1. Start C:\Program Files\MongoDB\Server\6.0\bin\mongod.exe
2. Start C:\Program Files\mongosh-1.6.0\bin\mongosh.exe

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **show all databases** | | show dbs |
| **create database** | | use cscorner |
| **show current database** | | db |
| **create collection in chose db** | | db.createCollection(“students”) |
| **insert in collection in chosen db** | | db.students.insertOne({"rollno":1, "name":"sunita"}) |
| **show collections in chosen db** | | show collections |
| **find all documents in collection** | | db.students.find() |
| **find all documents in collection** | | db.students.find().pretty() |
| **rename collection** | | db.unicorns.renameCollection("unicorns2") |
| **statistics about collection** | | db.unicorns2.stats() |
| **drop collection** | | db.students.drop() |
| **drop database** | | db.dropDatabase() |

Выборка

* 1. Найти всех пользователей с параметрами в скобках

db.users.find({name: "Tom", age: 32})

* 1. Найти только одного пользователя с параметрами в скобках

db.users.findOne({name: "Tom", age: 32})

* 1. Найти пользователей с параметрами в первых скобках и не включить в выборку всё false (0) из вторых скобок (не показывать) и включить всё true (1)

db.users.find({name: "Tom"}, {age: true, \_id: false})

* 1. Ограничить вывод пользователей до 3

db.users.find().limit(3)

* 1. Пропустить первых 3 пользователей

db.users.find().skip(3)

Сортировка

1. Сортировка по имени, по возрастанию 1, по убыванию -1

db.users.find().sort({name: 1})

2. Сортировка по имени, $natural – документы передаются в том порядке, в каком они были добавлены в коллекцию (1), либо в обратном порядке (-1)

db.users.find().sort({ $natural: -1 }).limit(5)

3. Оператор $slice принимает два параметра.

Первый параметр указывает на общее количество возвращаемых документов. Второй параметр необязательный, но если он используется, тогда первый параметр указывает на смещение относительно начала (как функция skip), а второй - на ограничение количества извлекаемых документов (как функция limit).

Из языков в запрос попадает только один с начала массива языков

db.users.find ({name: "Tom"}, {languages: {$slice : 1}})

Из языков в запрос попадает только один с конца массива языков

db.users.find ({name: "Tom"}, {languages: {$slice : -1}});

В запросе skip один язык с конца и выводится только один

db.users.find ({name: "Tom"}, {languages: {$slice : [-1, 1]}});

Логические операторы

1. В выборку попадут все люди, у которых возраст меньше 30

db.users.find ({age: {$lt : 30}})

2. В выборку попадут все люди, у которых возраст больше 30 и меньше 50

db.users.find ({age: {$gt : 30, $lt: 50}})

3. В выборку попадут все люди, у которых возраст не 22

db.users.find ({age: {$ne : 22}})

4. В выборку попадут все люди, у которых возраст 22 или 32

db.users.find ({age: {$in : [22, 32]}}

5. В выборку попадут все люди, у которых возраст не 22 и не 32

db.users.find ({age: {$nin : [22, 32]}})

6. В выборку попадут все люди, у которых возраст и 22, и 32

db.users.find ({age: {$all : [22, 32]}})

В выборку попадут все люди, у которых возраст 22

db.users.find ({age: {$all : [22]}})

7. В выборку попадут все люди, у которых имя Том + или возраст 22 или язык немецкий

db.users.find ({name: "Tom", $or : [{age: 22}, {languages: "german"}]})

8. В выборку попадут все люди, у которых есть поле company

db.users.find ({company: {$exists:true}})

**Практическое задание 1**

*Создайте базу данных learn.*

use learn

*Заполните коллекцию единорогов unicorns:*

db.unicorns.insertOne({name: 'Horny', dob: new Date(1992,2,13,7,47), loves: ['carrot','papaya'], weight: 600, gender: 'm', vampires: 63});

db.unicorns.insertOne({name: 'Aurora', dob: new Date(1991, 0, 24, 13, 0), loves: ['carrot', 'grape'], weight: 450, gender: 'f', vampires: 43});

db.unicorns.insertOne({name: 'Unicrom', dob: new Date(1973, 1, 9, 22, 10), loves: ['energon', 'redbull'], weight: 984, gender: 'm', vampires: 182});

db.unicorns.insertOne({name: 'Roooooodles', dob: new Date(1979, 7, 18, 18, 44), loves: ['apple'], weight: 575, gender: 'm', vampires: 99});

db.unicorns.insertOne({name: 'Solnara', dob: new Date(1985, 6, 4, 2, 1), loves:['apple', 'carrot', 'chocolate'], weight:550, gender:'f', vampires:80});

db.unicorns.insertOne({name:'Ayna', dob: new Date(1998, 2, 7, 8, 30), loves: ['strawberry', 'lemon'], weight: 733, gender: 'f', vampires: 40});

db.unicorns.insertOne({name:'Kenny', dob: new Date(1997, 6, 1, 10, 42), loves: ['grape', 'lemon'], weight: 690, gender: 'm', vampires: 39});

db.unicorns.insertOne({name: 'Raleigh', dob: new Date(2005, 4, 3, 0, 57), loves: ['apple', 'sugar'], weight: 421, gender: 'm', vampires: 2});

db.unicorns.insertOne({name: 'Leia', dob: new Date(2001, 9, 8, 14, 53), loves: ['apple', 'watermelon'], weight: 601, gender: 'f', vampires: 33});

db.unicorns.insertOne({name: 'Pilot', dob: new Date(1997, 2, 1, 5, 3), loves: ['apple', 'watermelon'], weight: 650, gender: 'm', vampires: 54});

db.unicorns.insertOne({name: 'Nimue', dob: new Date(1999, 11, 20, 16, 15), loves: ['grape', 'carrot'], weight: 540, gender: 'f'});

*Используя второй способ, вставьте в коллекцию единорогов документ:*

newUnicorn = ({name: 'Dunx', dob: new Date(1976, 6, 18, 18, 18), loves: ['grape', 'watermelon'], weight: 704, gender: 'm', vampires: 165});

db.unicorns.insertOne(newUnicorn);

*Проверьте содержимое коллекции с помощью метода find*

db.unicorns.find();

**Практическое задание 2**

*Сформируйте запросы для вывода списков самцов и самок единорогов: 1Ограничьте список самок первыми тремя особями. Отсортируйте списки по имени.*

db.unicorns.find({gender: 'f'}).limit(3).sort({name: 1});

*2Найдите всех самок, которые любят carrot. Ограничьте этот список первой особью с помощью функций findOne и limit*

db.unicorns.find({gender: 'f', loves: 'carrot'}).limit(1);

db.unicorns.findOne({gender: 'f', loves: 'carrot'});

**Практическое задание 3**

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

db.unicorns.find({gender: 'm'}, {gender: false, dob: false})

db.unicorns.find({gender: 'm'}, {gender: false, dob: false})

**Практическое задание 4**

*Вывести список единорогов в обратном порядке добавления.*

db.unicorns.find().sort({ $natural: -1})

***Практическое задание 5***

*Вывести список единорогов с названием первого любимого фрукта, исключив идентификатор.*

db.unicorns.find({}, { \_id: false, loves: {$slice: 1} })

**Практическое задание 5**

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

db.unicorns.find({weight: {$gte : 500, $lt : 700}}, {id: false}) – >= <

db.unicorns.find({weight: {$gte : 500, $lte : 700}}, {id: false}) – >= <=

**Практическое задание 6**

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

db.unicorns.find( {weight: {$gt:500}, loves: {$all: ['grape', 'lemon']}}, {\_id:false} )

**Практическое задание 7**

*Найти всех единорогов, не имеющих ключ vampires.*

db.unicors.find( { vampires: {$exists:false} } )