

Разработка документной базы данных

Цель работы

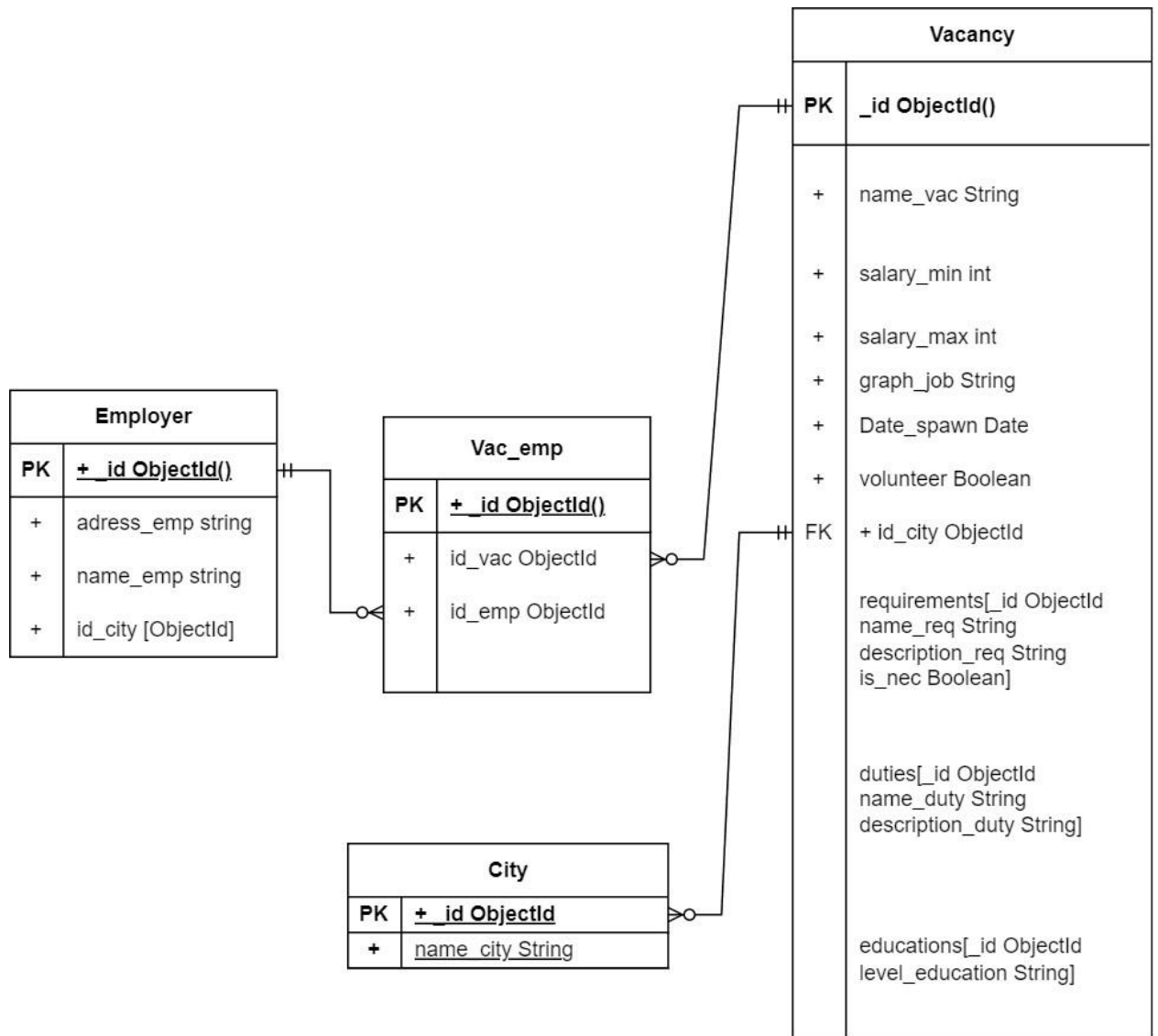
- 1) Спроектировать структуру json файла, соответствующую предметной области по варианту задания.
Хотя бы в одной коллекции документы должны иметь вложенные.
- 2) Создать базу MongoDB с тестовыми данными для запросов из варианта задания.

Вариант 12

вакансии: волонтерские позиции, название вакансии, организация работодатель, адрес работодателя, диапазон зарплаты, требования к образованию, Обязанности, график работы, требования обязательные, желательные, дата выставления вакансии

- а. вакансии, имеющие в названии SQL, но не заканчивающиеся на него
- б. работодатели в Санкт-Петербурге, выставившие несколько вакансий
- в. вакансии с наибольшей зарплатой
- г. волонтерские позиции с максимальным количеством требований
- д. вакансии, в которых нет требования к опыту работы

1. Физическая модель:



2. Структура json файла:

```

{
  City: {
    "_id": ObjectId()
    "name_city":
  }

  Vacancy: {
    "_id": ObjectId()
    "name_vac":
    "salary_min":
    "salary_max":
    "graph_job":
    "date_spawn":
    "volunteer":
    "id_city": ObjectId
    "requirements": [{"_id": ObjectId,
                      "name_req":
                      "description_req":
                      "is_nec":}]
  }
}
  
```

```

        "duties": [{ "_id": ObjectId,
                        "name_duty":
                        "description_duty": }]
        "educations": [{ "_id": ObjectId,
                            "level_education":}]
    }

    Employer: {
        "_id": ObjectId()
        "name_emp":
        "adress_emp":
        "id_city": [ObjectId]
    }

    Vac_emp: {
        "_id": ObjectId()
        "id_vac": ObjectId
        "id_emp": ObjectId
    }
}

```

3. Создание базы MongoDB с тестовыми данными для запросов из варианта задания.

```
idSpb = ObjectId()
```

```
db.City.insertOne({"_id": idSpb, "name_city": "Санкт-Петербург"})
```

```
idMsc = ObjectId()
```

```
db.City.insertOne({"_id": idMsc, "name_city": "Москва"})
```

```
idPsc = ObjectId()
```

```
db.City.insertOne({"_id": idPsc, "name_city": "Псков"})
```

```
idEmpSt = ObjectId()
```

```
idEmpOm = ObjectId()
```

```
idEmpRe = ObjectId()
```

```
idEmpSp = ObjectId()
```

```
db.Employer.insertMany([{"_id": idEmpSt, "name_emp": "Stole", "adress_emp": "Ул.
Казанская, д. 5", "id_city": [idSpb, idMsc]},
```

```
                        {"_id": idEmpOm, "name_emp": "Omega",
```

```
"adress_emp": "Ул. Ленина, д. 7", "id_city": [idMsc]},
```

```
                        {"_id": idEmpRe, "name_emp": "Reset",
```

```
"adress_emp": "Ул. Псковская, д. 23", "id_city": [idPsc]},
```

```
        {"_id": idEmpSp, "name_emp": "Spb here",  
"adress_emp": "Ул. Советская, д. 50", "id_city": [idSpb]}})
```

```
idVacSQL = ObjectId()
```

```
db.Vacancy.insertOne({"_id": idVacSQL, "name_vac": "Разработчик SQL БД",  
"salary_min": 60000,  
"salary_max": 80000,  
"graph_job": "5/2",  
"date_spawn": Date(),  
"volunteer": false,  
"id_city": idSpb,  
"requirements": [{"_id": ObjectId(), "name_req": "Опрятность", "description_req":  
"Необходимо иметь опрятный вид", "is_nec": false},  
{"_id": ObjectId(), "name_req": "Пунктуальность", "description_req": "Необходимо  
всегда приходить вовремя", "is_nec": true}],  
"duties": [{"_id": ObjectId(), "name_duty": "Разработка баз данных", "description_duty":  
"Необходимо разрабатывать реляционные базы данных на языке MySQL"}],  
"educations": [{"_id": ObjectId(), "level_education": "Высшее образование"}]  
}))
```

```
db.Vac_emp.insertOne({"_id": ObjectId(), "id_vac": idVacSQL, "id_emp": idEmpSt}))
```

```
idVacDriver = ObjectId()
```

```
db.Vacancy.insertOne({"_id": idVacDriver, "name_vac": "Водитель",  
"salary_min": 30000,  
"salary_max": 40000,  
"graph_job": "5/2",  
"date_spawn": Date(),  
"volunteer": false,  
"id_city": idMsc,  
"requirements": [{"_id": ObjectId(), "name_req": "Опрятность", "description_req":  
"Необходимо иметь опрятный вид", "is_nec": false},  
{"_id": ObjectId(), "name_req": "Пунктуальность", "description_req": "Необходимо  
всегда приходить вовремя", "is_nec": true},  
{"_id": ObjectId(), "name_req": "Водительские права", "description_req":  
"Необходимо иметь водительские права категории В", "is_nec": true}],
```

```

"duties": [{"_id": ObjectId(), "name_duty": "Управление автомоилем", "description_duty":
"Необходимо возить человека"}],
"educations": [{"_id": ObjectId(), "level_education": "Без образования"}]
})
db.Vac_emp.insertOne({"_id": ObjectId(), "id_vac": idVacDriver, "id_emp": idEmpOm})

```

```

idVacTeach = ObjectId()
db.Vacancy.insertOne({"_id": idVacTeach, "name_vac": "Преподаватель SQL",
"salary_min": 40000,
"salary_max": 50000,
"graph_job": "5/2",
"date_spawn": Date(),
"volunteer": false,
"id_city": idSpb,
"requirements": [{"_id": ObjectId(), "name_req": "Опрятность", "description_req":
"Необходимо иметь опрятный вид", "is_nec": false},
{"_id": ObjectId(), "name_req": "Пунктуальность", "description_req": "Необходимо
всегда приходить вовремя", "is_nec": true},
{"_id": ObjectId(), "name_req": "Уверенные знания", "description_req": "Необходимо
хорошо разбираться в SQL", "is_nec": true},
{"_id": ObjectId(), "name_req": "Опыт работы", "description_req": "Опыт работы
более двух лет", "is_nec": true}],
"duties": [{"_id": ObjectId(), "name_duty": "Обучение SQL", "description_duty":
"Обучение студентов SQL"}],
"educations": [{"_id": ObjectId(), "level_education": "Высшее образование"}]
})
db.Vac_emp.insertOne({"_id": ObjectId(), "id_vac": idVacTeach, "id_emp": idEmpSt})

```

```

idVacClean = ObjectId()
db.Vacancy.insertOne({"_id": idVacClean, "name_vac": "Уборщик",
"salary_min": 25000,
"salary_max": 30000,
"graph_job": "2/2",
"date_spawn": Date(),

```

```

"volunteer": false,
"id_city": idSpb,
"requirements": [{"_id": ObjectId(), "name_req": "Опрятность", "description_req":
"Необходимо иметь опрятный вид", "is_nec": false},
{"_id": ObjectId(), "name_req": "Пунктуальность", "description_req": "Необходимо
всегда приходить вовремя", "is_nec": true}],
"duties": [{"_id": ObjectId(), "name_duty": "Уборка помещений", "description_duty":
"Влажная и сухая уборка помещений"}],
"educations": [{"_id": ObjectId(), "level_education": "Без образования"}]
})
db.Vac_emp.insertOne({"_id": ObjectId(), "id_vac": idVacClean, "id_emp": idEmpSp})

```

```

idVacHelp = ObjectId()
db.Vacancy.insertOne({"_id": idVacHelp, "name_vac": "Волонтер гуманитарной
помощи",
"salary_min": 0,
"salary_max": 0,
"graph_job": "1/2",
"date_spawn": Date(),
"volunteer": true,
"id_city": idSpb,
"requirements": [{"_id": ObjectId(), "name_req": "Опрятность", "description_req":
"Необходимо иметь опрятный вид", "is_nec": false}],
"duties": [{"_id": ObjectId(), "name_duty": "Разнос гум. помощи по квартирам",
"description_duty": "Доставка до нуждающихся еды и воды"}],
"educations": [{"_id": ObjectId(), "level_education": "Без образования"}]
})
db.Vac_emp.insertOne({"_id": ObjectId(), "id_vac": idVacClean, "id_emp": idEmpSp})

```

```

idVacVol = ObjectId()
db.Vacancy.insertOne({"_id": idVacVol, "name_vac": "Волонтер на хоккейную арену",
"salary_min": 0,
"salary_max": 0,
"graph_job": "2/2",
"date_spawn": Date(),

```

```

"volunteer": true,
"id_city": idPsc,
"requirements": [{"_id": ObjectId(), "name_req": "Опрятность", "description_req":
"Необходимо иметь опрятный вид", "is_nec": true},
{"_id": ObjectId(), "name_req": "Умение разговаривать с людьми", "description_req":
"Необходимо постоянно общаться с людьми", "is_nec": true},
{"_id": ObjectId(), "name_req": "Возраст", "description_req": "До 40 лет", "is_nec":
true}],
"duties": [{"_id": ObjectId(), "name_duty": "Помощь с навигацией", "description_duty":
"Подсказывать людям верное направление"}],
"educations": [{"_id": ObjectId(), "level_education": "Без образования"}]
})
db.Vac_emp.insertOne({"_id": ObjectId(), "id_vac": idVacVol, "id_emp": idEmpRe})

```

4. Набор тестовых данных

Текст запроса	Данные удовлетворяющие условиям запросов	Данные не удовлетворяющие условиям запросов
а. вакансии, имеющие в названии SQL, но не заканчивающиеся на него	Коллекция Vacansy <code>_id: ObjectId('65d3b95785efc9e3bf2a2111')</code> <code>name_vac : "Разработчик SQL БД"</code>	Коллекция Vacansy Все остальные Водитель <code>65d3b95785efc9e3bf2a2117</code> Преподаватель SQL <code>65d3b95785efc9e3bf2a211e</code> Уборщик <code>65d3b95785efc9e3bf2a2126'</code> <code>)</code> Волонтер гуманитарной помощи <code>65d3b95785efc9e3bf2a212c</code> Волонтер на хоккейную арену <code>65d3b95785efc9e3bf2a2131</code>
б. работодатели в Санкт-Петербурге, выставившие несколько вакансий	Коллекция Employer <code>_id: ObjectId('65d3b95785efc9e3bf2a210d')</code> <code>name_emp : "Stole"</code>	Коллекция Employer Все остальные Omega <code>65d3b95785efc9e3bf2a210e</code> Reset <code>65d3b95785efc9e3bf2a210f</code> Spb here

		65d3b95785efc9e3bf2a2110
в. Вакансия с наибольшей зарплатой	Коллекция Vacancy _id: ObjectId('65d3b95785efc9e3bf2a2111') name_vac : "Разработчик SQL БД"	Коллекция Vacancy Все остальные Водитель 65d3b95785efc9e3bf2a2117 Преподаватель SQL 65d3b95785efc9e3bf2a211e Уборщик 65d3b95785efc9e3bf2a2126 Волонтер гуманитарной помощи 65d3b95785efc9e3bf2a212c Волонтер на хоккейную арену 65d3b95785efc9e3bf2a2131
г. волонтерские позиции с максимальным количеством требований	Коллекция Vacancy _id: ObjectId('65d3b95785efc9e3bf2a2131') name_vac : "Волонтер на хоккейную арену"	Коллекция Vacancy Все остальные Разработчик SQL БД 65d3b95785efc9e3bf2a2111 Водитель 65d3b95785efc9e3bf2a2117 Преподаватель SQL 65d3b95785efc9e3bf2a211e Уборщик 65d3b95785efc9e3bf2a2126 Волонтер гуманитарной помощи 65d3b95785efc9e3bf2a212c
д. Вакансии, в которых нет требования к опыту работы	Коллекция Vacancy _id: ObjectId('65d3b95785efc9e3bf2a2111') name_vac : "Разработчик SQL БД" _id: ObjectId('65d3b95785efc9e3bf2a2117') name_vac : "Водитель" _id: ObjectId('65d3b95785efc9e3bf2a2126') name_vac : "Уборщик" _id: ObjectId('65d3b95785efc9e3bf2a212c') name_vac : "Волонтер гуманитарной помощи"	Коллекция Vacancy Преподаватель SQL 65d3b95785efc9e3bf2a211e

	<code>_id: ObjectId('65d3b95785efc9e3bf2a2131')</code> <code>name_vac : "Волонтер на хоккейную арену"</code>	
--	---	--

5. Вывод

В ходе работы я спроектировал структуру json файла, соответствующую предметной области по варианту задания и создал базу данных mongoDB с тестовыми данными.