|  |
| --- |
| **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.

1. 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

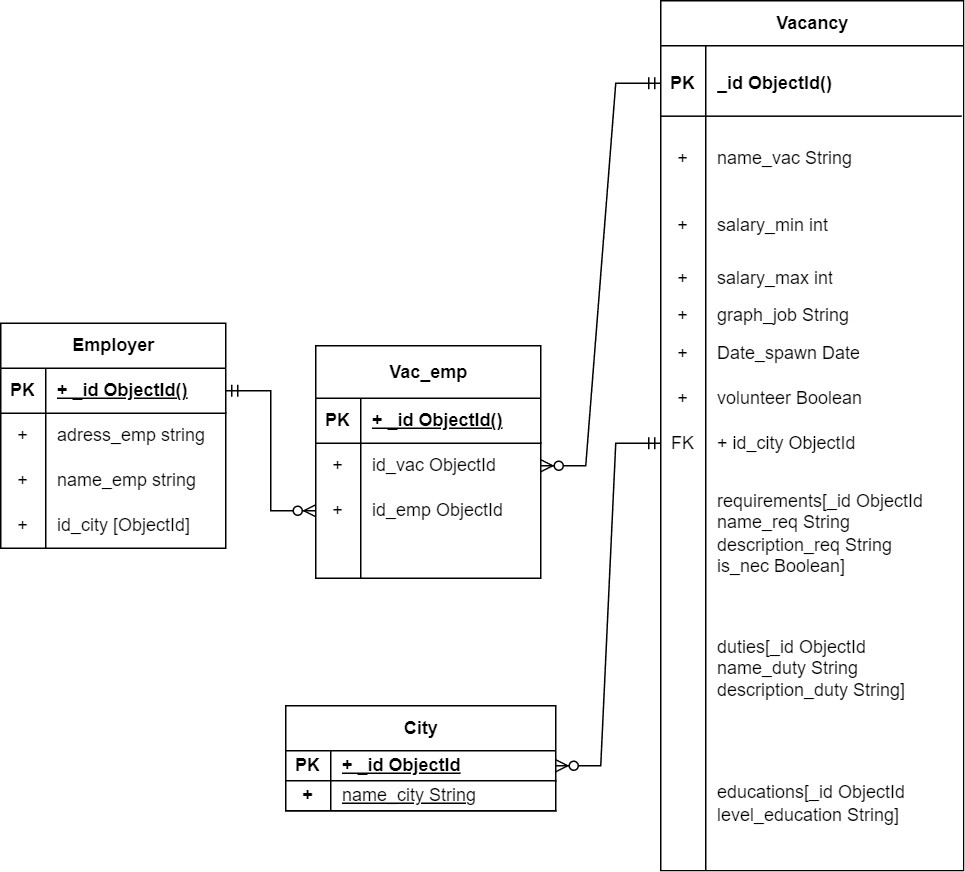
б. работодатели в Санкт-Петербурге, выставившие несколько вакансий

в. вакансия с наибольшей зарплатой

г. волонтерские позиции с максимальным количеством требований

д. вакансии, в которых нет требования к опыту работы

1. **Физическая модель:**



1. **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

}

}

1. **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 B", "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": "Преподователь 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})

1. **Набор тестовых данных**

|  |  |  |
| --- | --- | --- |
| Текст запроса | Данные удовлетворяющие условиям запросов | Данные не удовлетворяющие условиям запросов |
| 1. вакансии, имеющие в названии SQL, но не заканчивающиеся на него | Коллекция Vacancy | Коллекция Vacancy  Все остальные  Водитель  65d3b95785efc9e3bf2a2117  Преподаватель SQL  65d3b95785efc9e3bf2a211e  Уборщик 65d3b95785efc9e3bf2a2126')  Волонтер гуманитарной помощи  65d3b95785efc9e3bf2a212c  Волонтер на хоккейную арену  65d3b95785efc9e3bf2a2131 |
| 1. работодатели в Санкт-Петербурге, выставившие несколько вакансий | Коллекция Employer | Коллекция Employer  Все остальные  Omega  65d3b95785efc9e3bf2a210e  Reset  65d3b95785efc9e3bf2a210f  Spb here  65d3b95785efc9e3bf2a2110 |
| 1. Вакансия с наибольшей зарплатой | Коллекция Vacancy | Коллекция Vacancy  Все остальные  Водитель  65d3b95785efc9e3bf2a2117  Преподаватель SQL  65d3b95785efc9e3bf2a211e  Уборщик  65d3b95785efc9e3bf2a2126  Волонтер гуманитарной помощи  65d3b95785efc9e3bf2a212c  Волонтер на хоккейную арену  65d3b95785efc9e3bf2a2131 |
| 1. волонтерские позиции с максимальным количеством требований | Коллекция Vacancy | Коллекция Vacancy  Все остальные  Разработчик SQL БД  65d3b95785efc9e3bf2a2111  Водитель  65d3b95785efc9e3bf2a2117  Преподователь SQL  65d3b95785efc9e3bf2a211e  Уборщик  65d3b95785efc9e3bf2a2126  Волонтер гуманитарной помощи  65d3b95785efc9e3bf2a212c |
| 1. Вакансии, в которых нет требования к опыту работы | Коллекция Vacancy | Коллекция Vacancy  Преподователь SQL  65d3b95785efc9e3bf2a211e |

1. **Вывод**

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