

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет инфокоммуникационных технологий

Дисциплина:

«Проектирование и реализация баз данных»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
«Работа с БД в СУБД MongoDB»

Выполнила:

Арнаутова Елизавета
группа К32422

Проверила:

Говорова Марина Михайловна

Санкт-Петербург
2023

Цель работы: овладеть практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

2 Практическое задание

Практическое задание 8.1.1:

1. Создайте базу данных learn.
2. Заполните коллекцию единорогов unicorns.
3. Используя второй способ, вставьте в коллекцию единорогов документ.
4. Проверьте содержимое коллекции с помощью метода find.

Практическое задание 8.1.2:

1. Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.
2. Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.

Практическое задание 8.1.3:

Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпочтениях и поле.

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

Практическое задание 8.1.5:

Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.

Практическое задание 8.1.6:

Вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора

Практическое задание 8.1.7:

Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

Практическое задание 8.1.9:

Вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.

Практическое задание 8.2.1:

1. Создайте коллекцию towns
2. Сформировать запрос, который возвращает список городов с независимыми мэрами (party="I"). Вывести только название города и информацию о мэре.
3. Сформировать запрос, который возвращает список беспартийных мэров (party отсутствует). Вывести только название города и информацию о мэре.

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.

2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.
3. Вывести результат, используя `forEach`.
4. Содержание коллекции единорогов `unicorns`:

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

Практическое задание 8.2.4:

Вывести список предпочтений.

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

Практическое задание 8.2.6:

1. Выполнить команду:
2. Проверить содержимое коллекции `unicorns`.

Практическое задание 8.2.7:

1. Для самки единорога Аупа внести изменения в БД: теперь ее вес 800, она убила 51 вампира.
2. Проверить содержимое коллекции `unicorns`.

Практическое задание 8.2.8:

1. Для самца единорога Raleigh внести изменения в БД: теперь он любит рэдбул.

2. Проверить содержимое коллекции unicorns.

Практическое задание 8.2.9:

1. Всем самцам единорогов увеличить количество убитых вампиров на 5.
2. Проверить содержимое коллекции unicorns.

Практическое задание 8.2.10:

1. Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.
2. Проверить содержимое коллекции towns.

Практическое задание 8.2.11:

1. Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.
2. Проверить содержимое коллекции unicorns.

Практическое задание 8.2.12:

1. Изменить информацию о самке единорога Aurora: теперь она любит еще и сахар, и лимоны.
2. Проверить содержимое коллекции unicorns.

Практическое задание 8.2.13:

1. Создайте коллекцию towns
2. Удалите документы с беспартийными мэрами.
3. Проверьте содержание коллекции.
4. Очистите коллекцию.

5. Просмотрите список доступных коллекций.

Практическое задание 8.3.1:

1. Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.
2. Включите для нескольких единорогов в документы ссылку на зону обитания, используя второй способ автоматического связывания.
3. Проверьте содержание коллекции единорогов.
4. Содержание коллекции единорогов unicorns:

Практическое задание 8.3.2:

1. Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique.
2. Содержание коллекции единорогов unicorns:

Практическое задание 8.3.3:

1. Получите информацию о всех индексах коллекции unicorns .
2. Удалите все индексы, кроме индекса для идентификатора.
3. Попытайтесь удалить индекс для идентификатора.

Практическое задание 8.3.4:

1. Создайте объемную коллекцию numbers,
2. Выберите последних четыре документа.
3. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)
4. Создайте индекс для ключа value.

5. Получите информацию о всех индексах коллекции `nombres`.
6. Выполните запрос 2.
7. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
8. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

3 Выполнение

Практическое задание 8.1.1:

1. Создайте базу данных learn.
2. Заполните коллекцию единорогов unicorns.
3. Используя второй способ, вставьте в коллекцию единорогов документ.
4. Проверьте содержимое коллекции с помощью метода find.

```
learn> use learn
already on db learn
learn> db.createCollection("unicorns")
{ ok: 1 }
learn> db.unicorns.insert({name: 'Horny', loves: ['carrot', 'papaya'], weight: 600, gender: 'm', vampires: 63});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e3b") }
}
learn> db.unicorns.insert({name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e3c") }
}
learn> db.unicorns.insert({name: 'Unicrom', loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e3d") }
}
learn> db.unicorns.insert({name: 'Rooooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e3e") }
}
learn> db.unicorns.insert({name: 'Solnara', loves: ['apple', 'carrot', 'chocolate'], weight: 550, gender: 'f', vampires: 80});
elon', weight: 650, gender: 'm', vampires: 54});
db.unicorns.insert({name: 'Nimue', loves: ['grape', 'carrot'], weight: 540, gender: 'f'});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e3f") }
}
learn> db.unicorns.insert({name: 'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e40") }
}
learn> db.unicorns.insert({name: 'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e41") }
}
learn> db.unicorns.insert({name: 'Raleigh', loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e42") }
}
learn> db.unicorns.insert({name: 'Leia', loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});
{
  acknowledged: true,
  insertedIds: { '_id': ObjectId("647237ac1a17d8fb18328e43") }
}
learn> db.unicorns.insert({name: 'Pilot', loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});
{
```

Рисунок 1 - Практическое задание 8.1.1 Пункты 1-2


```
learn> document={name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
{
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
learn> db.unicorns.insert(document)
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64723a591a17d8fb18328e46") }
}
```

Рисунок 2- Практическое задание 8.1.1 Пункты 3

```

learn> db.unicorns.find()
[
  {
    _id: ObjectId("647237ac1a17d8fb18328e3b"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e3c"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e3d"),
    name: 'Unicrom',
    loves: [ 'energon', 'redbull' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e3e"),
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e3f"),
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e40"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e41"),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]

```

Рисунок 3 - Практическое задание 8.1.1 Пункт 4 Часть 1

```

{
  _id: ObjectId("647237ac1a17d8fb18328e40"),
  name: 'Ayna',
  loves: [ 'strawberry', 'lemon' ],
  weight: 733,
  gender: 'f',
  vampires: 40
},
{
  _id: ObjectId("647237ac1a17d8fb18328e41"),
  name: 'Kenny',
  loves: [ 'grape', 'lemon' ],
  weight: 690,
  gender: 'm',
  vampires: 39
},
{
  _id: ObjectId("647237ac1a17d8fb18328e42"),
  name: 'Raleigh',
  loves: [ 'apple', 'sugar' ],
  weight: 421,
  gender: 'm',
  vampires: 2
},
{
  _id: ObjectId("647237ac1a17d8fb18328e43"),
  name: 'Leia',
  loves: [ 'apple', 'watermelon' ],
  weight: 601,
  gender: 'f',
  vampires: 33
},
{
  _id: ObjectId("647237ac1a17d8fb18328e44"),
  name: 'Pilot',
  loves: [ 'apple', 'watermelon' ],
  weight: 650,
  gender: 'm',
  vampires: 54
},
{
  _id: ObjectId("647237ac1a17d8fb18328e45"),
  name: 'Nimue',
  loves: [ 'grape', 'carrot' ],
  weight: 540,
  gender: 'f'
},
{
  _id: ObjectId("64723a591a17d8fb18328e46"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
]

```

Рисунок 4 - Практическое задание 8.1.1 Пункт 4 Часть 2

Практическое задание 8.1.2:

3. Сформируйте запросы для вывода списков самцов и самок единорогов. Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.
4. Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit.

```
[learn> db.unicorns.find({gender: "m"}).sort({name: 1}).limit(3)
[
  {
    _id: ObjectId("64723a591a17d8fb18328e46"),
    name: 'Dunx',
    loves: [ 'grape', 'watermelon' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e3b"),
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e41"),
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
[learn> db.unicorns.find({gender: "f"}).sort({name: 1}).limit(3)
[
  {
    _id: ObjectId("647237ac1a17d8fb18328e3c"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e40"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e43"),
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  }
]
```

Рисунок 5- Практическое задание 8.1.2 Пункт 1

```
[learn> db.unicorns.findOne({gender: "f", loves: "carrot"})
{
  _id: ObjectId("647237ac1a17d8fb18328e3c"),
  name: 'Aurora',
  loves: [ 'carrot', 'grape' ],
  weight: 450,
  gender: 'f',
  vampires: 43
}
[learn> db.unicorns.find({gender: "f", loves: "carrot"}).limit(1)
[
  {
    _id: ObjectId("647237ac1a17d8fb18328e3c"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
```

Рисунок 6 - Практическое задание 8.1.2 Пункт 2

Практическое задание 8.1.3:

Модифицируйте запрос для вывода списков самцов единорогов, исключив из результата информацию о предпочтениях и поле.

```

[learn> db.unicorns.find({gender: "m"}, {loves: 0, gender: 0})
[
  {
    _id: ObjectId("647237ac1a17d8fb18328e3b"),
    name: 'Horny',
    weight: 600,
    vampires: 63
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e3d"),
    name: 'Unicrom',
    weight: 984,
    vampires: 182
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e3e"),
    name: 'Roooooodles',
    weight: 575,
    vampires: 99
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e41"),
    name: 'Kenny',
    weight: 690,
    vampires: 39
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e42"),
    name: 'Raleigh',
    weight: 421,
    vampires: 2
  },
  {
    _id: ObjectId("647237ac1a17d8fb18328e44"),
    name: 'Pilot',
    weight: 650,
    vampires: 54
  },
  {
    _id: ObjectId("64723a591a17d8fb18328e46"),
    name: 'Dunx',
    weight: 704,
    vampires: 165
  }
]

```

Рисунок 7- Практическое задание 8.1.3

Практическое задание 8.1.4:

Вывести список единорогов в обратном порядке добавления.

```

[learn> db.unicorns.find({}, {name: 1}).sort({ $natural: -1 })
[
  { _id: ObjectId("64723a591a17d8fb18328e46"), name: 'Dunx' },
  { _id: ObjectId("647237ac1a17d8fb18328e45"), name: 'Nimue' },
  { _id: ObjectId("647237ac1a17d8fb18328e44"), name: 'Pilot' },
  { _id: ObjectId("647237ac1a17d8fb18328e43"), name: 'Leia' },
  { _id: ObjectId("647237ac1a17d8fb18328e42"), name: 'Raleigh' },
  { _id: ObjectId("647237ac1a17d8fb18328e41"), name: 'Kenny' },
  { _id: ObjectId("647237ac1a17d8fb18328e40"), name: 'Ayna' },
  { _id: ObjectId("647237ac1a17d8fb18328e3f"), name: 'Solnara' },
  { _id: ObjectId("647237ac1a17d8fb18328e3e"), name: 'Rooooooodles' },
  { _id: ObjectId("647237ac1a17d8fb18328e3d"), name: 'Unicrom' },
  { _id: ObjectId("647237ac1a17d8fb18328e3c"), name: 'Aurora' },
  { _id: ObjectId("647237ac1a17d8fb18328e3b"), name: 'Horny' }
]

```

Рисунок 8 - Практическое задание 8.1.4

Практическое задание 8.1.5:

Вывести список единорогов с названием первого любимого предпочтения, исключив идентификатор.

```

learn> db.unicorns.find({}, {_id: 0, loves: {$slice : 1}})
[
  {
    name: 'Horny',
    loves: [ 'carrot' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    name: 'Aurora',
    loves: [ 'carrot' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  },
  {
    name: 'Unicrom',
    loves: [ 'energon' ],
    weight: 984,
    gender: 'm',
    vampires: 182
  },
  {
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Ayna',
    loves: [ 'strawberry' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  },
  {
    name: 'Kenny',
    loves: [ 'grape' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    name: 'Raleigh',
    loves: [ 'apple' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  },
  {
    name: 'Leia',
    loves: [ 'apple' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    name: 'Pilot',
    loves: [ 'apple' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  { name: 'Nimue', loves: [ 'grape' ], weight: 540, gender: 'f' },
  {
    name: 'Dunx',
    loves: [ 'grape' ],
    weight: 704,
    gender: 'm',
    vampires: 165
  }
]

```

Рисунок 9 - Практическое задание 8.1.5

Практическое задание 8.1.6:

Вывести список самок единорогов весом от полутонны до 700 кг, исключив вывод идентификатора

```

[learn> db.unicorns.find({weight: {$gt : 500, $lt: 700}}, {_id: 0})
[
  {
    name: 'Horny',
    loves: [ 'carrot', 'papaya' ],
    weight: 600,
    gender: 'm',
    vampires: 63
  },
  {
    name: 'Rooooooodles',
    loves: [ 'apple' ],
    weight: 575,
    gender: 'm',
    vampires: 99
  },
  {
    name: 'Solnara',
    loves: [ 'apple', 'carrot', 'chocolate' ],
    weight: 550,
    gender: 'f',
    vampires: 80
  },
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  },
  {
    name: 'Leia',
    loves: [ 'apple', 'watermelon' ],
    weight: 601,
    gender: 'f',
    vampires: 33
  },
  {
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 54
  },
  {
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]

```

Рисунок 10 - Практическое задание 8.1.6

Практическое задание 8.1.7:

Вывести список самцов единорогов весом от полутонны и предпочитающих grape и lemon, исключив вывод идентификатора.

```
[learn> db.unicorns.find({weight: {$gt : 500}, loves: ["grape", "lemon"], gender: "m"}, {_id: 0})
[
  {
    name: 'Kenny',
    loves: [ 'grape', 'lemon' ],
    weight: 690,
    gender: 'm',
    vampires: 39
  }
]
```

Рисунок 11 - Практическое задание 8.1.7

Практическое задание 8.1.8:

Найти всех единорогов, не имеющих ключ vampires.

```
[learn> db.unicorns.find({vampires: {$exists:false}})
[
  {
    _id: ObjectId("647237ac1a17d8fb18328e45"),
    name: 'Nimue',
    loves: [ 'grape', 'carrot' ],
    weight: 540,
    gender: 'f'
  }
]
```

Рисунок 12 - Практическое задание 8.1.8

Практическое задание 8.1.9:

Вывести список упорядоченный список имен самцов единорогов с информацией об их первом предпочтении.

```
[learn> db.unicorns.find({gender: "m"}, {name: 1, _id: 0, loves: {$slice : 1}}).sort({name: 1})
[
  { name: 'Dunx', loves: [ 'grape' ] },
  { name: 'Horny', loves: [ 'carrot' ] },
  { name: 'Kenny', loves: [ 'grape' ] },
  { name: 'Pilot', loves: [ 'apple' ] },
  { name: 'Raleigh', loves: [ 'apple' ] },
  { name: 'Rooooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
```

Рисунок 13 - Практическое задание 8.1.9

Практическое задание 8.2.1:

1. Создайте коллекцию towns
2. Сформировать запрос, который возвращает список городов с независимыми мэрами (party="I"). Вывести только название города и информацию о мэре.
3. Сформировать запрос, который возвращает список беспартийных мэров (party отсутствует). Вывести только название города и информацию о мэре.

```
[learn> db.towns.find({"mayor.party": "I"}, {name: 1, mayor: 1, _id: 0})  
[  
  {  
    name: 'New York',  
    mayor: { name: 'Michael Bloomberg', party: 'I' }  
  }  
]
```

Рисунок 14 - Практическое задание 8.2.1 Пункт 2

```
[learn> db.towns.find({"mayor.party":{"exists":false}}, {name: 1, mayor: 1, _id: 0})  
[ { name: 'Punxsutawney ', mayor: { name: 'Jim Wehrle' } } ]
```

Рисунок 15 - Практическое задание 8.2.1 Пункт 3

Практическое задание 8.2.2:

1. Сформировать функцию для вывода списка самцов единорогов.
2. Создать курсор для этого списка из первых двух особей с сортировкой в лексикографическом порядке.

3. Вывести результат, используя `forEach`.

```
[learn> var cursor = db.unicorns.find({"$where": fn});null;
null
[learn> cursor.sort({name:1}).limit(2);null
null
[learn> cursor.forEach(function(obj){print(obj)})
{
  _id: ObjectId("64723a591a17d8fb18328e46"),
  name: 'Dunx',
  loves: [ 'grape', 'watermelon' ],
  weight: 704,
  gender: 'm',
  vampires: 165
}
{
  _id: ObjectId("647237ac1a17d8fb18328e3b"),
  name: 'Horny',
  loves: [ 'carrot', 'papaya' ],
  weight: 600,
  gender: 'm',
  vampires: 63
}
```

Рисунок 16 - Практическое задание 8.2.2

Практическое задание 8.2.3:

Вывести количество самок единорогов весом от полутонны до 600 кг.

```
[learn> db.unicorns.find({weight: {$gt : 500, $lt: 600}, gender: "f"}).count()
2
```

Рисунок 17 - Практическое задание 8.2.3

Практическое задание 8.2.4:

Вывести список предпочтений.

```
[learn> db.unicorns.distinct("loves")
[
  'apple',      'carrot',
  'chocolate', 'energon',
  'grape',      'lemon',
  'papaya',     'redbull',
  'strawberry', 'sugar',
  'watermelon'
]
```

Рисунок 18 - Практическое задание 8.2.4

Практическое задание 8.2.5:

Посчитать количество особей единорогов обоих полов.

```
[learn> db.users.aggregate({"$group":{_id:"$name",count:{$sum:1}}})
[learn> db.unicorns.aggregate({"$group":{_id:"$gender",count:{$sum:1}}})
[ { _id: 'm', count: 7 }, { _id: 'f', count: 5 } ]
```

Рисунок 19 - Практическое задание 8.2.5

Практическое задание 8.2.6:

3. Выполнить команду:
4. Проверить содержимое коллекции unicorns.

```
[learn> db.unicorns.find({}, {name: 1})
[
  { _id: ObjectId("647249181a17d8fb18328e55"), name: 'Horny' },
  { _id: ObjectId("647249181a17d8fb18328e56"), name: 'Aurora' },
  { _id: ObjectId("647249181a17d8fb18328e57"), name: 'Unicrom' },
  { _id: ObjectId("647249181a17d8fb18328e58"), name: 'Solnara' },
  { _id: ObjectId("647249181a17d8fb18328e59"), name: 'Ayna' },
  { _id: ObjectId("647249181a17d8fb18328e5a"), name: 'Kenny' },
  { _id: ObjectId("647249181a17d8fb18328e5b"), name: 'Raleigh' },
  { _id: ObjectId("647249181a17d8fb18328e5c"), name: 'Leia' },
  { _id: ObjectId("6472491a1a17d8fb18328e5d"), name: 'Pilot' },
  { _id: ObjectId("6472491a1a17d8fb18328e5e"), name: 'Nimue' },
  { _id: ObjectId("6472491a1a17d8fb18328e5f"), name: 'Dunx' },
  { _id: ObjectId("64724a901a17d8fb18328e60"), name: 'Rooooooodles' }
]
```

Рисунок 20 – коллекция до выполнения команды

```
learn> db.unicorns.save({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
TypeError: db.unicorns.save is not a function
learn> db.unicorns.find({}, {name: 1})
```

Рисунок 21 – после применения save

Поскольку функция save не работала не работала, как ожидалось — не добавила в коллекцию необходимый документ — добавим вручную.

```
learn> db.unicorns.insert({name: 'Barney', loves: ['grape'], weight: 340, gender: 'm'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("64724bc01a17d8fb18328e61") }
}
learn> db.unicorns.find({}, {name: 1})
[
  { _id: ObjectId("647249181a17d8fb18328e55"), name: 'Horny' },
  { _id: ObjectId("647249181a17d8fb18328e56"), name: 'Aurora' },
  { _id: ObjectId("647249181a17d8fb18328e57"), name: 'Unicrom' },
  { _id: ObjectId("647249181a17d8fb18328e58"), name: 'Solnara' },
  { _id: ObjectId("647249181a17d8fb18328e59"), name: 'Ayna' },
  { _id: ObjectId("647249181a17d8fb18328e5a"), name: 'Kenny' },
  { _id: ObjectId("647249181a17d8fb18328e5b"), name: 'Raleigh' },
  { _id: ObjectId("647249181a17d8fb18328e5c"), name: 'Leia' },
  { _id: ObjectId("6472491a1a17d8fb18328e5d"), name: 'Pilot' },
  { _id: ObjectId("6472491a1a17d8fb18328e5e"), name: 'Nimue' },
  { _id: ObjectId("6472491a1a17d8fb18328e5f"), name: 'Dunx' },
  { _id: ObjectId("64724a901a17d8fb18328e60"), name: 'Rooooooodles' },
  { _id: ObjectId("64724bc01a17d8fb18328e61"), name: 'Barney' }
]
```

Рисунок 22 – коллекция после

Практическое задание 8.2.7:

- Для самки единорога Ауна внести изменения в БД: теперь ее вес 800, она убила 51 вампира.
- Проверить содержимое коллекции unicorns.

```
learn> db.unicorns.find({name: 'Ayna'})
[
  {
    _id: ObjectId("647249181a17d8fb18328e59"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 733,
    gender: 'f',
    vampires: 40
  }
]
```

Рисунок 23 – значение до выполнения команды


```

[learn> db.unicorns.update({name: 'Ayna'}, {weight: 800, vampires: 51})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
MongoInvalidArgumentError: Update document requires atomic operators
[learn> db.unicorns.updateOne({name: 'Ayna'}, {weight: 800, vampires: 51})
MongoInvalidArgumentError: Update document requires atomic operators
[learn> db.unicorns.updateOne({name: 'Ayna'}, {$set: {weight: 800, vampires: 51}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[learn> db.unicorns.find({name: 'Ayna'})
[
  {
    _id: ObjectId("647249181a17d8fb18328e59"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
    gender: 'f',
    vampires: 51
  }
]

```

Рисунок 24 - Практическое задание 8.2.7

Практическое задание 8.2.8:

3. Для самца единорога Raleigh внести изменения в БД: теперь он любит редбул.
4. Проверить содержимое коллекции unicorns.

```

[learn> db.unicorns.find({name: 'Raleigh'})
[
  {
    _id: ObjectId("647249181a17d8fb18328e5b"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
[learn> db.unicorns.updateOne({name: 'Raleigh'}, {$set: {loves: ['redbull']}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[learn> db.unicorns.find({name: 'Raleigh'})
[
  {
    _id: ObjectId("647249181a17d8fb18328e5b"),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]

```

Рисунок 25 - Практическое задание 8.2.8

Практическое задание 8.2.9:

3. Всем самцам единорогов увеличить количество убитых вампиров на 5.
4. Проверить содержимое коллекции unicorns.

```
[learn> db.unicorns.find({gender: 'm'}, {name: 1, gender: 1, vampires: 1, _id: 0})
[
  { name: 'Horny', gender: 'm', vampires: 63 },
  { name: 'Unicrom', gender: 'm', vampires: 182 },
  { name: 'Kenny', gender: 'm', vampires: 39 },
  { name: 'Raleigh', gender: 'm', vampires: 2 },
  { name: 'Pilot', gender: 'm', vampires: 54 },
  { name: 'Dunx', gender: 'm', vampires: 165 },
  { name: 'Rooooooodles', gender: 'm', vampires: 99 },
  { name: 'Barney', gender: 'm' }
]
[learn> db.unicorns.updateMany({gender: "m"}, {$inc: {vampires: 5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0
}
[learn> db.unicorns.find({gender: 'm'}, {name: 1, gender: 1, vampires: 1, _id: 0})
[
  { name: 'Horny', gender: 'm', vampires: 68 },
  { name: 'Unicrom', gender: 'm', vampires: 187 },
  { name: 'Kenny', gender: 'm', vampires: 44 },
  { name: 'Raleigh', gender: 'm', vampires: 7 },
  { name: 'Pilot', gender: 'm', vampires: 59 },
  { name: 'Dunx', gender: 'm', vampires: 170 },
  { name: 'Rooooooodles', gender: 'm', vampires: 104 },
  { name: 'Barney', gender: 'm', vampires: 5 }
]
]
```

Рисунок 26 - Практическое задание 8.2.9

Практическое задание 8.2.10:

3. Изменить информацию о городе Портланд: мэр этого города теперь беспартийный.
4. Проверить содержимое коллекции towns.

```

[learn> db.towns.find({name: 'Portland'})
[
  {
    _id: ObjectId("647243b61a17d8fb18328e49"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
[learn> db.towns.updateOne({name: 'Portland'}, {$unset: {"mayor.party": 1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[learn> db.towns.find({name: 'Portland'})
[
  {
    _id: ObjectId("647243b61a17d8fb18328e49"),
    name: 'Portland',
    populatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  }
]

```

Рисунок 27 - Практическое задание 8.2.10

Практическое задание 8.2.11:

3. Изменить информацию о самце единорога Pilot: теперь он любит и шоколад.
4. Проверить содержимое коллекции unicorns.

```

[learn> db.unicorns.find({name: 'Pilot'})
[
  {
    _id: ObjectId("6472491a1a17d8fb18328e5d"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
[learn> db.unicorns.updateOne({name: 'Pilot'}, {$push: {loves: "chocolate"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[learn> db.unicorns.find({name: 'Pilot'})
[
  {
    _id: ObjectId("6472491a1a17d8fb18328e5d"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]

```

Рисунок 28 - Практическое задание 8.2.11

Практическое задание 8.2.12:

3. Изменить информацию о самке единорога Аурога: теперь она любит еще и сахар, и лимоны.
4. Проверить содержимое коллекции unicorns.

```

[learn> db.unicorns.find({name: 'Aurora'})
[
  {
    _id: ObjectId("647249181a17d8fb18328e56"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
learn> db.unicorns.updateOne({name: 'Aurora'}, {$addToSet: {loves: {$each: ["surag", "lemons"]}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[learn> db.unicorns.find({name: 'Aurora'})
[
  {
    _id: ObjectId("647249181a17d8fb18328e56"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'surag', 'lemons' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]

```

Рисунок 29 - Практическое задание 8.2.12

Практическое задание 8.2.13:

1. Создайте коллекцию towns
2. Удалите документы с беспартийными мэрами.
3. Проверьте содержание коллекции.
4. Очистите коллекцию.
5. Просмотрите список доступных коллекций.

```

[learn> db.towns.find()
[
  {
    _id: ObjectId("64725a301a17d8fb18328e65"),
    name: 'Punxsutawney ',
    popujatiuon: 6200,
    last_sensus: ISODate("2008-01-31T00:00:00.000Z"),
    famous_for: [ 'phil the groundhog' ],
    mayor: { name: 'Jim Wehrle' }
  },
  {
    _id: ObjectId("64725a301a17d8fb18328e66"),
    name: 'New York',
    popujatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("64725a301a17d8fb18328e67"),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
[learn> db.towns.remove({"mayor.party": {$exists: false}})
{ acknowledged: true, deletedCount: 1 }
[learn> db.towns.find()
[
  {
    _id: ObjectId("64725a301a17d8fb18328e66"),
    name: 'New York',
    popujatiuon: 22200000,
    last_sensus: ISODate("2009-07-31T00:00:00.000Z"),
    famous_for: [ 'status of liberty', 'food' ],
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  },
  {
    _id: ObjectId("64725a301a17d8fb18328e67"),
    name: 'Portland',
    popujatiuon: 528000,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]

```

Рисунок 30 - Практическое задание 8.2.12, Пункты 2-3

```

[learn> db.towns.remove({})
{ acknowledged: true, deletedCount: 2 }
[learn> show collections
towns
unicorns
users

```

Рисунок 31 - - Практическое задание 8.2.12, Пункты 4-5

Поскольку мы очистили коллекцию, но не удалили ее саму, она продолжает отображаться в списке.

Практическое задание 8.3.1:

1. Создайте коллекцию зон обитания единорогов, указав в качестве идентификатора кратко название зоны, далее включив полное название и описание.
2. Включите для нескольких единорогов в документы ссылку на зону обитания, используя второй способ автоматического связывания.
3. Проверьте содержание коллекции единорогов.

```

[learn> db.unicorn_habitat.find()
[
  { _id: 'f', name: 'forest', description: 'mysterious forest' },
  { _id: 'l', name: 'lake', description: 'lake shore' },
  { _id: 'm', name: 'mountains', description: 'high mountains' },
  { _id: 'fi', name: 'fields', description: 'flower fields' }
]

```

Рисунок 32 - Практическое задание 8.3.1, Пункт 1

```

[learn> db.unicorns.updateMany({name: 'Rooooooodles'}, {$set: {habitat: {$ref: "unicorn_habitat", $id: "l"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
[learn> db.unicorns.find({}, {name: 1, habitat: 1, _id: 0})
[
  { name: 'Horny', habitat: DBRef("unicorn_habitat", 'f') },
  { name: 'Aurora' },
  { name: 'Unicrom' },
  { name: 'Solnara', habitat: DBRef("unicorn_habitat", 'fi') },
  { name: 'Ayna' },
  { name: 'Kenny' },
  { name: 'Raleigh' },
  { name: 'Leia' },
  { name: 'Pilot', habitat: DBRef("unicorn_habitat", 'm') },
  { name: 'Nimue' },
  { name: 'Rooooooodles', habitat: DBRef("unicorn_habitat", 'l') },
  { name: 'Dunx' }
]

```

Рисунок 33 - Практическое задание 8.3.1, Пункт 2-3

Практическое задание 8.3.2:

Проверьте, можно ли задать для коллекции unicorns индекс для ключа name с флагом unique.

```
[learn> db.unicorns.ensureIndex({"name" : 1}, {"unique" : true})
[ 'name_1' ]
```

Рисунок 34 - Практическое задание 8.3.2

Имена в датасете не повторяются — да, можно создать.

Практическое задание 8.3.3:

1. Получите информацию о всех индексах коллекции unicorns .
2. Удалите все индексы, кроме индекса для идентификатора.
3. Попробуйте удалить индекс для идентификатора.

```
[learn> db.unicorns.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1', unique: true }
]
[learn> db.unicorns.dropIndex("name_1")
{ nIndexWas: 2, ok: 1 }
[learn> db.unicorns.dropIndex("_id")
MongoShellInternalError: index not found with name [_id]
[learn> db.unicorns.dropIndex("_id_")
MongoServerError: cannot drop _id index
```

Рисунок 35 - Практическое задание 8.3.3

Практическое задание 8.3.4:

1. Создайте объемную коллекцию numbers,
2. Выберите последних четыре документа.
3. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)
4. Создайте индекс для ключа value.

5. Получите информацию о всех индексах коллекции `numbers`.
6. Выполните запрос 2.
7. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
8. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

```
[learn> db.numbers.find().sort({_id: -1}).limit(4)
[
  { _id: ObjectId("647266341a17d8fb1834151f"), value: 99999 },
  { _id: ObjectId("647266341a17d8fb1834151e"), value: 99998 },
  { _id: ObjectId("647266341a17d8fb1834151d"), value: 99997 },
  { _id: ObjectId("647266341a17d8fb1834151c"), value: 99996 }
]
```

Рисунок 36 - Практическое задание 8.3.3, Пункт 2


```

[learn> db.numbersss.find().sort({_id: -1}).limit(4).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'learn.numbersss',
    indexFilterSet: false,
    parsedQuery: {},
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: { stage: 'EOF' },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 0,
    executionStages: {
      stage: 'EOF',
      nReturned: 0,
      executionTimeMillisEstimate: 0,
      works: 1,
      advanced: 0,
      needTime: 0,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1
    }
  },
  command: {
    find: 'numbersss',
    filter: {},
    sort: { _id: -1 },
    limit: 4,
    '$db': 'learn'
  },
  serverInfo: {
    host: 'MacBook-Air-Liza.local',
    port: 27017,
    version: '6.0.6',
    gitVersion: '26b4851a412cc8b9b4a18cdb6cd0f9f642e06aa7'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,

```

Рисунок 37 - Практическое задание 8.3.3, Пункт 3

```

[learn> db.numbers.ensureIndex({"value" : 1})
[ 'value_1' ]
[learn> db.numbers.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { value: 1 }, name: 'value_1' }
]

```

Рисунок 38 - Практическое задание 8.3.3, Пункт 4-5

```

learn> db.numberss.find().sort({_id: -1}).limit(4).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'learn.numberss',
    indexFilterSet: false,
    parsedQuery: {},
    queryHash: '51B6F510',
    planCacheKey: '51B6F510',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'LIMIT',
      limitAmount: 4,
      inputStage: {
        stage: 'FETCH',
        inputStage: {
          stage: 'IXSCAN',
          keyPattern: { _id: 1 },
          indexName: '_id_',
          isMultiKey: false,
          multiKeyPaths: { _id: [] },
          isUnique: true,
          isSparse: false,
          isPartial: false,
          indexVersion: 2,
          direction: 'backward',
          indexBounds: { _id: [ '[MaxKey, MinKey]' ] }
        }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 1,
    totalKeysExamined: 0,
    totalDocsExamined: 0,
    executionStages: {
      stage: 'LIMIT',
      nReturned: 0,
      executionTimeMillisEstimate: 0,
      works: 1,
      advanced: 0,
      needTime: 0,
      needYield: 0,

```

Рисунок 39 - Практическое задание 8.3.3, Пункт 6-7

Похоже, что значение миллисекунд округляется. Потому сделать вывод о том, какой из вариантов продуктивнее, не представляется возможным.

Выводы:

В результате работы овладели практическими навыками работы с CRUD-операциями, с вложенными объектами в коллекции базы данных MongoDB, агрегации и изменения данных, со ссылками и индексами в базе данных MongoDB.

Обнаружили ряд обновленных команд, расширяющих функционал по сравнению с тем, что представлено в пособии.