

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

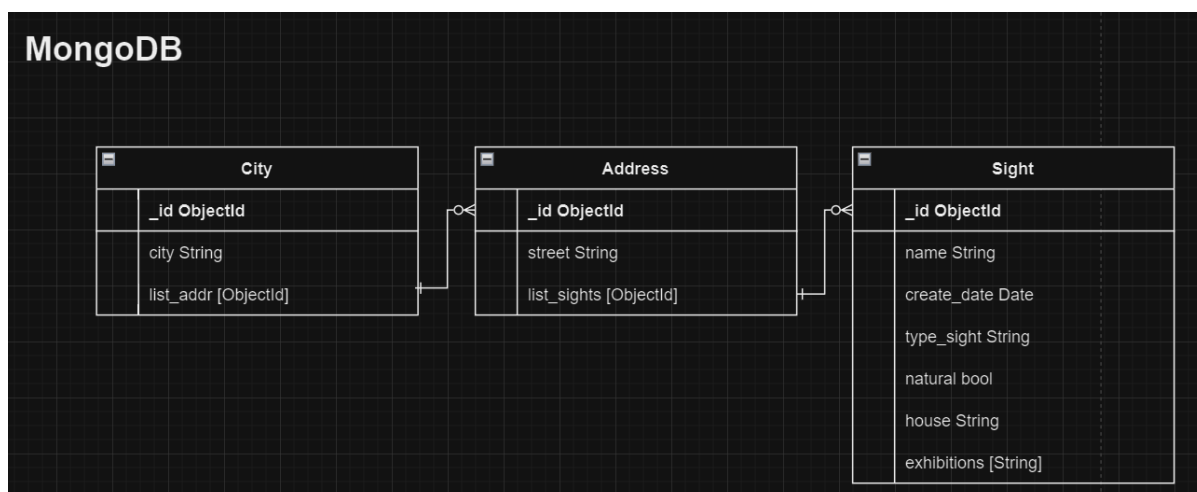
а. достопримечательности, в которых есть слово «мать», но с него название не начинается

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	Красная площадь	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

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

```
});
```

```
// б. улица, на которой есть 2 разных вида достопримечательности
```

```
// Запрос на использование одного документа несколько раз
```

```
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  
}  
}  
]);  
  
// 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"  
    }  
  },  
  {  
    $project: {  
      street: 1,  
      sight_types: 1  
    }  
  }  
]);
```

```

{
  $unwind: {
    path: "$sights",
    preserveNullAndEmptyArrays: true
  }
},
{
  $group: {
    _id: "$_id",
    city: { $first: "$city" },
    hasExhibition: {
      $sum: {
        $cond: [{ $gt: [{ $size: { $ifNull: ["$sights.exhibitions", []] } }, 0] }, 1, 0]
      }
    }
  },
  {
    $match: {
      hasExhibition: 0
    }
  }
}
]);

```

// г. местность, где больше всего природных достопримечательностей

// Запрос получением агрегатной функции от агрегатной функции

```

db.Address.aggregate([
  {
    $unwind: "$list_sights"
  }
]

```

```
},  
{  
  $lookup: {  
    from: "Sight",  
    localField: "list_sights",  
    foreignField: "_id",  
    as: "sight_info"  
  }  
},  
{  
  $unwind: "$sight_info"  
},  
{  
  $match: {  
    "sight_info.natural": true  
  }  
},  
{  
  $group: {  
    _id: "$_id",  
    street: { $first: "$street" },  
    count: { $sum: 1 }  
  }  
},  
{  
  $sort: { count: -1 }  
},  
{  
  $limit: 1  
}
```

```
}  
]);  
  
// Запрос на разность 2 запросов  
// запрос а  
var resultA = db.Sight.find({  
  name: {  
    $regex: "мать",  
    $options: "i",  
    $not: /^мать/i  
  }  
}).toArray(); // Преобразуем результат в массив  
  
// запрос б  
var resultB = 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
  }
}
]).toArray(); // Преобразуем результат в массив

// Разность результатов
var difference = resultA.filter(docA => !resultB.some(docB => docB.street ===
docA.name)); // Здесь используем имя улицы для сравнения

// д. города с самыми старыми достопримечательностями
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"
},
{
  $sort: { "sights.create_date": 1 } // Сортируем достопримечательности по
возрастанию даты создания
},
{
  $group: {
    _id: "$_id",
    city: { $first: "$city" },
    oldest_sight: { $first: "$sights" } // Берем самую старую достопримечательность
для каждого города
  }
},
{
  $project: {
    _id: 0,
```

```

city: 1,
oldest_sight: {
  name: 1,
  create_date: 1
}
},
{
  $sort: { "oldest_sight.create_date": 1 } // Сортируем города по возрастанию даты
создания достопримечательности
},
{
  $limit: 1 // Получаем только один город с самой старой достопримечательностью
}
]);

```

Результаты работы

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

```

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

```

б. улица, на которой есть 2 разных вида достопримечательности

```

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

```

в. город, в котором не проводится выставок

```

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

```

г. местность, где больше всего природных достопримечательностей

```
< {  
  _id: ObjectId("6671647aabc36cc982a1182f"),  
  street: 'Красная площадь',  
  count: 2  
}
```

д. города с самыми старыми достопримечательностями

```
< {  
  city: 'Москва',  
  oldest_sight: {  
    name: 'Кремль',  
    create_date: 1156-01-01T00:00:00.000Z  
  }  
}
```