCompositions(4, 3),
    OrchestraCompositions(5, 3),
    OrchestraCompositions(6, 4),
    OrchestraCompositions(7, 5),
    OrchestraCompositions(8, 6),
    OrchestraCompositions(9, 6)
]

"""1 inquiry"""
FirstLetter = "A"
CompositionsBeginA = list(x for x in Compositions if
x.Title.startswith(FirstLetter))
IDOrchestrasWithBatA = list(map(lambda x: x.IDorc, CompositionsBeginA))
OrchestrasWithBatA = list(
    map(lambda x: (x.Name, x.IDorch), filter(lambda x: x.IDorch in
IDOrchestrasWithBatA, Orchestras)))

OrchestrasById = {}
for (Name, IDorch) in OrchestrasWithBatA:
    OrchestrasById[IDorch] = Name

print("Composition Surname | Orchestra name")

for i in CompositionsBeginA:
    print(i.Title, " | ", OrchestrasById[i.IDorc], "\n")
print("\n")

"""2 inquiry"""

IDOrchestras = list(map(lambda x: x.IDorch, Orchestras))
Ages = list(map(lambda x: (x.Age, x.IDorc), filter(lambda x: x.IDorc in
IDOrchestras, Compositions)))

MinAgeByID = {}
for (Age, IDorc) in Ages:
    if IDorc not in MinAgeByID or MinAgeByID[IDorc] > Age:
        MinAgeByID[IDorc] = Age

ListOrchestraMinAges = list()
for i in Orchestras:
    ListOrchestraMinAges.append(OrcheNamesMinAges(i.Name,
MinAgeByID[i.IDorch]))

ListOrchestraMinAges.sort(key=compositorsAgeGet)

```

```

print("Orchestra name | Composition min age")

for i in ListOrchestraMinAges:
    print(i.Name, " | ", i.Age, "\n")
print("\n")

"""3 inquiry"""

CompositionsInfo = list(map(lambda x: (x.IDorc, x.Title, x.CompositorName),
                             filter(lambda x: x.IDorc in IDOrchestras,
                                    Compositions)))

CompositionsInfo.sort(key=lambda x: x[2])

print("Composition Title | Composition exp | Orchesra name | Orchesra
MemberNum")
for i in CompositionsInfo:
    for j in Orchestras:
        if (i[0] == j.IDorch): {
            print(i[1], " | ", i[2], " | ", j.Name, " | ", j.MemberNum,
"\n")
        }

```

Результат работы программы:

1.

```

Composition Surname | Orchestra name
A BAD HABITS | Simple music ensemble

```

2.

```

Orchestra name | Composition min age
Russian Folk Orchestra Moscow | 3

Neorchestra | 5

Acapella-Sakartvelo | 8

Imperialis Orchestra | 9

Simple music ensemble | 10

TCHAIKOVSKY ORCHESTRA | 10

```

3.

```

Composition Title | Composition exp | Orchesra name | Orchesra MemberNum
EASY ON ME | ADELE | Russian Folk Orchestra Moscow | 80

LEVITATING | DUA LIPA | TCHAIKOVSKY ORCHESTRA | 18

A BAD HABITS | ED SHEERAN | Simple music ensemble | 20

HEAT WAVES | GLASS ANIMALS | Russian Folk Orchestra Moscow | 80

PEACHES | JUSTIN BIEBER/CAESAR/GIVEON | Neorchestra | 13

STAY | KID LAROI & JUSTIN BIEBER | Acapella-Sakartvelo | 12

MONTERO (CALL ME BY YOUR NAME) | LIL NAS X | Imperialis Orchestra | 28

GOOD 4 U | OLIVIA RODRIGO | Simple music ensemble | 20

SAVE YOUR TEARS | WEEKND | Imperialis Orchestra | 28

```