

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

Старший преподаватель  
\_\_\_\_\_  
должность, уч. степень,  
звание

\_\_\_\_\_  
подпись, дата

Н.В Путилова  
\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

Манипулирование данными в документной базе данных.

**по дисциплине: Проектирование баз данных**

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

4134к  
\_\_\_\_\_

\_\_\_\_\_  
подпись, дата

Столяров Н.С.  
\_\_\_\_\_  
инициалы, фамилия

## Описание задания

1)Привести пример обновления и удаления данных из документной базы данных, разработанной в работе 11. (по 2 примера: один из вложенного документа, другой из документа коллекции верхнего уровня)

2)Выполнить запросы на выборку из базы данных в соответствии с вариантом задания .

(Запрос на поиск подстроки.

Запрос на использование одного документа несколько раз.

Запрос с условием на агрегатную функцию.

Запрос получением агрегатной функции от агрегатной функции.

Запрос на разность 2 запросов)

19 вариант

Туристический путеводитель с учетом не только «адресных» достопримечательностей, но и

природных: местность, город, достопримечательность, адрес, тип достопримечательности

(памятник, архитектурный комплекс, природный комплекс), дата создания и выставок

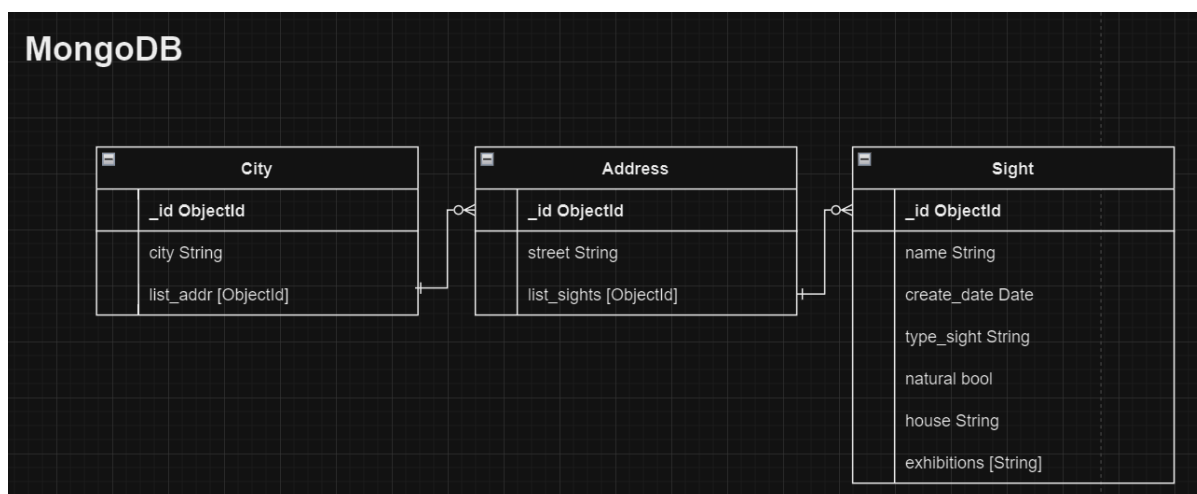
а. достопримечательности, в которых есть слово «мать», но с него название не начинается

б. улица, на которой есть 2 разных вида достопримечательности

в. город, в котором не проводится выставок

г. местность, где больше всего природных достопримечательностей

д. города с самыми старыми достопримечательностями



Текст запроса	Данные, удовлетворяющие условиям запросов	Данные не удовлетворяющие условиям запросов
а. достопримечательности, в которых есть слово «мать», но с него название не начинается	Родина мать	Кремль Большой театр Эрмитаж Петропавловская крепость Казанский Кремль Мечеть Кул-Шариф Горький Парк Таврический сад Центральный парк культуры и отдыха
б. улица, на которой есть 2 разных вида достопримечательности	Красная площадь Невский проспект Баумана	Арбат Дворцовая площадь Кремлёвская
в. город, в котором не проводится выставок	Москва	
г. местность, где больше всего природных достопримечательностей	Москва Санкт-Петербург	Санкт-Петербург Казань
д. города с самыми старыми достопримечательностями	Москва	Санкт-Петербург Казань

### Листинг запросов

// а. достопримечательности, в которых есть слово «мать», но с него название не начинается

```
db.Sight.find({
  name: {
    $regex: "мать",
    $options: "i",
    $not: /^мать/i // Не начинается с "мать"
  }
});
```

```
< {
  _id: ObjectId("6671647aabc36cc982a11840"),
  name: 'Родина мать',
  create_date: 1931-06-01T00:00:00.000Z,
  type_sight: 'парк',
  natural: true,
  house: '0',
  exhibitions: []
}
```

// 6. улица, на которой есть 2 разных вида достопримечательности

```
db.Address.aggregate([
  {
    $lookup: {
      from: "Sight",
      localField: "list_sights",
      foreignField: "_id",
      as: "sights"
    }
  },
  {
    $unwind: "$sights"
  },
  {
    $group: {
      _id: "$_id",
      street: { $first: "$street" },
      sight_types: { $addToSet: "$sights.type_sight" }
    }
  },
  {
    $match: {
```

```

    $expr: { $gte: [{ $size: "$sight_types" }, 2] }
  }
},
{
  $project: {
    street: 1,
    sight_types: 1
  }
}
]);

```

```

< {
  _id: ObjectId("6671647aabc36cc982a11834"),
  street: 'Баумана',
  sight_types: [
    'музей',
    'религиозное здание',
    'парк'
  ]
}
{
  _id: ObjectId("6671647aabc36cc982a1182f"),
  street: 'Красная площадь',
  sight_types: [
    'исторический памятник',
    'парк'
  ]
}
{
  _id: ObjectId("6671647aabc36cc982a11831"),
  street: 'Невский проспект',
  sight_types: [
    'музей',
    'парк'
  ]
}

```

```

// в. город, в котором не проводится выставок
// Запрос с условием на агрегатную функцию
db.City.aggregate([
  {
    $lookup: {

```

```
    from: "Address",
    localField: "list_addr",
    foreignField: "_id",
    as: "addresses"
  }
},
{
  $unwind: "$addresses"
},
{
  $lookup: {
    from: "Sight",
    localField: "addresses.list_sights",
    foreignField: "_id",
    as: "sights"
  }
},
{
  $unwind: {
    path: "$sights",
    preserveNullAndEmptyArrays: true
  }
},
{
  $group: {
    _id: "$_id",
    city: { $first: "$city" },
    hasExhibition: {
      $sum: {
```

```

    $cond: [{ $gt: [{ $size: { $ifNull: ["$sights.exhibitions", []] } }, 0] }, 1, 0]
  }
}
},
{
  $match: {
    hasExhibition: 0
  }
}
]);

```

```

< {
  _id: ObjectId("66716479abc36cc982a1182c"),
  city: 'Москва',
  hasExhibition: 0
}

```

```

// д. города с самыми старыми достопримечательностями
// города с самыми старыми достопримечательностями
db.Sight.aggregate([
  {
    $group: {
      _id: null,
      minDate: { $min: "$create_date" }
    }
  },
  {
    $project: {
      _id: 0,

```

```
      minDate: 1
    }
  },
  {
    $lookup: {
      from: "Sight",
      localField: "minDate",
      foreignField: "create_date",
      as: "oldest_sights"
    }
  },
  {
    $unwind: "$oldest_sights"
  },
  {
    $project: {
      minDate: 1,
      sight_id: "$oldest_sights._id"
    }
  },
  {
    $lookup: {
      from: "Address",
      localField: "sight_id",
      foreignField: "list_sights",
      as: "addresses"
    }
  },
  {
```



```
$unwind: "$addresses"
},
{
  $project: {
    minDate: 1,
    address_id: "$addresses._id"
  }
},
{
  $lookup: {
    from: "City",
    localField: "address_id",
    foreignField: "list_addr",
    as: "cities"
  }
},
{
  $unwind: "$cities"
},
{
  $project: {
    _id: 0,
    city: "$cities.city"
  }
},
{
  $group: {
    _id: "$city"
  }
}
```

```
}  
]).pretty();
```

```
]).pretty();  
< {  
  _id: 'Москва'  
}
```

// г. местность, где больше всего природных достопримечательностей

```
db.City.aggregate([  
  {  
    $lookup: {  
      from: "Address",  
      localField: "list_addr",  
      foreignField: "_id",  
      as: "addresses"  
    }  
  },  
  {  
    $unwind: "$addresses"  
  },  
  {  
    $lookup: {  
      from: "Sight",  
      localField: "addresses.list_sights",  
      foreignField: "_id",  
      as: "sights"  
    }  
  },  
  {  
    $unwind: "$sights"
```

```
},
{
  $match: {
    "sights.natural": true
  }
},
{
  $group: {
    _id: "$city",
    count: { $sum: 1 }
  }
},
{
  $sort: { count: -1 }
},
{
  $group: {
    _id: null,
    maxCount: { $first: "$count" },
    cities: { $push: { city: "$_id", count: "$count" } }
  }
},
{
  $project: {
    _id: 0,
    cities: {
      $filter: {
        input: "$cities",
        as: "city",
```

```

        cond: { $eq: ["$$city.count", "$maxCount"] }
      }
    }
  },
  {
    $unwind: "$cities"
  },
  {
    $replaceRoot: { newRoot: "$cities" }
  }
])

```

```

< {
  city: 'Санкт-Петербург',
  count: 2
}
{
  city: 'Москва',
  count: 2
}

```

```

// Вставка города
db.City.insertMany([
  { _id: ObjectId(), city: "Москва", list_addr: [] },
  { _id: ObjectId(), city: "Санкт-Петербург", list_addr: [] },
  { _id: ObjectId(), city: "Казань", list_addr: [] }
]);

// Удаление города
db.City.deleteOne({ city: "Казань" });

```

