

- 1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
- 2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
- 3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Код программы.

```
from operator import itemgetter

class Table():
    def __init__(self, id, name, base_id, kol):
        self.id = id
        self.name = name
        self.base_id = base_id
        self.kol = kol

class Base():
    def __init__(self, id, name):
        self.id = id
        self.name = name

class TableBase():
    def __init__(self, base_id, table_id):
        self.base_id = base_id
        self.table_id = table_id

base: list[Base] = [
    Base(1, "Учебное заведение"),
    Base(2, "Магазин"),
    Base(3, "Социальная сеть"),
    Base(4, "Проект управления")
]

table: list[Table] = [
    Table(1, "Студенты", 1, 5),
    Table(2, "Курсы", 3, 4),
    Table(3, "Преподаватели", 2, 6),
    Table(4, "Товары", 2, 7),
    Table(5, "Заказы", 1, 5),
    Table(6, "Клиенты", 4, 3),
    Table(7, "Пользователи", 1, 8),
    Table(8, "Сообщения", 1, 4),
    Table(9, "Друзья", 3, 2),
    Table(10, "Проекты", 1, 6),
    Table(11, "Задачи", 2, 5),
    Table(12, "Сотрудники", 3, 4)
]

table_b = []
table_base = [
    TableBase(1, 3),
    TableBase(1, 9),
    TableBase(1, 5),
```

```

TableBase(1, 11),
TableBase(1, 7),
TableBase(2, 2),
TableBase(2, 8),
TableBase(2, 4),
TableBase(2, 6),
TableBase(2, 10),
TableBase(2, 12),
TableBase(3, 10),
TableBase(3, 1),
TableBase(3, 5),
TableBase(3, 7),
TableBase(3, 3),
TableBase(4, 12),
TableBase(4, 6),
TableBase(4, 9),
TableBase(4, 3)
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(t.name, t.kol, b.name)
                    for b in base
                    for t in table
                    if t.base_id == b.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(b.name, tb.table_id)
                           for b in base
                           for tb in table_base if tb.base_id == b.id]

    many_to_many = [(base_name, t.name, t.kol)
                     for base_name, table_id in many_to_many_temp
                     for t in table if t.id == table_id]

    print('Задание B1')
    searched_char = '3'
    res_11 = [el for el in one_to_many if el[0][0] == searched_char]
    print(*res_11, sep='\n')

    print('\nЗадание B2')
    res_12_unsorted = [[el.name, 1000000] for el in base]
    for t in res_12_unsorted:
        t[1] = min(map(lambda i: i[1], list(filter(lambda i: i[2] == t[0],
one_to_many))))

    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(*res_12, sep='\n')

    print('\nЗадание B3')
    res_13 = sorted(many_to_many, key=itemgetter(1), reverse=True)

    print(*res_13, sep='\n')

if __name__ == '__main__':
    main()

```

Вывод программы.

Задание B1

('Заказы', 5, 'Учебное заведение')

('Задачи', 5, 'Магазин')

Задание B2

['Магазин', 5]

['Учебное заведение', 4]

['Проект управления', 3]

['Социальная сеть', 2]

Задание B3

('Магазин', 'Товары', 7)

('Социальная сеть', 'Студенты', 5)

('Магазин', 'Сотрудники', 4)

('Проект управления', 'Сотрудники', 4)

('Магазин', 'Сообщения', 4)

('Магазин', 'Проекты', 6)

('Социальная сеть', 'Проекты', 6)

('Учебное заведение', 'Преподаватели', 6)

('Социальная сеть', 'Преподаватели', 6)

('Проект управления', 'Преподаватели', 6)

('Учебное заведение', 'Пользователи', 8)

('Социальная сеть', 'Пользователи', 8)

('Магазин', 'Курсы', 4)

('Магазин', 'Клиенты', 3)

('Проект управления', 'Клиенты', 3)

('Учебное заведение', 'Заказы', 5)

('Социальная сеть', 'Заказы', 5)

('Учебное заведение', 'Задачи', 5)

('Учебное заведение', 'Друзья', 2)

('Проект управления', 'Друзья', 2)

Process finished with exit code 0