

Задача Python – иерархическое получение данных

Формулировка задачи	1
Цели.....	1
Требования к выполнению задачи.....	1
JSON файл.....	2

Формулировка задачи

Необходимо импортировать данные по организациям и пользователям в SQL таблицу из приложенного к заданию json файла.

У компании есть офисы (type = 1) в нескольких городах, в каждом из них есть отделы (type = 2), где работают сотрудники (type = 3). Необходимо реализовать выборку всех сотрудников по указанному идентификатору сотрудника.

Пример тестовых данных:

Id	ParentId	Name	Type
1	NULL	Офис в Санкт-Петербурге	1
2	1	Отдел разработки	2
3	2	Иванов	3
4	2	Сидоров	3
5	1	Отдел тестирования	2
6	5	Петров	3
7	NULL	Офис в Москве	1
8	7	Аналитический отдел	2
9	8	Винтиков	3
10	8	Шпунтиков	3
11	7	Отдел продаж	2
12	11	Отдел обслуживания корпоративных клиентов	2
13	12	Белова	3
14	12	Крылова	3
16	11	Отдел обслуживания физ лиц	2
17	16	Петрова	3
18	16	Иванова	3
19	7	Тех. поддержка	2
20	19	Морозов	3

На вход утилите подается идентификатор сотрудника = 3

На выходе получается список сотрудников в офисе **Санкт-Петербурга: Иванов, Сидоров, Петров.**

Цели

1. Продемонстрировать работоспособность поставленной задачи

Требования к выполнению задачи

1. Задача выполняется на Python в виде консольного приложения
2. Оформление кода должно быть выполнено согласно PEP8

3. Для каждого метода/процедуры необходимо писать комментарий
4. В качестве БД можно использовать PostgreSQL 9.6 / 10
5. Для работы с SQL нельзя использовать любые ORM решения
6. Решение должно состоять из:
 - a. документа описывающего текущее решение
 - b. документа описывающего текущую сложность выборки SQL и перспективы оптимизации;
 - c. исходных файлов (которые можно выполнить и посмотреть на работу программы);
 - d. сторонних библиотек, которые нужны для работы.

JSON файл

```
[{
  "id": 1,
  "ParentId": null,
  "Name": "Офис в Санкт-Петербурге",
  "Type": 1
},
{
  "id": 2,
  "ParentId": 1,
  "Name": "Отдел разработки",
  "Type": 2
},
{
  "id": 3,
  "ParentId": 2,
  "Name": "Иванов",
  "Type": 3
},
{
  "id": 4,
  "ParentId": 2,
  "Name": "Сидоров",
  "Type": 3
},
{
  "id": 5,
  "ParentId": 1,
  "Name": "Отдел тестирования",
  "Type": 2
},
{
  "id": 6,
  "ParentId": 5,
  "Name": "Петров",
  "Type": 3
},
{
  "id": 7,
  "ParentId": null,
  "Name": "Офис в Москве",
  "Type": 1
},
]
```

```
{
  "id": 8,
  "ParentId": 7,
  "Name": "Аналитический отдел",
  "Type": 2
},
{
  "id": 9,
  "ParentId": 8,
  "Name": "Винтиков",
  "Type": 3
},
{
  "id": 10,
  "ParentId": 8,
  "Name": "Шпунтиков",
  "Type": 3
},
{
  "id": 11,
  "ParentId": 7,
  "Name": "Отдел продаж",
  "Type": 2
},
{
  "id": 12,
  "ParentId": 11,
  "Name": "Отдел обслуживания корпоративных клиентов",
  "Type": 2
},
{
  "id": 13,
  "ParentId": 12,
  "Name": "Белова",
  "Type": 3
},
{
  "id": 14,
  "ParentId": 12,
  "Name": "Крылова",
  "Type": 3
},
{
  "id": 16,
  "ParentId": 11,
  "Name": "Отдел обслуживания физ лиц",
  "Type": 2
},
{
  "id": 17,
  "ParentId": 16,
  "Name": "Петрова",
  "Type": 3
},
{
  "id": 18,
  "ParentId": 16,
```

```
"Name": "Иванова",  
"Type": 3  
},  
{  
  "id": 19,  
  "ParentId": 7,  
  "Name": "Тех. поддержка",  
  "Type": 2  
},  
{  
  "id": 20,  
  "ParentId": 19,  
  "Name": "Морозов",  
  "Type": 3  
}]
```