ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Н.В Путилова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1  Разработка документной базы данных |
| **по дисциплине: Проектирование баз данных** |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Столяров Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2024

## 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.

## Задание

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

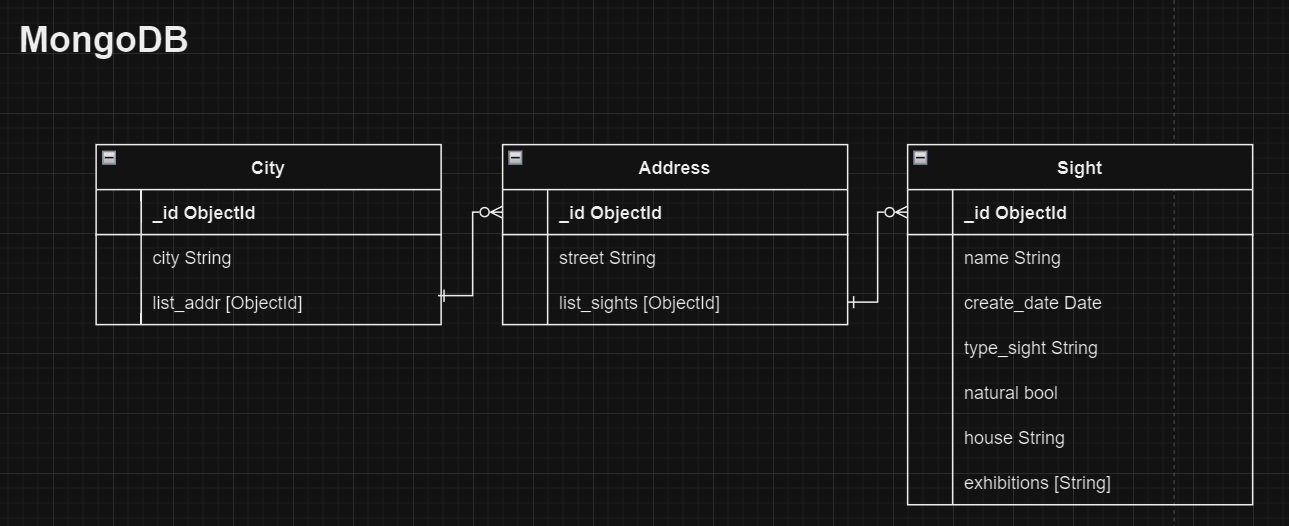
а. достопримечательности, в которых есть слово «мать», но с него название не начинается

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 неудовлетворяющие 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 |

## Ход Выполнения

|  |
| --- |
| // Переход в нужную базу данных  use sights  // Вставка коллекций  db.createCollection("City");  db.createCollection("Address");  db.createCollection("Sight");  // Вставка городов  db.City.insertMany([  {  \_id: ObjectId(),  city: "Москва",  list\_addr: []  },  {  \_id: ObjectId(),  city: "Санкт-Петербург",  list\_addr: []  },  {  \_id: ObjectId(),  city: "Казань",  list\_addr: []  }  ]);  // Вставка адресов  db.Address.insertMany([  {  \_id: ObjectId(),  street: "Красная площадь",  list\_sights: []  },  {  \_id: ObjectId(),  street: "Арбат",  list\_sights: []  },  {  \_id: ObjectId(),  street: "Невский проспект",  list\_sights: []  },  {  \_id: ObjectId(),  street: "Дворцовая площадь",  list\_sights: []  },  {  \_id: ObjectId(),  street: "Кремлёвская",  list\_sights: []  },  {  \_id: ObjectId(),  street: "Баумана",  list\_sights: []  }  ]);  // Вставка достопримечательностей  db.Sight.insertMany([  {  \_id: ObjectId(),  name: "Кремль",  create\_date: new Date("1156-01-01"),  type\_sight: "исторический памятник",  natural: false,  house: "1",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Большой театр",  create\_date: new Date("1825-01-01"),  type\_sight: "театр",  natural: false,  house: "10",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Эрмитаж",  create\_date: new Date("1764-12-07"),  type\_sight: "музей",  natural: false,  house: "30",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Русский музей",  create\_date: new Date("1895-04-13"),  type\_sight: "музей",  natural: false,  house: "4",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Музей современного искусства",  create\_date: new Date("1999-12-12"),  type\_sight: "музей",  natural: false,  house: "16",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Петропавловская крепость",  create\_date: new Date("1703-05-16"),  type\_sight: "крепость",  natural: false,  house: "2",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Казанский Кремль",  create\_date: new Date("1556-01-01"),  type\_sight: "исторический памятник",  natural: false,  house: "2",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Мечеть Кул-Шариф",  create\_date: new Date("2005-06-24"),  type\_sight: "религиозное здание",  natural: false,  house: "50",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Горький Парк",  create\_date: new Date("1928-08-12"),  type\_sight: "парк",  natural: true,  house: "9",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Таврический сад",  create\_date: new Date("1783-01-01"),  type\_sight: "парк",  natural: true,  house: "15",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Центральный парк культуры и отдыха",  create\_date: new Date("1931-06-01"),  type\_sight: "парк",  natural: true,  house: "7",  exhibitions: []  },  {  \_id: ObjectId(),  name: "Родина мать",  create\_date: new Date("1931-06-01"),  type\_sight: "парк",  natural: true,  house: "0",  exhibitions: []  }  ]);  // Получение ObjectId адресов  const address1 = db.Address.findOne({ street: "Красная площадь" }).\_id;  const address2 = db.Address.findOne({ street: "Арбат" }).\_id;  const address3 = db.Address.findOne({ street: "Невский проспект" }).\_id;  const address4 = db.Address.findOne({ street: "Дворцовая площадь" }).\_id;  const address5 = db.Address.findOne({ street: "Кремлёвская" }).\_id;  const address6 = db.Address.findOne({ street: "Баумана" }).\_id;  // Обновление городов  db.City.updateOne(  { city: "Москва" },  { $push: { list\_addr: { $each: [address1, address2] } } }  );  db.City.updateOne(  { city: "Санкт-Петербург" },  { $push: { list\_addr: { $each: [address3, address4] } } }  );  db.City.updateOne(  { city: "Казань" },  { $push: { list\_addr: { $each: [address5, address6] } } }  );  // Получение ObjectId достопримечательностей  const sight1 = db.Sight.findOne({ name: "Кремль" }).\_id;  const sight2 = db.Sight.findOne({ name: "Большой театр" }).\_id;  const sight3 = db.Sight.findOne({ name: "Эрмитаж" }).\_id;  const sight4 = db.Sight.findOne({ name: "Русский музей" }).\_id;  const sight5 = db.Sight.findOne({ name: "Музей современного искусства" }).\_id;  const sight6 = db.Sight.findOne({ name: "Петропавловская крепость" }).\_id;  const sight7 = db.Sight.findOne({ name: "Казанский Кремль" }).\_id;  const sight8 = db.Sight.findOne({ name: "Мечеть Кул-Шариф" }).\_id;  const sight9 = db.Sight.findOne({ name: "Родина мать" }).\_id;  const park1 = db.Sight.findOne({ name: "Горький Парк" }).\_id;  const park2 = db.Sight.findOne({ name: "Таврический сад" }).\_id;  const park3 = db.Sight.findOne({ name: "Центральный парк культуры и отдыха" }).\_id;  // Обновление адресов  db.Address.updateOne(  { street: "Красная площадь" },  { $push: { list\_sights: { $each: [sight1, sight9, park1] } } }  );  db.Address.updateOne(  { street: "Арбат" },  { $push: { list\_sights: sight2 } }  );  db.Address.updateOne(  { street: "Невский проспект" },  { $push: { list\_sights: { $each: [sight3, sight4, park2] } } }  );  db.Address.updateOne(  { street: "Дворцовая площадь" },  { $push: { list\_sights: sight6 } }  );  db.Address.updateOne(  { street: "Кремлёвская" },  { $push: { list\_sights: sight7 } }  );  db.Address.updateOne(  { street: "Баумана" },  { $push: { list\_sights: { $each: [sight8, sight5, park3] } } }  );  // Добавление выставок  const exhibitions1 = [  "Выставка 'Искусство и технологии'",  "Фотовыставка 'Города мира'"  ];  const exhibitions2 = [  "Выставка 'История и культура'",  "Выставка 'Современное искусство'"  ];  const exhibitions3 = [  "Фотовыставка 'Природа России'",  "Выставка 'Научные достижения'"  ];  db.Sight.updateOne(  { name: "Эрмитаж" },  { $push: { exhibitions: { $each: exhibitions1 } } }  );  db.Sight.updateOne(  { name: "Русский музей" },  { $push: { exhibitions: { $each: exhibitions2 } } }  );  db.Sight.updateOne(  { name: "Музей современного искусства" },  { $push: { exhibitions: { $each: exhibitions3 } } }  ); |