



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ГУИМЦ

КАФЕДРА «Автоматизированные системы обработки информации и управления»

ДИСЦИПЛИНА «Разработка интернет-приложений»

РК №1

Вариант 27 (преподаватель -> учебный курс)

Студент: Дюжев С.А.

Группа: ИУ5Ц – 73Б

Преподаватель: Гапанюк Ю. Е.

Москва 2020 г.

Вариант 27 (преподаватель -> учебный курс)

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.

Исходный текст программы

```
# PK1 Дюжев С.А. ИУ5Ц-73Б
# Вариант 27 (преподаватель -> учебный курс)
# Тип запроса В
from operator import itemgetter

class Prep:
    """Преподаватель"""
    def __init__(self, id, fio, sal, course_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.course_id = course_id

class Course:
    """Учебный курс"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class PrepCourse:
    """
    'Преподаватели учебного курса' для реализации
    связи многие-ко-многим
    """
    def __init__(self, course_id, prep_id):
        self.course_id = course_id
        self.prep_id = prep_id

# Отделы
Courses = [
    Course(1, 'математический анализ'),
    Course(2, 'python3'),
    Course(3, 'Jango'),
    Course(4, 'алгебра'),
```

```
Course(5, 'литература 19 века'),
Course(6, 'подготовительная программа C/C++'),
Course(7, 'углубленный курс C/C++'),
Course(8, 'история искусств'),
Course(9, 'академическая живопись'),
Course(10, 'портретная фотография'),
Course(11, 'разработка на Go'),
Course(12, 'высоконагруженные сервисы'),
Course(13, 'аналитическая геометрия'),
Course(14, 'машинное обучение'),
Course(15, 'химия'),
]
```

Сотрудники

```
Preps = [
    Prep(1, 'Артамонов', 25000, 1),
    Prep(2, 'Петров', 35000, 2),
    Prep(3, 'Титов', 35000, 3),
    Prep(4, 'Митичев', 55000, 4),
    Prep(5, 'Гусев', 34000, 5),
    Prep(6, 'Иваненко', 45000, 6),
    Prep(7, 'Алиманов', 36000, 7),
    Prep(8, 'Иванов', 36000, 8),
    Prep(9, 'Иванин', 25000, 9),
    Prep(10, 'Семакин', 15000, 10),
    Prep(11, 'Лебедев', 29000, 11),
    Prep(12, 'Уткин', 25800, 12),
    Prep(13, 'Стрельцов', 22000, 13),
    Prep(14, 'Соболев', 50000, 14),
    Prep(15, 'Ларин', 42000, 15),
    Prep(16, 'Ларцов', 4000, 11),
    Prep(17, 'Огурцов', 34000, 1),
    Prep(18, 'Скворцов', 40300, 4),
    Prep(18, 'Солнцев', 12000, 1),
]
```

```
Preps_Courses = [
    PrepCourse(1,1),
    PrepCourse(2,2),
    PrepCourse(3,3),
    PrepCourse(4,4),
    PrepCourse(5,5),
    PrepCourse(6,6),
    PrepCourse(7,7),
    PrepCourse(8,8),
    PrepCourse(9,9),
    PrepCourse(10,10),
    PrepCourse(11,11),
    PrepCourse(12,12),
    PrepCourse(13,13),
]
```

```
PrepCourse(14,14),
PrepCourse(15,15),
PrepCourse(16,11),
PrepCourse(17,1),
PrepCourse(18,4),
]
```

```
def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(e.fio, e.sal, d.name)
                    for d in Courses
                    for e in Preps
                    if e.course_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_tPrep = [(d.name, ed.course_id, ed.prep_id)
                           for d in Courses
                           for ed in Preps_Courses
                           if d.id == ed.course_id]

    many_to_many = [(e.fio, e.sal, Course_name)
                     for Course_name, course_id, prep_id in many_to_many_tPrep
                     for e in Preps if e.id == prep_id]

    print('РК1 Выполнил Дюжев С.А. ИУ5Ц-73Б')
    print('Задание В1')
    res_b1 = []
    for i in one_to_many:
        if i[0][0] == "А":
            res_b1.append((i[0], i[2]))
    print(res_b1)

    print('Задание В2')
    res_b2_unsorted = []
    # Перебираем все курсы
    for d in Courses:
        # Список преподавателей курса
        d_Preps = list(filter(lambda i: i[2]==d.name, one_to_many))
        # Если курс не пустой
        if len(d_Preps) > 0:
            # Зарплаты преподавателей
            d_sals = [sal for _,sal,_ in d_Preps]
            # минимальная зарплата преподавателя
            d_sals_min = min(d_sals)
            res_b2_unsorted.append((d.name, d_sals_min))
```

```
# Сортировка по минимальной зарплате
res_b2 = sorted(res_b2_unsorted, key=itemgetter(1))
print(res_b2)

print('Задание B3')
res_b3 = []
for i in one_to_many:
    res_b3.append((i[0], i[2]))
res_b3_sorted = sorted(res_b3, key=itemgetter(0))
print(res_b3_sorted)

if __name__ == '__main__':
    main()
```

Результаты выполнения:

```
C:\Users\amart\Desktop\RIP_5sem>python rk1.py
PK1 Выполнил Дюжев С.А. ИУ5Ц-73Б
Задание B1
[('Артамонов', 'математический анализ'), ('Алиманов', 'углубленный курс C/C++')]
Задание B2
[('разработка на Go', 4000), ('математический анализ', 12000), ('портретная фотография', 15000), ('аналитическая геометрия', 22000), ('академическая живопись', 25000), ('высоконагруженные сервисы', 25800), ('литература 19 века', 34000), ('python3', 35000), ('Jango', 35000), ('углубленный курс C/C++', 36000), ('история искусств', 36000), ('алгебра', 40300), ('химия', 42000), ('подготовительная программа C/C++', 45000), ('машинное обучение', 50000)]
Задание B3
[('Алиманов', 'углубленный курс C/C++'), ('Артамонов', 'математический анализ'), ('Гусев', 'литература 19 века'), ('Иваненко', 'подготовительная программа C/C++'), ('Иванин', 'академическая живопись'), ('Иванов', 'история искусств'), ('Ларин', 'химия'), ('Ларцов', 'разработка на Go'), ('Лебедев', 'разработка на Go'), ('Митичев', 'алгебра'), ('Огурцов', 'математический анализ'), ('Петров', 'python3'), ('Семакин', 'портретная фотография'), ('Скворцов', 'алгебра'), ('Соболев', 'машинное обучение'), ('Солнцев', 'математический анализ'), ('Стрельцов', 'аналитическая геометрия'), ('Титов', 'Jango'), ('Уткин', 'высоконагруженные сервисы')]
```

C:\Users\amart\Desktop\RIP_5sem>python rk1.py

PK1 Выполнил Дюжев С.А. ИУ5Ц-73Б

Задание B1

[('Артамонов', 'математический анализ'), ('Алиманов', 'углубленный курс C/C++')]

Задание B2

[('разработка на Go', 4000), ('математический анализ', 12000), ('портретная фотография', 15000), ('аналитическая геометрия', 22000), ('академическая живопись', 25000), ('высоконагруженные сервисы', 25800), ('литература 19 века', 34000), ('python3', 35000), ('Jango', 35000), ('углубленный курс C/C++', 36000), ('история искусств', 36000), ('алгебра', 40300), ('химия', 42000), ('подготовительная программа C/C++', 45000), ('машинное обучение', 50000)]

Задание B3

[('Алиманов', 'углубленный курс C/C++'), ('Артамонов', 'математический анализ'), ('Гусев', 'литература 19 века'), ('Иваненко', 'подготовительная программа C/C++'), ('Иванин', 'академическая живопись'), ('Иванов', 'история искусств'), ('Ларин', 'химия'), ('Ларцов', 'разработка на Go'), ('Лебедев', 'разработка на Go'), ('Митичев', 'алгебра'), ('Огурцов', 'математический анализ'), ('Петров', 'python3'), ('Семакин', 'портретная фотография'), ('Скворцов', 'алгебра'), ('Соболев', 'машинное обучение'), ('Солнцев', 'математический анализ'), ('Стрельцов', 'аналитическая геометрия'), ('Титов', 'Jango'), ('Уткин', 'высоконагруженные сервисы')]