

ИУ5-55 Турчин Денис. РК-1. 12.10.2021.

```
# ИУ5-55 Турчин Денис. РК-1. 12.10.2021. Предметная область 14: CD-диск (класс-1) и Библиотека CD-дисков (класс-2). Запросы Д:
#
# «Библиотека CD-дисков» и «CD-диск» связаны соотношением один-ко-многим.
# Выведите список всех CD-дисков, у которых название хранимой записи содержит "Терминатор", и названия Библиотек, в которых они хранятся.
# «Библиотека CD-дисков» и «CD-диск» связаны соотношением один-ко-многим.
# Выведите список Библиотек CD-дисков со средним объемом записи, отсортированный по среднему объему.
# (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).
# «Библиотека CD-дисков» и «CD-диск» связаны соотношением многие-ко-многим.
# Выведите список всех библиотек, у которых название начинается с буквы «А», и список хранящихся в них CD-дисков.
#
```

Создам файл с описанием моделей:

```
rk-1 > models.py > ...
1 class CD_disk:
2     """CD-диск"""
3     def __init__(self, id, title, data_format, data_size_mb, lib_id):
4         self.id = id
5         self.title = title
6         self.data_format = data_format
7         self.data_size_mb = data_size_mb
8
9         self.lib_id = lib_id
10
11     def __repr__(self):
12         return f"CD-диск {self.title}"
13
14
15 class CD_disk_lib:
16     """Библиотека CD-дисков"""
17     def __init__(self, id, name):
18         self.id = id
19         self.name = name
20
21     def __repr__(self):
22         return "Библиотека CD-дисков"
23
24
25 class Disk_lib__disk__matching_table:
26     """
27     'CD-диски в Библиотеке CD-дисков отдела' для реализации
28     связи многие-ко-многим
29     """
30     def __init__(self, lib_id, disk_id):
31         self.disk_id = disk_id
32         self.lib_id = lib_id
```

Создам второй файл с тестовой базой данных с таблицами согласно модели.

```
1  from models import CD_disk, CD_disk_lib, Disk_lib_disk_matching_table
2  |
3  libs = [
4      CD_disk_lib(1, 'Сборник фильмов Леши Иванова'),
5      CD_disk_lib(2, 'ISO-образы ОС и программ сисадмина ООО "Храню диски"'),
6      CD_disk_lib(3, 'Архив репортажей ВГТРК 2q2005'),
7      CD_disk_lib(11, 'Игры Ubisoft'),
8      CD_disk_lib(22, 'Архив музея CD-дисков и доисторических устройств хранения данных'),
9      CD_disk_lib(33, 'Список дисков, уничтоженных дисководом моего ноутбука'),
10 ]
11
12 disks = [
13     CD_disk(1, 'FarCry 6 (2021)', 'iso', 700, 11),
14     CD_disk(2, 'Запись концерта "Владимир Путин молодец"', 'mp4', 350, 3),
15     CD_disk(3, 'Терминатор 4: Да придёт спаситель (2009)', 'mp4', 450, 1),
16     CD_disk(4, 'Терминатор: Тёмные судьбы мексиканца (2019)', 'mp4', 656, 1),
17     CD_disk(11, 'Терминатор (1984)', 'mp4', 656, 22),
18     CD_disk(10, 'Бабушка легкого поведения 2 (2021)', 'mp4', 620, 1),
19     CD_disk(5, 'Assassin's Creed 3 (2008)', 'iso', 650, 11),
20     CD_disk(6, 'Урок химии. Сера (1978)', 'mp4', 50, 22),
21     CD_disk(7, 'Windows 98', 'iso', 450, 2),
22     CD_disk(8, 'Arch Linux', 'iso', 325, 2),
23     CD_disk(9, 'GTA San Andreas', 'iso', 125, 2),
24 ]
```

```
26  libs_disks_matching_table = [
27      Disk_lib_disk_matching_table(1, 3),
28      Disk_lib_disk_matching_table(1, 4),
29      Disk_lib_disk_matching_table(1, 10),
30
31      Disk_lib_disk_matching_table(2, 1),
32      Disk_lib_disk_matching_table(2, 5),
33      Disk_lib_disk_matching_table(2, 7),
34      Disk_lib_disk_matching_table(2, 8),
35      Disk_lib_disk_matching_table(2, 9),
36
37      Disk_lib_disk_matching_table(3, 2),
38
39      Disk_lib_disk_matching_table(11, 1),
40      Disk_lib_disk_matching_table(11, 5),
41
42      Disk_lib_disk_matching_table(22, 11),
43      Disk_lib_disk_matching_table(22, 6),
44      Disk_lib_disk_matching_table(22, 3),
45      Disk_lib_disk_matching_table(22, 7),
46      Disk_lib_disk_matching_table(22, 9),
47
48      # no disks in 33
49  ]
```

Важно относится к таблицам на первом скриншоте как к самостоятельной БД, т.к. связь многие-ко-многим, описанная таблицей на втором скриншоте значительно меняет модель, что приводит к нарушению принципа согласованности ACID и денормализации модели в том плане, что ограничение на связь диска с библиотекой, где он хранится, становится бесполезным.

Запрос Д-1 можно описать на sql следующим образом:

```
23     print('Задание Д-1')
24     # select * from tables.disks as d where name contains 'Терминатор'
25     # join tables.libs as l on d.lib_id = l.id
```

Запрос Д-2:

```
34     print('\nЗадание Д-2')
35     # select l.name, avg(d.size) as avg_size from tables.disks as d
36     # join tables.libs as l on l.id = d.lib_id
37     # group by l.name order by avg(d.size) desc
```

Запрос Д-3:

```
59     print('\nЗадание А3')
60     # select l.name, d.id, d.title, d.data_format from tables.libs
61     # join tables.libs__disks__matching_table as mt on mt.id_lib = l.id
62     # join tables.disks as d on mt.id_lib = d.id
63
```

Чтобы реализовать задуманное средствами языка Python без сторонних библиотек понадобятся внутренние таблицы:

- Для первых двух запросов:

```
16     # Соединение данных один-ко-многим
17     disks_loop_join_libs = [{'disks': d, 'libs': l}
18                             for d in disks
19                             for l in libs
20                             if d.lib_id == l.id]
21
```

- Для третьего:

```
68     # Соединение данных многие-ко-многим
69     libs_loop_join_mt_loop_join_disks = [{'libs': l, 'mt': mt, 'disks': d}
70                                           for l in libs
71                                           for mt in libs__disks__matching_table
72                                           for d in disks
73                                           if d.id == mt.disk_id
74                                           if l.id == mt.lib_id]
75
```

* loop-join – join перебором без оптимизаций

Остается составить сами запросы:

- Д-1:

```

23     print('Задание Д-1')
24     # select * from tables.disks as d where name contains 'Терминатор'
25     # join tables.libs as l on d.lib_id = l.id
26     D1 = [(x['disks'].id, x['disks'].title, x['disks'].data_format, x['disks'].data_size_mb)
27           for x in disks_loop_join_libs
28           if x['disks'].title.find('Терминатор') != - 1
29           ]

```

Результат:

```

Задание Д-1
(3, 'Терминатор 4: Да придёт спаситель (2009)', 'mp4', 450)
(4, 'Терминатор: Тёмные судьбы мексиканца (2019)', 'mp4', 656)
(11, 'Терминатор (1984)', 'mp4', 656)

```

- Д-2:

```

34     print('\nЗадание Д-2')
35     # select l.name, avg(d.size) as avg_size from tables.disks as d
36     # join tables.libs as l on l.id = d.lib_id
37     # group by l.name order by avg(d.size) desc
38
39     # подкапотная таблица вида: {lib_title: {sum, count}}
40     title_sum_count_dict = {}
41     for disks_libs_row in disks_loop_join_libs:
42         lib_title = disks_libs_row['libs'].name
43         disk_size = disks_libs_row['disks'].data_size_mb
44
45         if lib_title in title_sum_count_dict:
46             title_sum_count_dict[lib_title]['sum'] = title_sum_count_dict[lib_title]['sum'] + disk_size
47             title_sum_count_dict[lib_title]['count'] = title_sum_count_dict[lib_title]['count'] + 1
48         else:
49             title_sum_count_dict[lib_title] = {'sum': disk_size, 'count': 1}
50
51     D2 = sorted(
52         [(lib_title, title_sum_count_dict[lib_title]['sum'] / title_sum_count_dict[lib_title]['count'])
53          for lib_title in title_sum_count_dict
54          if title_sum_count_dict[lib_title]['count'] != 0
55          ],
56         key=itemgetter(1), reverse=True
57     )

```

Результат:

```

Задание Д-2
('Игры Ubisoft', 675.0)
('Сборник фильмов Леши Иванова', 575.3333333333334)
('Архив музея CD-дисков и доисторических устройств хранения данных', 353.0)
('Архив репортажей ВГТРК 2q2005', 350.0)
('ISO-образы ОС и программ сисадмина ООО "Храню диски"', 300.0)

```

- Д-3:

```

77     D3 = [(x['libs'].name, x['disks'].id, x['disks'].title, x['disks'].data_format, x['disks'].data_size_mb)
78           for x in libs_loop_join_mt_loop_join_disks
79           ]

```

Результат:

Задание АЗ

('Сборник фильмов Леши Иванова', 3, 'Терминатор 4: Да придёт спаситель (2009)', 'mp4', 450)
('Сборник фильмов Леши Иванова', 4, 'Терминатор: Тёмные судьбы мексиканца (2019)', 'mp4', 656)
('Сборник фильмов Леши Иванова', 10, 'Бабушка легкого поведения 2 (2021)', 'mp4', 620)
('ISO-образы ОС и программ сисадмина ООО "Храню диски"', 1, 'FarCry 6 (2021)', 'iso', 700)
('ISO-образы ОС и программ сисадмина ООО "Храню диски"', 5, "Assassin's Creed 3 (2008)", 'iso', 650)
('ISO-образы ОС и программ сисадмина ООО "Храню диски"', 7, 'Windows 98', 'iso', 450)
('ISO-образы ОС и программ сисадмина ООО "Храню диски"', 8, 'Arch Linux', 'iso', 325)
('ISO-образы ОС и программ сисадмина ООО "Храню диски"', 9, 'GTA San Andreas', 'iso', 125)
('Архив репортажей ВГТРК 2q2005', 2, 'Запись концерта "Владимир Путин молодец"', 'mp4', 350)
('Игры Ubisoft', 1, 'FarCry 6 (2021)', 'iso', 700)
('Игры Ubisoft', 5, "Assassin's Creed 3 (2008)", 'iso', 650)
('Архив музея CD-дисков и доисторических устройств хранения данных', 11, 'Терминатор (1984)', 'mp4', 656)
('Архив музея CD-дисков и доисторических устройств хранения данных', 6, 'Урок химии. Сера (1978)', 'mp4', 50)
('Архив музея CD-дисков и доисторических устройств хранения данных', 3, 'Терминатор 4: Да придёт спаситель (2009)', 'mp4', 450)
('Архив музея CD-дисков и доисторических устройств хранения данных', 7, 'Windows 98', 'iso', 450)
('Архив музея CD-дисков и доисторических устройств хранения данных', 9, 'GTA San Andreas', 'iso', 125)