Втори стъпки в MongoDB

Юрий Донев

Съдържание

- Скриптове (продължение)
- Заявки (продължение)
- Атомарни операции

Скриптове

Помощни функции в mongo shell	Еквивалент в скриптов файл
show dbs, show databases	db.adminCommand('listDatabases')
use <db>	db = db.getSiblingDB('<db>')
show collections	db.getCollectionNames()
show users	db.system.users.find()
show log <logname>	db.adminCommand({ 'getLog' : '<logname>' })
show logs	db.adminCommand({ 'getLog' : '*' })

it	cursor = db.collection.find()
	if (cursor.hasNext()){
	  cursor.next();
	}

Малко повече за find

- read (retrieve) частта в CRUD
- db.collection.find(<criteria>, <projection>)
  - criteria
  - projection
- db.collection.findOne(<criteria>, <projection>)
  - criteria
  - projection

Find: курсори

Базата данни връща резултати от операцията find чрез курсор

```
var cursor = db.collection.find()
```

- по резултатите от заявката се итерира чрез функцията next
- проверката за съществуване на следващ резултат става чрез функцията hasNext

Find: курсори

```
var cursor = db.books.find({pages:{$lt:1000}})
```

- при изпълняване на горния ред заявката не е върнала резултати
- при извикване на cursor.hasNext() или на cursor.next() се връща отговор на заявката, който съдържа по-малкото като памет измежду първите 100 резултата и 4 MB резултати
- ако има повече резултати се прави повторна заявка към базата

Find

Опции на командата find:

- sort - сортира върнатите резултати по зададено поле

  sort({ field1: value1, field2: value2, .. })

  * ако value=1 връща стойностите сортирани в нарастващ ред

  * ако value=-1 връща стойностите сортиране в намаляващ ред

Find

Опции на командата find:

- limit - ограничава броя на върнати резултати

  limit(n)
- skip - прескача зададания брой резултати

  skip(n)

  Внимание: прескачане на голям брой резултати може да забави заявката

Сравнения

Важат за произволни видове заявки

- $gt, $gte

  {field: {$gt: value} } или {field: {$gte: value} }

- $lt, $lte

  {field: {$lt: value} } или {field: {$lte: value} }

- $ne

  {field: {$ne: value} }

Регулярни изрази

$regex - предоставя възможност за намиране на низове по зададен регулярен израз

- { field: /reg.*rene.pression/i }}
- { field: { $regex: 'reg.*rene.pression', $options: 'i' } }

$regex използва механизма за регулярни изрази на Perl (Perl Compatible Regular Expressions - http://perldoc.perl.org/perlre.html)

Регулярни изрази

Флагове за опции на $regex:

- i - не отчита малки и големи букви
- m - прилага регулярния израз върху повече от един ред от полето (ако стойността на полето има повече от един ред)
- s - позволява на ‘.’ да замества произволен символ, включително знак за нов ред
- x - не отчита празните символи

Последните две опции работят само с втория синтаксис на $regex

Работа със списъчни стойности

- прилагане на обикновен find
- $in

  { field: { $in: [<value1>, <value2>, ... <valueN>] } }

  * търсене на стойност на поле измежду няколко възможни

  * търсене на елемент на списък измежду няколко възможни

  * търсене на стойност по повече от един регулярен израз

Работа със списъчни стойности

- $nin

  { field: { $nin: [<value1>, <value2>, ... <valueN>] } }

- $all

  { field: { $all: [<value> , <value1> ...] }

Връща точно съвпадение по зададените стойности

Работа със списъчни стойности

- Проектиране на списък
  - $slice - db.collection.find({ field: value }, { array: {$slice: count } })

  Връща проекция на документа с променен списък. Не променя оригиналния документ
  - $elemMatch

Булеви сравнения

- $and - връща документите, които отговарят на всички условия

  { $and: [{ <expression1> }, { <expression2> } , ... , { <expressionN> }] }

  * short-circuit evaluation - ако някое от условията пропадне за даден документ, останалите условия не се проверяват

  * $and се изпълнява по подразбиране, когато имате повече от едно условие за филтриране

Булеви сравнения

- $or - връща документите, които отговарят на поне едно от зададените условия

  { $or: [{ <expression1> }, { <expression2> } , ... , { <expressionN> }] }

  * използвайте $in при повече от една проверка за едно и също поле

Булеви сравнения

- $not - връща документите, които не отговарят на даденото условие

  { field: {$not: { <expression> } } }

Използва се за отрицание на други оператори. За отрицание на стойност използвайте $ne.

Други

- count
  db.books.find(<expression>).count()
  db.books.count(<expression>)
- $exists - проверка за съществуванеfind
  {field: {$exists: true/false } }
- $where - позволява търсене с JavaScript код

Update: изтриване на поле

- $unset

```
{ $unset: {
                    field1: "",
                    field2: ""
          }
}
```

Работа със списъчни стойности

- Манипулиране на списък
  - $push - добавя елемент към списък
    - db.collection.update(<query>,
      { $push: { <field>: <value> })
  - $pop - маха елемент от списък
    - db.collection.update(<query>, { $pop: { field: 1 } });
  - $pull - маха конкретен елемент от списък, който отговаря на стойност или на заявка
    - db.collection.update(<query>, {$pull: {field: value|query}})

Атомарни операции

- Една операция е атомарна, ако:
  - друг процес или операция не разбира за промените докато всички промени не са извършени
  - ако една от промените, включени в атомарната операция пропадне, то цялата атомарна операция пропада
- MongoDB поддържа атомарни операции върху един документ
- При атомарните операции в други системи имаме заключване на данни

Атомарни операции

- При MongoDB заключването на данни би било скъпо в случай на хоризонтално скалиране на базата.
- Освен това заключването противоречи на принципите на опростеност и на бързодействие на MongoDB

Атомарни операции

```
db.collection.findAndModify({
    query, sort, remove, update,
    new, fields, upsert});
```

- намира и обновява точно един документ
- връща документа преди да бъде обновен

Атомарни операции

- query - задава параметрите за търсене
- sort - сортира резултатите, ако са повече от един и определя кой документ ще бъде променен
- new - връща променения документ
- remove - ако стойността е true, ще изтрие намерения документ
- update - задава израза, с който ще се обнови намереният документ
- upsert

Въпроси

astea SOLUTIONS