

Отчет
Рубежный контроль №1
Вариант 12. Запрос А.

Студент: Крючков Д.И.
Группа: ИБМ3-34Б
Преподаватель: Гапанюк Ю.Е.
Кафедра: ИУ-5

Вариант А.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с суммарной зарплатой сотрудников в каждом отделе, отсортированный по суммарной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.

12

Язык программирования

Средство разработки

Код программы

```
# РК 1 ПИКЯП

# Классы
# Класс средство разработки
class DevTool:
    def __init__(self, id, title):
        self.id = id
        self.title = title

# Класс язык программирования
class ProgrammingLanguage:
    def __init__(self, id, name, vacancies_count, primary_tool_id):
        self.id = id
        self.name = name
        self.vacancies_count = vacancies_count # количественный признак
        self.primary_tool_id = primary_tool_id # внешний ключ на DevTool_id

# Класс для упразднения ММ
class LangTool:
    def __init__(self, lang_id, tool_id):
        self.lang_id = lang_id
        self.tool_id = tool_id

# Базовые данные
# Средства разработки
tools = [
    DevTool(1, "Visual Studio Code"),
    DevTool(2, "PyCharm"),
    DevTool(3, "IntelliJ IDEA"),
    DevTool(4, "Xcode")
]
# Языки программирования
languages = [
    ProgrammingLanguage(1, "Python", 950, 2), # основной инструмент: PyCharm
    ProgrammingLanguage(2, "Java", 1100, 3), # основной инструмент:
```

```

IntelliJ IDEA
    ProgrammingLanguage(3, "C#", 700, 1), # основной инструмент: VS
Code (условно)
    ProgrammingLanguage(4, "Swift", 400, 4), # основной инструмент:
Xcode
    ProgrammingLanguage(5, "JavaScript", 1300, 1) # основной инструмент: VS
Code
]

# Связи M<-->M: язык поддерживается разными средствами разработки
lang_tools = [
    # Python
    LangTool(1, 1), LangTool(1, 2),
    # Java
    LangTool(2, 1), LangTool(2, 3),
    # C#
    LangTool(3, 1),
    # Swift
    LangTool(4, 4),
    # JavaScript
    LangTool(5, 1), LangTool(5, 3),
]

# Запросы к нашим "данным"

# Запрос №1, здесь мы выводим пары средство разработки- язык программирования
# сортируя по средствам (связь одно средство-много языков)
tools_sorted = sorted(tools, key=lambda t: t.title.lower())

print("ЗАПРОС 1: 'Средство разработки' → все связанные 'Языки' (1→M):")
for tool in tools_sorted:
    # соберём все языки, у которых primary_tool_id совпадает с текущим
    tool_id
    langs_for_tool = []
    for lang in languages:
        if lang.primary_tool_id == tool.id:
            langs_for_tool.append(lang.name)

    langs_for_tool.sort()
    if langs_for_tool:
        print(f" {tool.title}: {' '.join(langs_for_tool)}")
    else:
        print(f" {tool.title}: -") # явный маркер, что языков нет

# Запрос №2: я ввел для каждого языка условное количество вакансий,
# здесь для каждого средства разработки будем выводить суммарное количество
# вакансий для поддерживаемых языков

vacancies_by_tool_m2m = []

for tool in tools:

    lang_ids_set = []
    for lt in lang_tools:
        if lt.tool_id == tool.id:
            # добавим, если ещё не было, чтобы избежать двойного учёта
            if lt.lang_id not in lang_ids_set:
                lang_ids_set.append(lt.lang_id)

    # суммируем вакансии по найденным языкам
    total = 0
    for lang_id in lang_ids_set:

```

```
for lang in languages:
    if lang.id == lang_id:
        total += lang.vacancies_count

vacancies_by_tool_m2m.append((tool.title, total))

# сортировка по сумме по убыванию
vacancies_by_tool_m2m.sort(key=lambda x: x[1], reverse=True)

print("\nЗАПРОС 2: (M→M) Суммарное количество вакансий по каждому средству разработки:")
for tool_name, total in vacancies_by_tool_m2m:
    print(f" {tool_name}: {total}")

# Запрос №3, здесь будем искать средства разработки у которых в названии есть
# ключевое слово(я взял "code")
# и выводить поддерживаемые ими языки

print("\nЗАПРОС 3: Средства с 'code' в названии и языки (M→M через LangTool):")

for tool in tools:
    if "code" in tool.title.lower():
        langs = []
        for lt in lang_tools:
            if lt.tool_id == tool.id:
                # найти язык по id
                for lang in languages:
                    if lang.id == lt.lang_id:
                        langs.append(lang.name)
        # убрать дубликаты и отсортировать
        unique_sorted_langs = sorted(list(set(langs)))
        print(f" {tool.title}: {'.'.join(unique_sorted_langs)} if
unique_sorted_langs else '-')")
```

Результат выполнения программы

ЗАПРОС 2: (МкМ) Суммарное количество вакансий по каждому средству разработки:

Visual Studio Code: 4050

IntelliJ IDEA: 2400

PyCharm: 950

Xcode: 400

ЗАПРОС 3: Средства с 'code' в названии и языки (МкМ через LangTool):

Visual Studio Code: C#, Java, JavaScript, Python

Xcode: Swift

ЗАПРОС 1: 'Средство разработки' → все связанные 'Языки' (1→M):

IntelliJ IDEA: Java

PyCharm: Python

Visual Studio Code: C#, JavaScript

Xcode: Swift