Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет: **Инфокоммуникационных технологий**
Образовательная программа: **Интеллектуальные системы в гуманитарной сфере**
Направление подготовки: **45.03.04 Интеллектуальные системы в гуманитарной сфере**

Лабораторная работа №5
«Работа с БД в СУБД MongoDB»

по дисциплине:
«Базы данных»

Выполнила: Плеханова Дарья
Группа: **К324212**
Преподаватель
Говорова М. М.

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. Программное обеспечение: Miro, Draw.io.

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. {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
5. 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, включающую следующие документы:
   {name: "Punxsutawney ",
   populatiuon: 6200,
   last_sensus: ISODate("2008-01-31"),
   famous_for: [""],
   mayor: {
   name: "Jim Wehrle"
   }}

   {name: "New York",
   populatiuon: 22200000,
   last_sensus: ISODate("2009-07-31"),
   famous_for: ["status of liberty", "food"],
   mayor: {
   name: "Michael Bloomberg",
   party: "I"}}

   {name: "Portland",
   populatiuon: 528000,
   last_sensus: ISODate("2009-07-20"),
   famous_for: ["beer", "food"],
   mayor: {
   name: "Sam Adams",
   party: "D"}}
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.

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:
db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
2. Проверить содержимое коллекции unicorns.

Практическое задание 8.2.7:
1. Для самки единорога Ayna внести изменения в БД: теперь ее вес 800, она убила 51 вапмира.
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 вапмиров 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, включающую следующие документы:
```
{name: "Punxsutawney ",
popujatiuon: 6200,
last_sensus: ISODate("2008-01-31"),
famous_for: ["phil the groundhog"],
mayor: {
        name: "Jim Wehrle"
}}
{name: "New York",
popujatiuon: 22200000,
```

last_sensus: ISODate("2009-07-31"),
famous_for: ["status of liberty", "food"],
mayor: {
name: "Michael Bloomberg",
party: "I"}}
{name: "Portland",
popujatiuon: 528000,
last_sensus: ISODate("2009-07-20"),
famous_for: ["beer", "food"],
mayor: {
name: "Sam Adams",
party: "D"}}

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, использую второй способ автоматического связывания.
3. Проверьте содержание коллекции единорогов.

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.

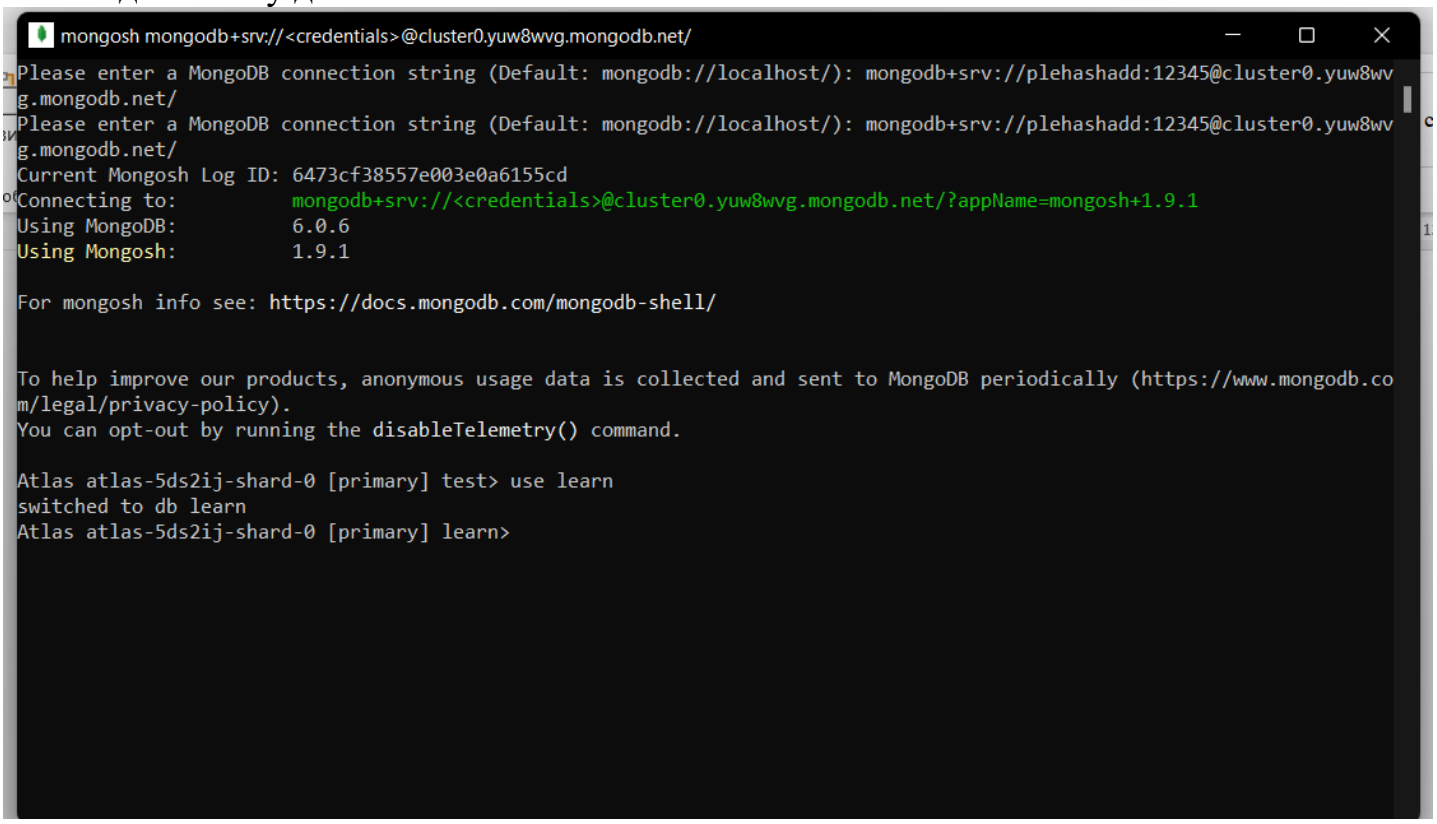
Практическое задание 8.3.4:

1. Создайте объемную коллекцию numbers, задействовав курсор:
2. for(i = 0; i < 100000; i++){db.numbers.insert({value: i})}
3. Выберите последних четыре документа.
4. Проанализируйте план выполнения запроса 2. Сколько потребовалось времени на выполнение запроса? (по значению параметра executionTimeMillis)
5. Создайте индекс для ключа value.
6. Получите информацию о всех индексах коллекции numbres.
7. Выполните запрос 2.
8. Проанализируйте план выполнения запроса с установленным индексом. Сколько потребовалось времени на выполнение запроса?
9. Сравните время выполнения запросов с индексом и без. Дайте ответ на вопрос: какой запрос более эффективен?

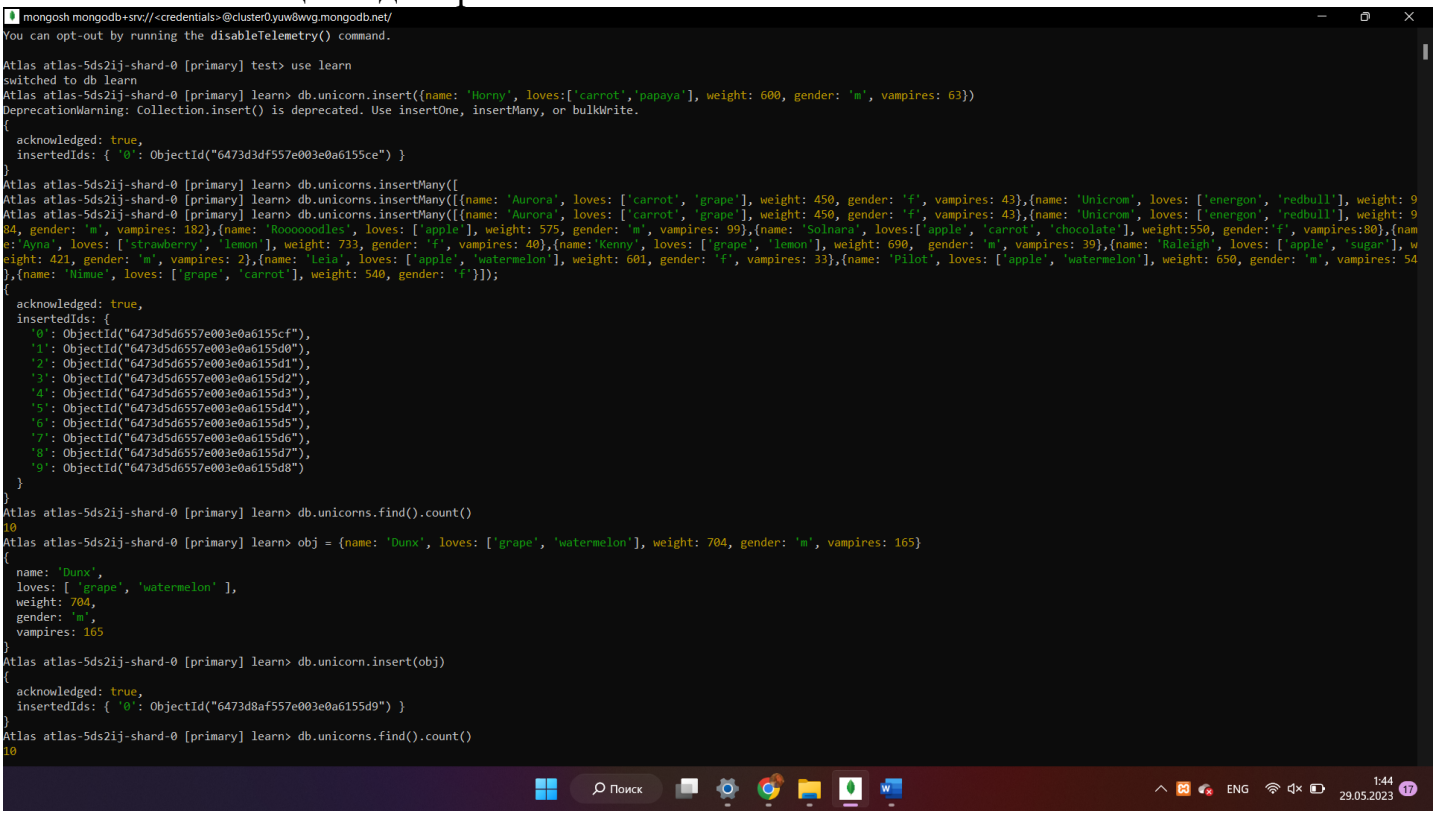
Выполнение:

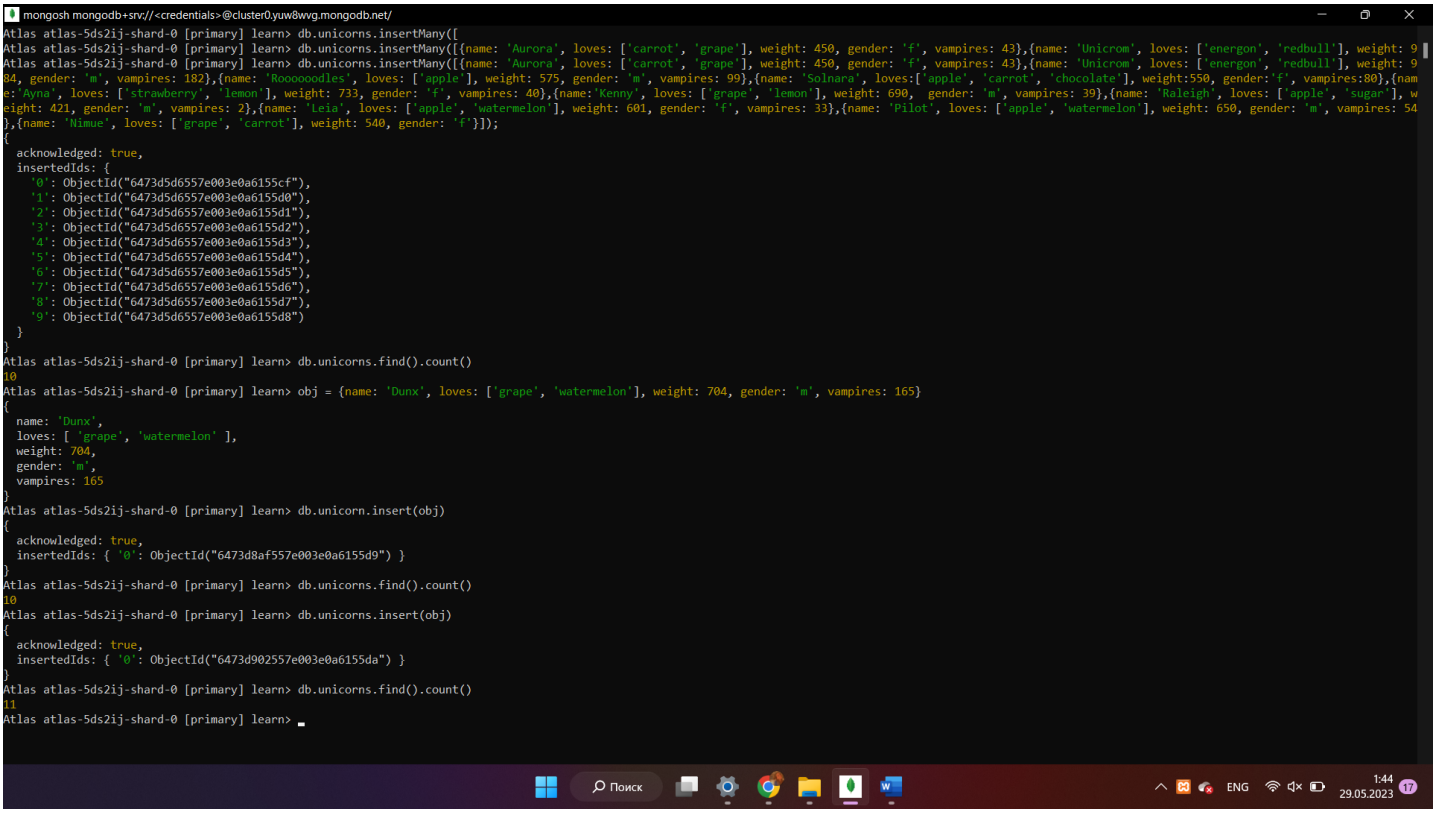
Практическое задание 8.1.1:

1. Создайте базу данных learn.

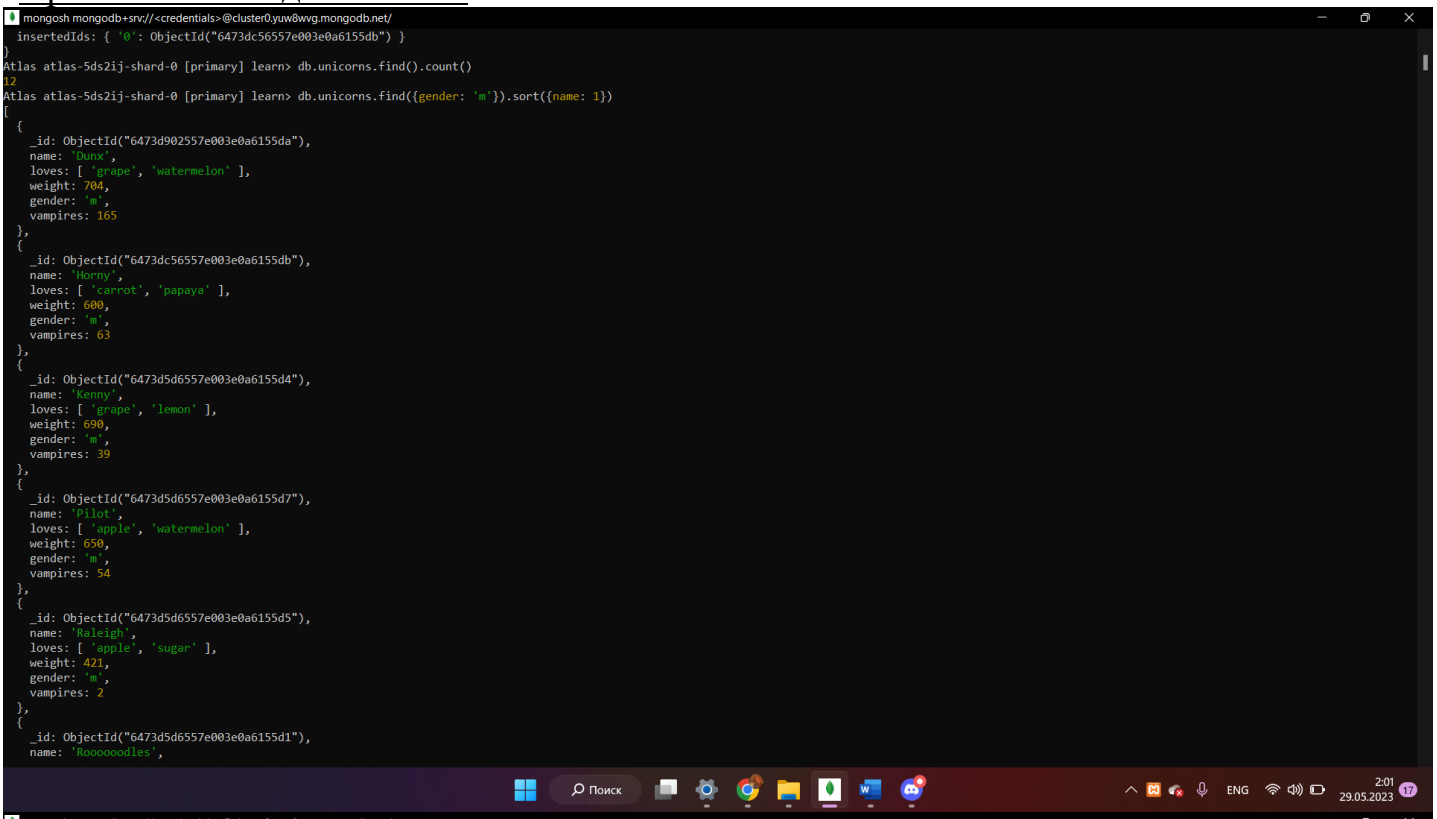


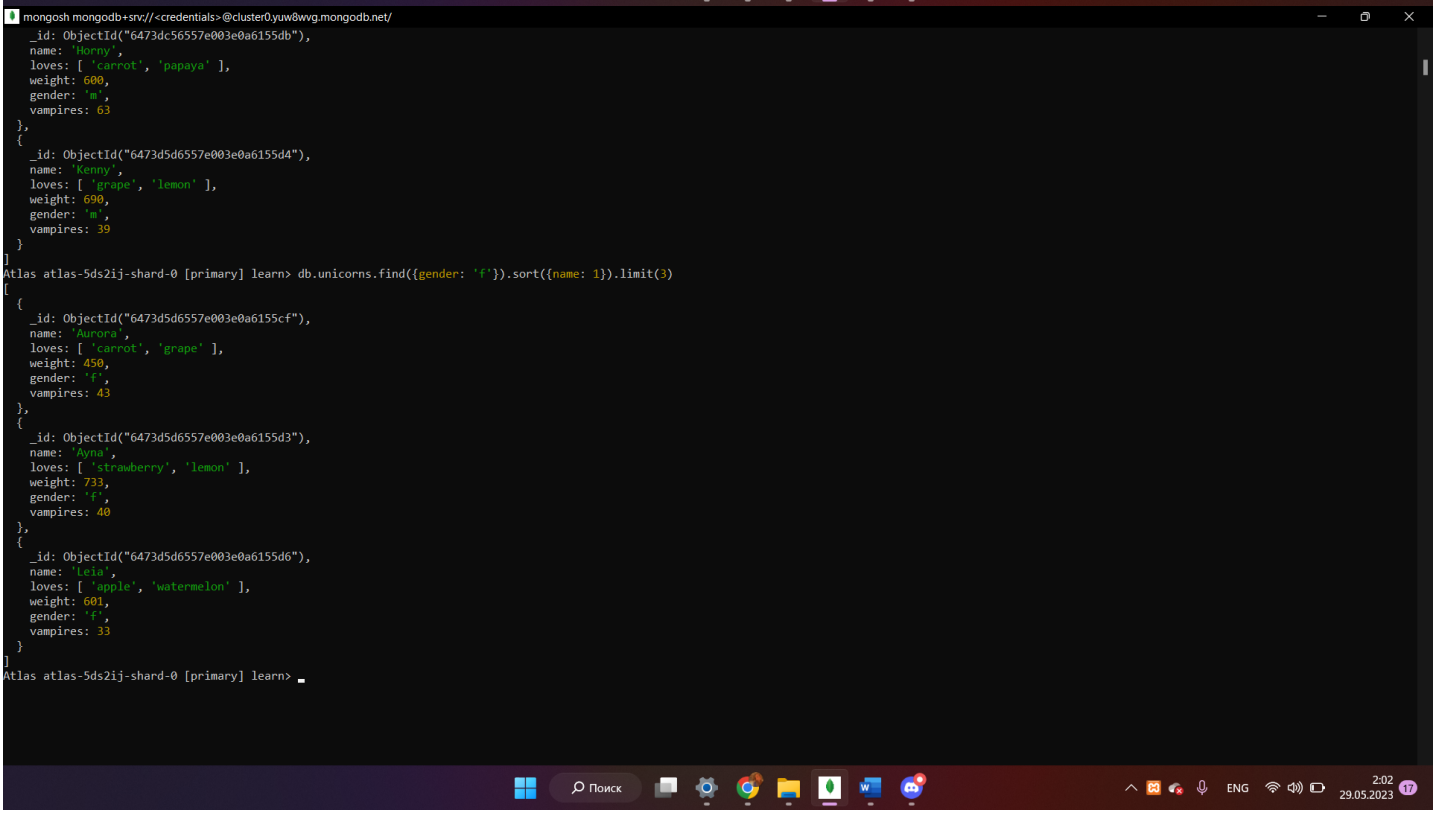
2. Заполните коллекцию единорогов unicorns:

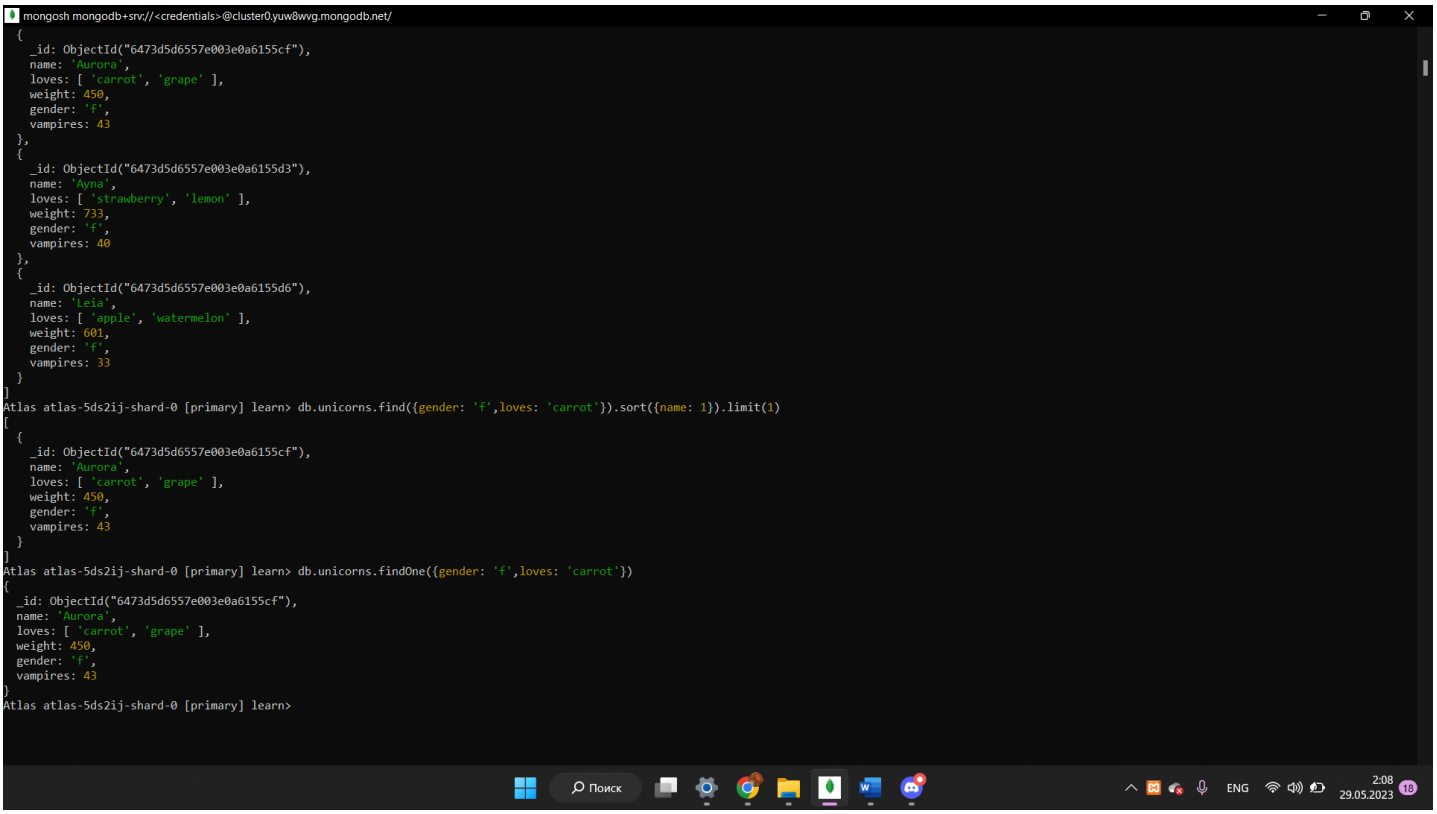



mongosh mongodb+srv://<credentials>@cluster0.yuw8wvg.mongodb.net/
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.insertMany([
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.insertMany([{name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43},{name: 'Unicrom', loves: ['energon', 'redbull'], weight: 9
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.insertMany([{name: 'Aurora', loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43},{name: 'Unicrom', loves: ['energon', 'redbull'], weight: 9
84, gender: 'm', vampires: 182},{name: 'Roooooodles', loves: ['apple'], weight: 575, gender: 'm', vampires: 99},{name: 'Solnara', loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80},{nam
e:'Ayna', loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40},{name:'Kenny', loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39},{name: 'Raleigh', loves: ['apple', 'sugar'], w
eight: 421, gender: 'm', vampires: 2},{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'}]);
{
acknowledged: true,
insertedIds: {
'0': ObjectId("6473d5d6557e003e0a6155cf"),
'1': ObjectId("6473d5d6557e003e0a6155d0"),
'2': ObjectId("6473d5d6557e003e0a6155d1"),
'3': ObjectId("6473d5d6557e003e0a6155d2"),
'4': ObjectId("6473d5d6557e003e0a6155d3"),
'5': ObjectId("6473d5d6557e003e0a6155d4"),
'6': ObjectId("6473d5d6557e003e0a6155d5"),
'7': ObjectId("6473d5d6557e003e0a6155d6"),
'8': ObjectId("6473d5d6557e003e0a6155d7"),
'9': ObjectId("6473d5d6557e003e0a6155d8")
}
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find().count()
10
Atlas atlas-5ds2ij-shard-0 [primary] learn> obj = {name: 'Dunx', loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165}
{
name: 'Dunx',
loves: ['grape', 'watermelon'],
weight: 704,
gender: 'm',
vampires: 165
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorn.insert(obj)
{
acknowledged: true,
insertedIds: { '0': ObjectId("6473d8af557e003e0a6155d9") }
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find().count()
10
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.insert(obj)
{
acknowledged: true,
insertedIds: { '0': ObjectId("6473d902557e003e0a6155da") }
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find().count()
11
Atlas atlas-5ds2ij-shard-0 [primary] learn>
Поиск
ENG
1:44
29.05.2023

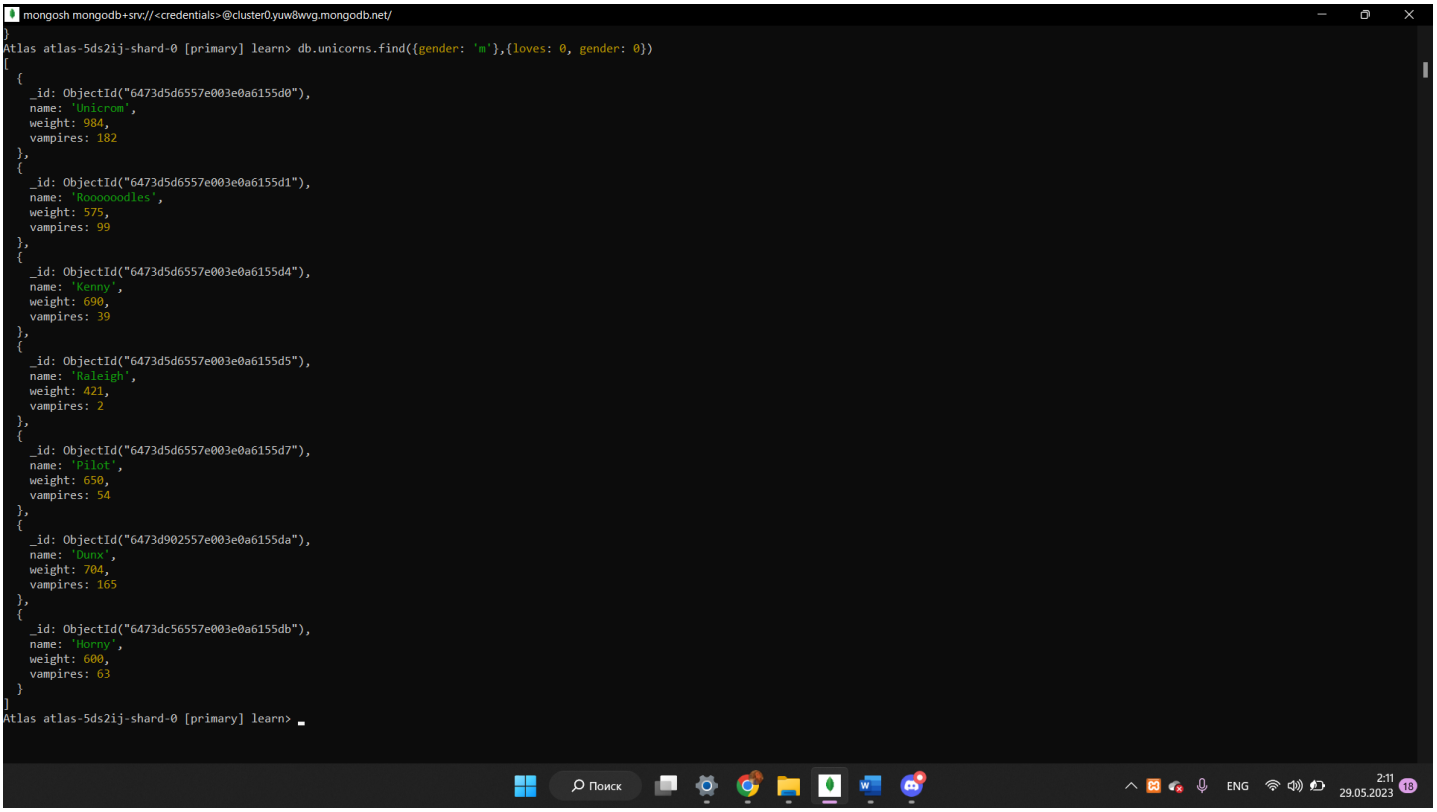
Практическое задание 8.1.2:



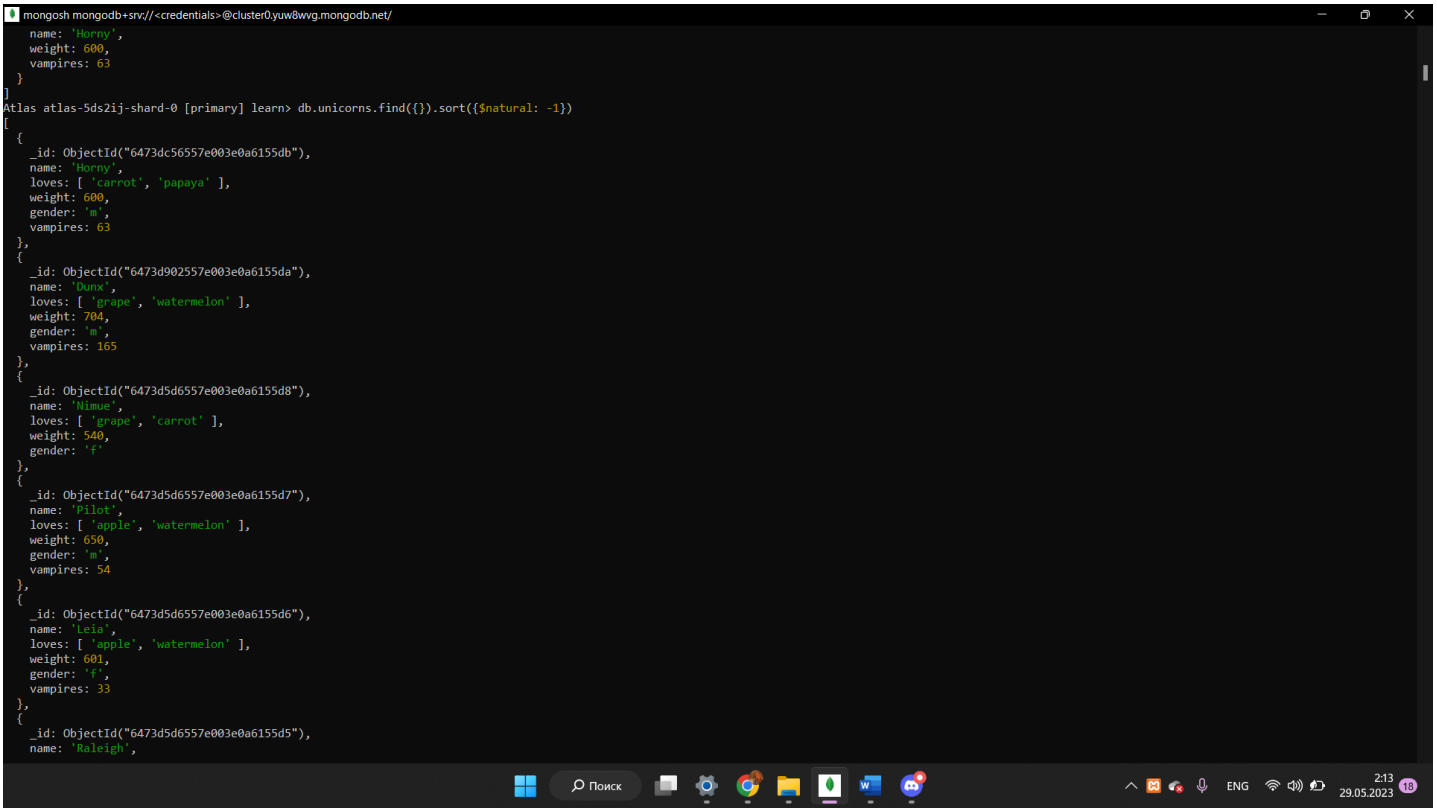



mongosh mongodb+srv://<credentials>@cluster0.yuw8wvg.mongodb.net/
{
_id: ObjectId("6473d5d6557e003e0a6155cf"),
name: 'Aurora',
loves: ['carrot', 'grape'],
weight: 450,
gender: 'f',
vampires: 43
},
{
_id: ObjectId("6473d5d6557e003e0a6155d3"),
name: 'Ayna',
loves: ['strawberry', 'lemon'],
weight: 733,
gender: 'f',
vampires: 40
},
{
_id: ObjectId("6473d5d6557e003e0a6155d6"),
name: 'Leia',
loves: ['apple', 'watermelon'],
weight: 601,
gender: 'f',
vampires: 33
}
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'f',loves: 'carrot'}).sort({name: 1}).limit(1)
[
{
_id: ObjectId("6473d5d6557e003e0a6155cf"),
name: 'Aurora',
loves: ['carrot', 'grape'],
weight: 450,
gender: 'f',
vampires: 43
}
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.findOne({gender: 'f',loves: 'carrot'})
{
_id: ObjectId("6473d5d6557e003e0a6155cf"),
name: 'Aurora',
loves: ['carrot', 'grape'],
weight: 450,
gender: 'f',
vampires: 43
}
Atlas atlas-5ds2ij-shard-0 [primary] learn>
Поиск
ENG
2:08
29.05.2023
18

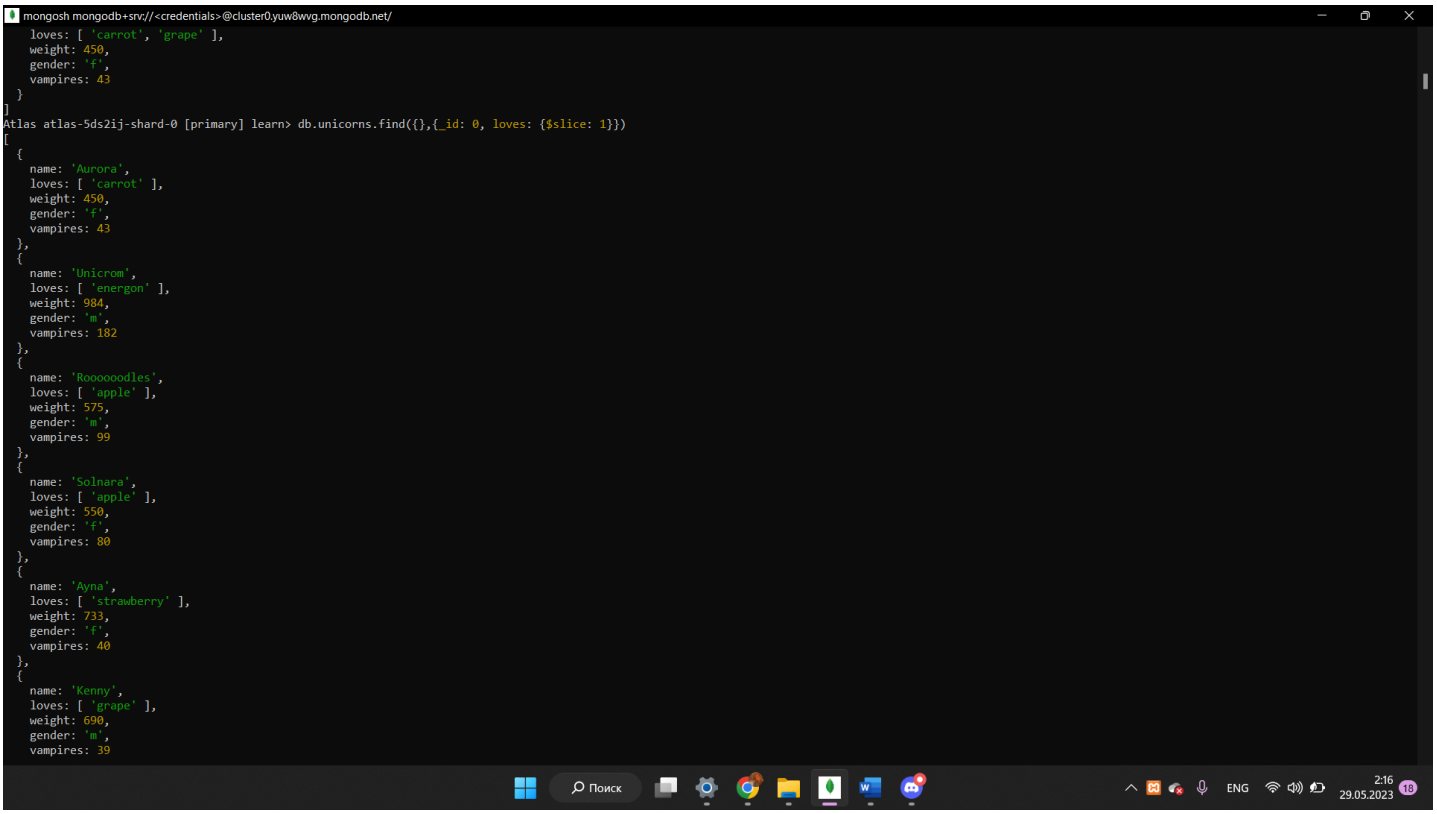
Практическое задание 8.1.3:

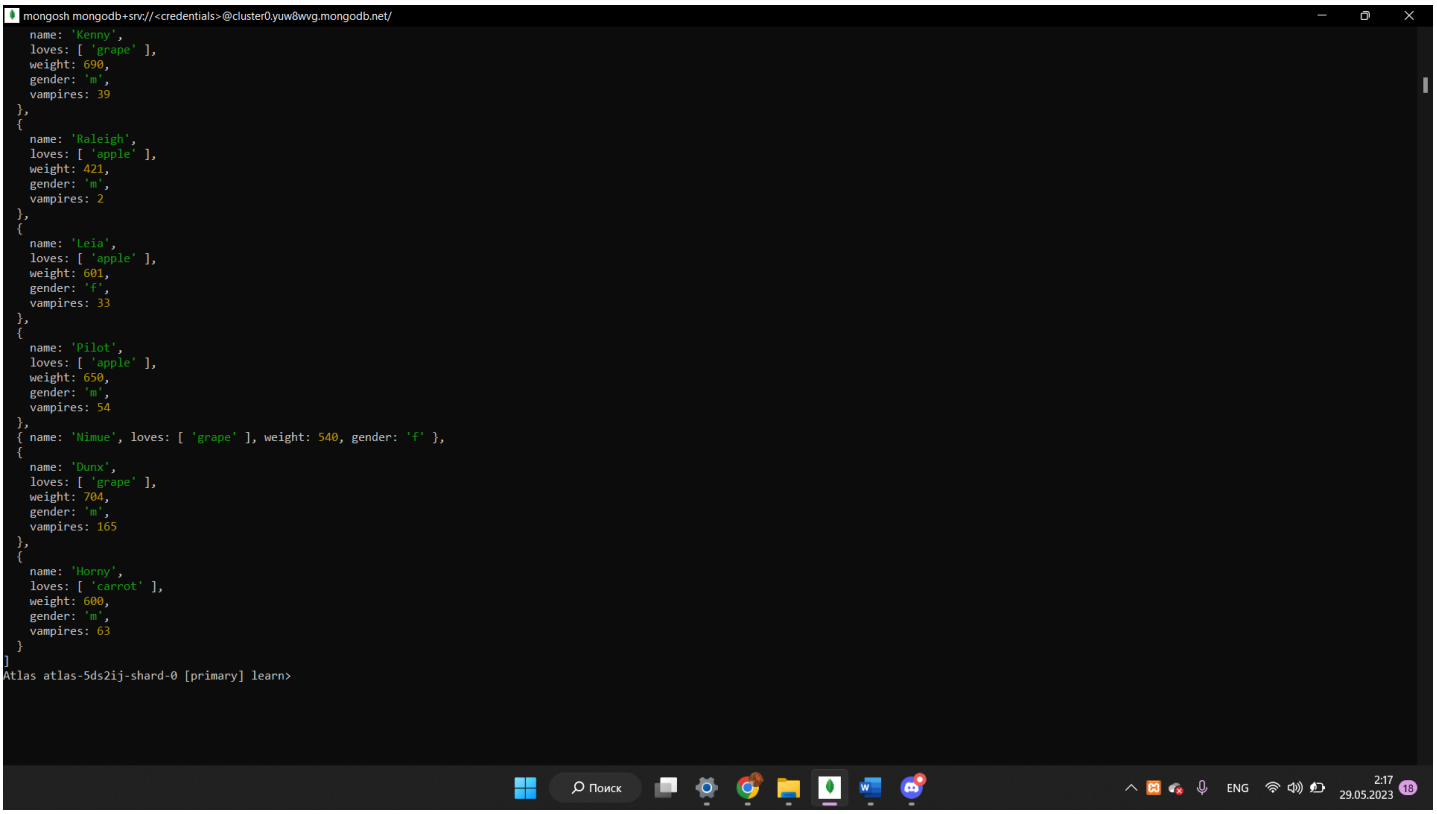


Практическое задание 8.1.4:



Практическое задание 8.1.5:




mongosh mongodb+srv://<credentials>@cluster0.yuw8wvg.mongodb.net/
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
},
{
name: 'Horny',
loves: ['carrot'],
weight: 600,
gender: 'm',
vampires: 63
}
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
Поиск
ENG
2:17
29.05.2023

Практическое задание 8.1.6:

```
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'f', weight: {$gt: 500, $lt: 700}},{_id: 0, name: 1, weight: 1})
[
  { name: 'Solnara', weight: 550 },
  { name: 'Leia', weight: 601 },
  { name: 'Nimue', weight: 540 }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.1.7:

```
  { name: 'Nimue', weight: 540 }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'm', weight: {$gt: 500},loves: {$all: ['lemon', 'grape']}},{_id: 0, name: 1, weight: 1})
[ { name: 'Kenny', weight: 690 } ]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.1.8:

```
[ { name: 'Nimue' } ]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.1.9:

```
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'f', vampires: {$exists: false}},{_id: 0, name: 1})
[ { name: 'Nimue' } ]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'm'},{_id: 0, name: 1, loves: {$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: 'Roooooodles', loves: [ 'apple' ] },
  { name: 'Unicrom', loves: [ 'energon' ] }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.1:

```
mongosh mongodb+srv://<credentials>@cluster0.yuw8wvg.mongodb.net/
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find().count()
3
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.deleteMany({"mayor party": {$exists: false}})
{ acknowledged: true, deletedCount: 3 }
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find().count()
0
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.drop()
true
Atlas atlas-5ds2ij-shard-0 [primary] learn> show collections
unicorn
unicorns
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.createCollection('towns')
{ ok: 1 }
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.insert({name: "Punxsutawney ", populatiuon: 6200, last_sensus: ISODate("2008-01-31"),famous_for: ["phil the groundhog"],mayor: {name:"Jim Wehrle"}})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6473e7d3d9fad5d46a378f0b") }
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.insert({name: "New York", populatiuon: 22200000, last_sensus: ISODate("2009-07-31"),famous_for: ["status of liberty", "food"],mayor: {name:"Michael Bloomberg"
, party: "I"}})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6473e7dfd9fad5d46a378f0c") }
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.insert({name: "Portland", populatiuon: 52800, last_sensus: ISODate("2009-07-20"),famous_for: ["beer", "food"],mayor: {name:"Sam Adams", party: "D"}})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6473e7e7d9fad5d46a378f0d") }
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find({"mayor.party": "I"},{_id: 0, name: 1, mayor: 1})
[
  {
    name: 'New York',
    mayor: { name: 'Michael Bloomberg', party: 'I' }
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find({"mayor.party": {$exist: false}},{_id: 0, name: 1, mayor: 1})
MongoServerError: unknown operator: $exist
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find({"mayor.party": {$exists: false}},{_id: 0, name: 1, mayor: 1})
[ { name: 'Punxsutawney ', mayor: { name: 'Jim Wehrle' } } ]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.2:

```
Atlas atlas-5ds2ij-shard-0 [primary] learn> fn = function() {return this.gender == 'm';}
[Function: fn]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find(fn)
MongoInvalidArgumentError: Query filter must be a plain object or ObjectId
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Не выводит, потому что используется MondoDB Atlas, а не локальное серверное хранение MongoDB Community.

```
Atlas atlas-5ds2ij-shard-0 [primary] learn> var cursor = db.unicorns.find({"$where": fn}).sort({name: 1}).limit(2);null;
null
Atlas atlas-5ds2ij-shard-0 [primary] learn> cursor.forEach(function(obj) {print(obj)});
MongoServerError: $where not allowed in this atlas tier
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.3:

```
MongoServerError: $where not allowed in this atlas tier
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'f', weight: {$gt: 500, $lt: 600}}).count()
2
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.4:

```
2
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.distinct("loves")
[
  'apple',      'carrot',
  'chocolate',  'energon',
  'grape',      'lemon',
  'papaya',     'redbull',
  'strawberry', 'sugar',
  'watermelon'
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.5:

```
MongoServerError: unknown group operator '$summ'
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.aggregate({"$group": {_id: "$gender", count: {$sum: 1 }}})
[ { _id: 'f', count: 5 }, { _id: 'm', count: 7 } ]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

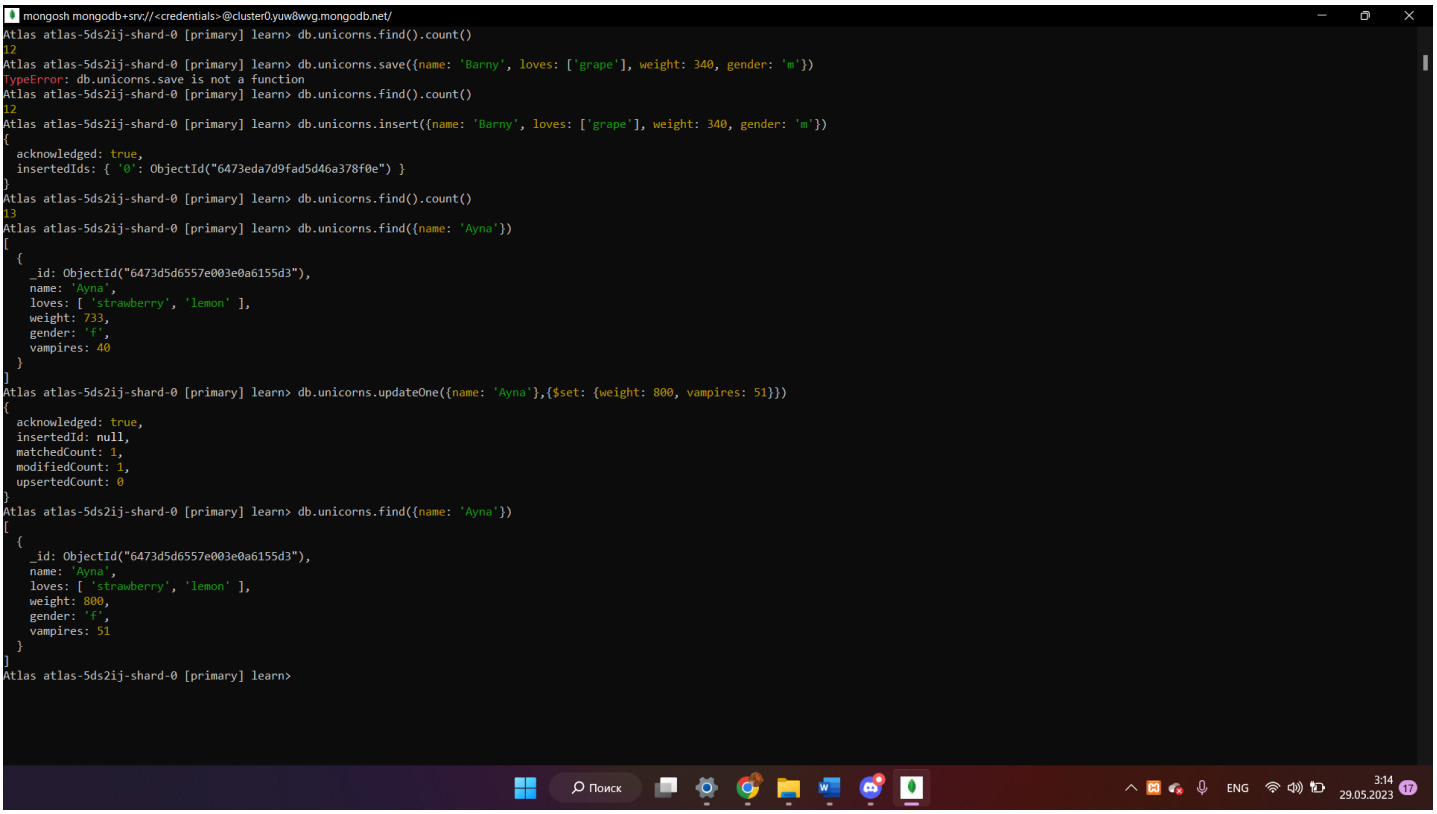
Практическое задание 8.2.6:

1. Выполнить команду:

db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})

```
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.aggregate({"$group": {_id: "$gender", count: {$summ: 1 }}})
MongoServerError: unknown group operator '$summ'
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.aggregate({"$group": {_id: "$gender", count: {$sum: 1 }}})
[ { _id: 'f', count: 5 }, { _id: 'm', count: 7 } ]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find().count()
12
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.save({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
TypeError: db.unicorns.save is not a function
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find().count()
12
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.insert({name: 'Barny', loves: ['grape'], weight: 340, gender: 'm'})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6473eda7d9fad5d46a378f0e") }
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find().count()
13
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.7:



Практическое задание 8.2.8:

```
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Raleigh'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155d5"),
    name: 'Raleigh',
    loves: [ 'apple', 'sugar' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.updateOne({name: 'Raleigh'},{$set: {loves: ['redbull']}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Raleigh'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155d5"),
    name: 'Raleigh',
    loves: [ 'redbull' ],
    weight: 421,
    gender: 'm',
    vampires: 2
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.9:

```
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'm'}, {_id: 0, name: 1, vampires: 1}).sort({name: 1}).toArray()
[
  { name: 'Barny' },
  { name: 'Dunx', vampires: 165 },
  { name: 'Horny', vampires: 63 },
  { name: 'Kenny', vampires: 39 },
  { name: 'Pilot', vampires: 54 },
  { name: 'Raleigh', vampires: 2 },
  { name: 'Roooooodles', vampires: 99 },
  { name: 'Unicrom', vampires: 182 }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.updateMany({gender: 'm'}, {$inc:{vampires: 5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 8,
  modifiedCount: 8,
  upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({gender: 'm'}, {_id: 0, name: 1, vampires: 1}).sort({name: 1}).toArray()
[
  { name: 'Barny', vampires: 5 },
  { name: 'Dunx', vampires: 170 },
  { name: 'Horny', vampires: 68 },
  { name: 'Kenny', vampires: 44 },
  { name: 'Pilot', vampires: 59 },
  { name: 'Raleigh', vampires: 7 },
  { name: 'Roooooodles', vampires: 104 },
  { name: 'Unicrom', vampires: 187 }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.10:

```
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find({name: 'Portland'})

[
  {
    _id: ObjectId("6473e7e7d9fad5d46a378f0d"),
    name: 'Portland',
    populatiuon: 52800,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams', party: 'D' }
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.updateOne({name: 'Portland'}, {$unset: {"mayor.party": 1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find({name: 'Portland'})

[
  {
    _id: ObjectId("6473e7e7d9fad5d46a378f0d"),
    name: 'Portland',
    populatiuon: 52800,
    last_sensus: ISODate("2009-07-20T00:00:00.000Z"),
    famous_for: [ 'beer', 'food' ],
    mayor: { name: 'Sam Adams' }
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.11:

```
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Pilot'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155d7"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.updateOne({name: 'Pilot'}, {$push: {loves: 'chocolate'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Pilot'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155d7"),
    name: 'Pilot',
    loves: [ 'apple', 'watermelon', 'chocolate' ],
    weight: 650,
    gender: 'm',
    vampires: 59
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.12:

```
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Aurora'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155cf"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.updateOne({name: 'Aurora'}, {$addToSet: {loves: {$each: ['sugar', 'lemon']}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Aurora'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155cf"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.2.13:

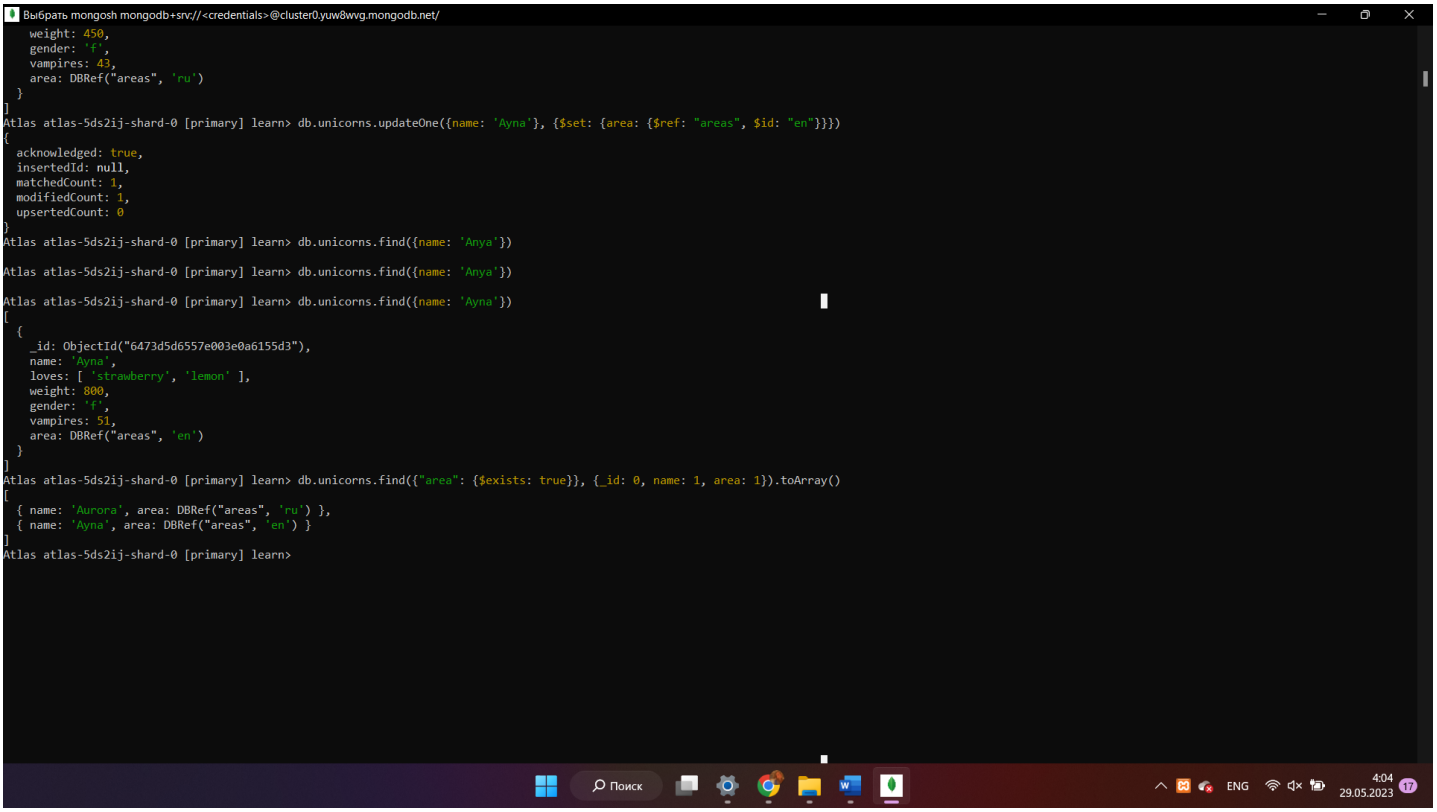
```
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find().count()
3
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.deleteMany({"mayor.party" : {$exists: false}})
{ acknowledged: true, deletedCount: 2 }
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.find().count()
1
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.towns.drop()
true
Atlas atlas-5ds2ij-shard-0 [primary] learn> show collections
unicorn
unicorns
Atlas atlas-5ds2ij-shard-0 [primary] learn>
```

Практическое задание 8.3.1:

```
Выбрать mongosh mongodb+srv://<credentials>@cluster0.yuw8wvg.mongodb.net/
unicorns
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.createCollection('areas')
{ ok: 1 }
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.areas.insertMany([{_id: 'ru', name: 'Russia', description: 'cold and snowy'},{_id: 'en', name: 'America', description: 'rainy but cool'}]);
{ acknowledged: true, insertedIds: { '0': 'ru', '1': 'en' } }
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.areas.find()
[
  { _id: 'ru', name: 'Russia', description: 'cold and snowy' },
  { _id: 'en', name: 'America', description: 'rainy but cool' }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.updateOne({name: 'Aurora'}, {$set: {area: {$ref: "areas", $id: "ru"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Aurora'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155cf"),
    name: 'Aurora',
    loves: [ 'carrot', 'grape', 'sugar', 'lemon' ],
    weight: 450,
    gender: 'f',
    vampires: 43,
    area: DBRef("areas", 'ru')
  }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.updateOne({name: 'Ayna'}, {$set: {area: {$ref: "areas", $id: "en"}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Anya'})

Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Anya'})

Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Ayna'})
[
  {
    _id: ObjectId("6473d5d6557e003e0a6155d3"),
    name: 'Ayna',
    loves: [ 'strawberry', 'lemon' ],
    weight: 800,
```


Выбрать mongosh mongodb+srv://<credentials>@cluster0.yuw8wvg.mongodb.net/
weight: 450,
gender: 'f',
vampires: 43,
area: DBRef("areas", 'ru')
}
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.updateOne({name: 'Ayna'}, {$set: {area: {$ref: "areas", $id: "en"}}})
{
acknowledged: true,
insertedId: null,
matchedCount: 1,
modifiedCount: 1,
upsertedCount: 0
}
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Anya'})
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Anya'})
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({name: 'Ayna'})
[
{
_id: ObjectId("6473d5d6557e003e0a6155d3"),
name: 'Ayna',
loves: ['strawberry', 'lemon'],
weight: 800,
gender: 'f',
vampires: 51,
area: DBRef("areas", 'en')
}
]
Atlas atlas-5ds2ij-shard-0 [primary] learn> db.unicorns.find({"area": {$exists: true}}, {_id: 0, name: 1, area: 1}).toArray()
[
{ name: 'Aurora', area: DBRef("areas", 'ru') },
{ name: 'Ayna', area: DBRef("areas", 'en') }
]
Atlas atlas-5ds2ij-shard-0 [primary] learn>
Поиск
ENG
4:04
29.05.2023