

## Файл main.py

```
1  from collections import Counter
2
3  class Detail:
4      def __init__(self, id, name, supplier_id):
5          self.id = id
6          self.name = name
7          self.supplier_id = supplier_id
8
9  class Supplier:
10     def __init__(self, id, name, department):
11         self.id = id
12         self.name = name
13         self.department = department
14
15     class SupplierDetailLink:
16         def __init__(self, supplier_id, detail_id):
17             self.supplier_id = supplier_id
18             self.detail_id = detail_id
19
20     def merge_one_to_many(details, suppliers):
21         return [(supplier.name, detail.name, supplier.department)
22                 for detail in details
23                 for supplier in suppliers
24                 if detail.supplier_id == supplier.id]
25
26     def merge_many_to_many(suppliers, supplier_detail_links, details):
27         many_to_many_temp = [(supplier.name, sdl.supplier_id, sdl.detail_id)
28                               for supplier in suppliers
29                               for sdl in supplier_detail_links
30                               if supplier.id == sdl.supplier_id]
31
32         many_to_many = [(supplier.id, detail.name)
33                          for supplier in suppliers
34                          for detail in details
35                          if detail.supplier_id == supplier.id]
36
37         return many_to_many_temp, many_to_many
38
39     def query_1_results(data):
40         return sorted(data, key=lambda x: x[0])
41
42     def query_2_results(suppliers):
43         department_counts = Counter(supplier.department for supplier in suppliers)
44         return sorted(department_counts.items(), key=lambda x: x[1], reverse=True)
```

```

def query_3_results(suppliers):
    return [(supplier.name, supplier.department) for supplier in suppliers if supplier.name.endswith("ов")]

def main():
    """Main function"""

    details = [
        Detail(1, "Гвоздь", 1),
        Detail(2, "Пробка", 2),
        Detail(3, "Шуруп", 2),
        Detail(4, "Замок", 3),
        Detail(5, "Призма", 1)
    ]

    suppliers = [
        Supplier(1, "Иванов", "ПИЛАТОС"),
        Supplier(2, "Петров", "Замок или замок?"),
        Supplier(3, "Сидоренко", "ГВОЗДИКА"),
        Supplier(4, "Смирный", "ГВОЗДИКА"),
        Supplier(5, "Козуб", "ПИЛАТОС"),
    ]

    supplier_detail_links = [
        SupplierDetailLink(1, 1),
        SupplierDetailLink(2, 2),
        SupplierDetailLink(2, 3),
        SupplierDetailLink(4, 2),
        SupplierDetailLink(5, 1)
    ]

    # Query 1
    result_query_1 = query_1_results(merge_one_to_many(details, suppliers))
    print_query_results("Задание №1", result_query_1, "Поставщик: {item[0]}, Деталь: {item[1]}, Отдел: {item[2]}")

    # Query 2
    result_query_2 = query_2_results(suppliers)
    print_query_results("Задание №2", result_query_2, "Отдел: {item[0]}, Количество поставщиков: {item[1]}")

    # Query 3
    result_query_3 = query_3_results(suppliers)
    print_query_results("Задание №3", result_query_3, "Поставщик: {item[0]}, Отдел: {item[1]}")

def print_query_results(title, result, format_string):
    print(f"\n{title}:")
    for item in result:
        print(format_string.format(item=item))

if __name__ == "__main__":
    main()

```

## Файл test\_main.py

```
1 import unittest
2 from main import Detail, Supplier, SupplierDetailLink, merge_one_to_many, query_2_results, query_3_results
3
4 class TestMainFunctions(unittest.TestCase):
5
6     def setUp(self):
7         self.details = [
8             Detail(1, "Гвоздь", 1),
9             Detail(2, "Пробка", 2),
10            Detail(3, "Шуруп", 2),
11            Detail(4, "Замок", 3),
12            Detail(5, "Призма", 1)
13        ]
14
15        self.suppliers = [
16            Supplier(1, "Иванов", "ПИЛАТОС"),
17            Supplier(2, "Петров", "Замок или замок?"),
18            Supplier(3, "Сидоренко", "ГВОЗДИКА"),
19            Supplier(4, "Смирный", "ГВОЗДИКА"),
20            Supplier(5, "Козуб", "ПИЛАТОС"),
21        ]
22
23        self.supplier_detail_links = [
24            SupplierDetailLink(1, 1),
25            SupplierDetailLink(2, 2),
26            SupplierDetailLink(2, 3),
27            SupplierDetailLink(4, 2),
28            SupplierDetailLink(5, 1)
29        ]
30
31    def test_merge_one_to_many(self):
32        result = merge_one_to_many(self.details, self.suppliers)
33        expected_result = [
34            ('Иванов', 'Гвоздь', 'ПИЛАТОС'),
35            ('Петров', 'Пробка', 'Замок или замок?'),
36            ('Петров', 'Шуруп', 'Замок или замок?'),
37            ('Смирный', 'Пробка', 'ГВОЗДИКА'),
38            ('Козуб', 'Гвоздь', 'ПИЛАТОС')
39        ]
40        self.assertEqual(result, expected_result)
41
42    def test_query_2_results(self):
43        result = query_2_results(self.suppliers)
44        expected_result = [('ГВОЗДИКА', 2), ('Замок или замок?', 2), ('ПИЛАТОС', 2)]
45
46        self.assertEqual(result, expected_result)
47
48    def test_query_3_results(self):
49        result = query_3_results(self.suppliers)
50        expected_result = [('Иванов', 'ПИЛАТОС'), ('Смирный', 'ГВОЗДИКА'), ('Козуб', 'ПИЛАТОС')]
51        self.assertEqual(result, expected_result)
52
53 if __name__ == '__main__':
54     unittest.main()
```