

## Вариант Г31

### Таблица данных - База данных

1. «База данных» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех баз данных, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «База данных» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список баз данных с максимальной зарплатой сотрудников в каждом баз данных, отсортированный по максимальной зарплате.
3. «База данных» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и баз данных, отсортированный по базам данных, сортировка по сотрудникам произвольная.

### Текст программы

```
from operator import itemgetter
```

```
class DataTable:
```

```
    def __init__(self, id, name, size_mb, db_id):  
        self.id = id  
        self.name = name  
        self.size_mb = size_mb  
        self.db_id = db_id
```

```
class Database:
```

```
    def __init__(self, id, name):  
        self.id = id  
        self.name = name
```

```
class TableDatabase:

    def __init__(self, db_id, dt_id):

        self.db_id = db_id

        self.dt_id = dt_id
```

```
databases = [

    Database(1, 'AnalyticsDB'),

    Database(2, 'ProductsDB'),

    Database(3, 'ArchiveDB'),

    Database(4, 'OrdersDB')

]
```

```
data_tables = [

    DataTable(1, 'Users', 50, 1),

    DataTable(2, 'Products', 100, 2),

    DataTable(3, 'Orders', 200, 4),

    DataTable(4, 'Archive', 500, 3),

    DataTable(5, 'Addresses', 20, 4)

]
```

```
table_dbs = [

    TableDatabase(1, 1),

    TableDatabase(2, 2),

    TableDatabase(3, 4),

    TableDatabase(4, 3),
```

```
TableDatabase(5, 4),  
TableDatabase(2, 5),  
TableDatabase(3, 1),  
TableDatabase(4, 5)  
]
```

```
def main():  
    one_to_many = [(dt.name, dt.size_mb, db.name)  
                   for db in databases  
                   for dt in data_tables  
                   if dt.db_id == db.id]  
  
    many_to_many_temp = [(db.name, tdb.db_id, tdb.dt_id)  
                          for db in databases  
                          for tdb in table_dbs  
                          if db.id == tdb.db_id]  
  
    many_to_many = [(dt.name, dt.size_mb, db_name)  
                    for db_name, db_id, dt_id in many_to_many_temp  
                    for dt in data_tables if dt.id == dt_id]  
  
    print("\nЗадание Г1")
```

```
    dbs_starting_with_a = [db for db in one_to_many if (db[2][0] ==  
"A")]
```

```
    for db in dbs_starting_with_a:
```

```
        dt_in_db = [db[0] for db in dbs_starting_with_a]
```

```
        print(f"База данных: {db[2]}, Таблицы: {dt_in_db}")
```

```
print("\nЗадание Г2")
```

```
db_max_sizes = {}
```

```
for _, size, db_name in one_to_many:
```

```
    if db_name not in db_max_sizes:
```

```
        db_max_sizes[db_name] = 0
```

```
    db_max_sizes[db_name] = max(db_max_sizes[db_name], size)
```

```
    sorted_db_max_sizes = sorted(db_max_sizes.items(),  
key=itemgetter(1), reverse=True)
```

```
    print(sorted_db_max_sizes)
```

```
print("\nЗадание Г3")
```

```
sorted_many_to_many = sorted(many_to_many, key=itemgetter(2))
```

```
print(sorted_many_to_many)
```

```
if __name__ == '__main__':
```

```
    main()
```

## **Пример работы программы**

### **Задание Г1**

База данных: AnalyticsDB, Таблицы: ['Users', 'Archive']

База данных: ArchiveDB, Таблицы: ['Users', 'Archive']

### **Задание Г2**

[('ArchiveDB', 500), ('OrdersDB', 200), ('ProductsDB', 100),  
('AnalyticsDB', 50)]

### **Задание Г3**

[('Users', 50, 'AnalyticsDB'), ('Archive', 500, 'ArchiveDB'), ('Users', 50,  
'ArchiveDB'), ('Orders', 200, 'OrdersDB'), ('Addresses', 20, 'OrdersDB'),  
('Products', 100, 'ProductsDB'), ('Addresses', 20, 'ProductsDB')]