# Introduction to MongoDB & Perl

Mirko Fluher

October 25, 2011

## 1 Introduction

This is an introduction to MongoDB and Perl using the Perl driver from CPAN http://search.cpan.org/~kristina/MongoDB/ and the data found in the FREE Epub book: *A little MongoDB book* http://www.mongly.com/ by Karl Seguin.

Some basic terminoly, especially if you have a background in SQL.

- **Collection** In mongoDB, a collection or schema, is similar to SQL's table. MongoDB does not required a pre-determined table with rows and columns (records/fields).
- **Document** A document is basically what is called a record in SQL.
- **Key** The equivalent of field (column) in SQL ... except that a key must be unique and it must have a value (or null). MongoDB is like a hash-table (keys/values) Values can be a string, an integer or an array.

In the examples used in this document, *unicorns* is the name of the database and the collection, but it does not have to that way. A name for the database could easily have been **mythcreatures** with several collections such as *unicorns*, *pegasi*, *phoenixes*, *sphinxes*, *centaurs*, etc ...

In most of the following examles, the first line will refer to the mongoDB command followed by the Perl code and its output.

## 1.1 Queries

### 1.2 *Count the number of entries in the database*

```
db.unicorns.count()
```

```
#!/usr/bin/perl

use MongoDB;
use boolean;
use strict;

my $conn     = MongoDB::Connection->new;
my $db       = $conn->get_database('unicorns');
my $coll     = $db->get_collection('unicorns');

my $doc = $coll->count();
print "$doc\n";
```

```
12
```

### 1.3 *Count the number of females*

```
db.unicorns.count({'gender': 'f'})
```

```
my $doc = $coll->count({'gender' => 'f'});
print "$doc\n";
```

```
5
```

### 1.4 *Count the number of males with weight greater than 600*

```
db.unicorns.count({'gender': 'm', 'weight':{'$gt': 600}})
```

```
my $doc = $coll->count({'gender' => 'm',
                        'weight' => {'$gt' => 600}});
print "$doc\n";
```

```
4
```

### 1.5 *Show all unicorns and everything about them: name, gender, weight, vampires, dob, loves*

```
db.unicorns.find()
```

```
my $doc = $coll->find();
while (my $docs = $doc->next)
{
    printf("%-13s %-2s %-3d %-3d %s %s\n", $docs->{'name'},
        $docs-> {'gender'},
        $docs->{'weight'}, $docs->{'vampires'}, $docs->{'dob'},
        join(", ", @{$docs->{'loves'}}));
}
```

```
Horny         m  600 63  1992-03-12T21:47:00 carrot, papaya
Aurora        f  450 43  1991-01-24T02:00:00 carrot, grape
Unicrom       m  984 182 1973-02-09T11:10:00 energon, redbull
Roooooodles   m  575 99  1979-08-18T08:44:00 apple
Solnara       f  550 80  1985-07-03T16:01:00 apple, carrot, chocolate
Ayna          f  733 40  1998-03-06T21:30:00 strawberry, lemon
Kenny         m  690 39  1997-07-01T00:42:00 grape, lemon
Raleigh       m  421 2   2005-05-02T14:57:00 apple, sugar
Leia          f  601 33  2001-10-08T04:53:00 apple, watermelon
Pilot         m  650 54  1997-02-28T18:03:00 apple, watermelon
Nimue         f  540 0   1999-12-20T05:15:00 grape, carrot
Dunx          m  704 165 1976-07-18T08:18:00 grape, watermelon
```

### 1.6 *Find a unicorn with the name Leia, show name and gender*

```
db.unicorns.find({name: "Leia"})
```

```
my $doc = $coll->find({'name' => 'Leia'});
while (my $docs = $doc->next)
{
  printf("%-13s %-2s\n", $docs->{'name'}, $docs->{'gender'});
}
```

```
Leia          f
```

### 1.7 *Find all unicorns that are male and whose weight is equal or greater than 700. Show name, gender, weight and dob (date of birth)*

```
db.unicorns.find({gender: 'm', weight: {$gte: 700}})
```

```
my $doc = $coll->find({'gender' => "m",
                       'weight' => {'$gte' => 700}});
while (my $docs = $doc->next)
{
  printf("%-13s %-2s %-8d %s\n", $docs->{'name'},
    $docs-> {'gender'}, $docs->{'weight'}, $docs->{'dob'});
}
```

```
Unicrom       m  984      1973-02-09T11:10:00
Dunx          m  704      1976-07-18T08:18:00
```

### 1.8 *Find the heaviest unicorn*

```
db.unicorns.find().limit(1).sort({weight: -1})

my $doc = $coll->find->limit(1)->sort({weight => -1});

while (my $docs = $doc->next)
{
  printf("%-13s %-2s %-8d\n", $docs->{'name'},
     $docs-> {'gender'}, $docs->{'weight'});
}

Unicrom         m  984
```

### 1.9 *Find all unicorns whose name contains 'ra', showing name, gender, weight*

```
db.unicorns.find({name: /ra/i})

my $doc = $coll->find({'name' => qr/ra/i});
while (my $docs = $doc->next)
{
  printf("%-13s %-2s %-3d\n", $docs->{'name'},
    $docs->{'gender'}, $docs->{'weight'});
}

Aurora          f  450
Solnara         f  550
Raleigh         m  421
```

### 1.10 *Find a unicorn with no vampires, showing name, gender, weight, dob and vampires*

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

my $doc = $coll->find({"vampires" =>
                  {'$exists' => boolean::false}});
while (my $docs = $doc->next)
{
  printf("%-13s %-2s %-4d %s %-3d\n", $docs->{'name'},
    $docs-> {'gender'}, $docs->{'weight'},
    $docs->{'dob'}, $docs->{'vampires'});
}

Nimue           f  540  1999-12-20T05:15:00
```

### 1.11 *Find all unicorn sorted by vampires, showing name, gender and vampires*

NOTE: sort 1 = ASC, sort -1 = DESC

```
db.unicorns.find().sort({vampires: 1})

my $doc = $coll->find->sort({vampires => 1});
while (my $docs = $doc->next)
{
  printf("%-13s %-8s %-8d\n", $docs->{'name'},
    $docs-> {'gender'}, $docs->{'vampires'});
}
```

```
Nimue          f        0
Raleigh        m        2
Leia           f        33
Kenny          m        39
Ayna           f        40
Aurora         f        43
Pilot          m        54
Horny          m        63
Solnara        f        80
Roooooodles    m        99
Dunx           m        165
Unicrom        m        182
```

### 1.12 *Find all unicorns who like watermelon and chocolate, show name, gender, weight and what they love*

```
db.unicorns.find({"$or": [{'loves': 'watermelon'},
                          {'loves': 'chocolate'}]})

my $doc = $coll->find({'$or' => [{'loves' => 'watermelon'},
                                 {'loves' => 'chocolate'}]});
while (my $docs = $doc->next)
{
  printf("%-13s %-2s %-3d %s\n", $docs->{'name'},
    $docs->{'gender'}, $docs->{'weight'},
    join(", ", @{$docs->{'loves'}}));
}
```

```
Solnara        f  550 apple, carrot, chocolate
Leia           f  601 apple, watermelon
Pilot          m  650 apple, watermelon
Dunx           m  704 grape, watermelon
```

### 1.13 *You can also use $in to produce the above output*

```
db.unicorns.find({'loves':
                          {'$in': ['watermelon', 'chocolate']}})

my $doc = $coll->find({'loves' =>
                        {'$in' => ['watermelon', 'chocolate']}});
while (my $docs = $doc->next)
{
  printf("%-13s %-2s %-3d %s\n", $docs->{'name'},
    $docs->{'gender'}, $docs->{'weight'},
    join(", ", @{$docs->{'loves'}}));
}
```

# 2 Create

Creating a database OR a collection in mongoDB is very easy. Any name will do and mongoDB will do the rest! There are few restrictions: A database name should have a maximum of 64 characters, all lowercases and should not contain any of the following ('single space', '.', '$', '\', '/'). A collection name cannot contain the $ reserved character.

# 3 Insert

### 3.1 *Insert a new document into the collection*

```
db.unicorns.insert({name: 'Aurora', gender: 'f', weight: 450})

my $doc = $coll->insert({'name' => 'Aurora',
                         'gender' => 'f', 'weight' => 450});
```

# 4 Update

### 4.1 *Add a key/value to an existing document*

```
db.unicorns.update({name: "Aurora"}, {"$set": {vampires: 43}})

my $doc = $coll->update({"name" => "Aurora"},
                                {'$set' => {"vampires" => 43}})
```

### 4.2 *Change the value of an existing key*

```
db.unicorns.update({name: "Aurora"}, {"$set": {vampires: 49}})

my $doc = $coll->update({"name" => "Aurora"},
                                {'$set' => {"vampires" => 49}})
```

### 4.3 *Add/remove items to/from the array of an existing key*

Using the $set Modifier, add 'sugar' and 'carrot' elements to Roooooodles' 'loves' key array. Then, remove the element 'sugar' and add 'watermelon'. This can be done using the $push and $pop modifiers. The latter can remove elements from the end or the start of the array by using {$pop: {key: 1}} or {$pop: {key: -1}}. There is also the $pull modifier which can remove an element based on given criteria.
The $push modifier can be used in conjunction with $ne to check for the existence of an element and thereby avoid duplicates. There is, however, another modifier called $addToSet which will avoid duplicates without having to use $ne.

```
db.unicorns.update({name: "Roooooodles"},
              {"$set": {loves: ["apple", "sugar", "carrot"]}});

my $doc = $coll->update({"name" => "Roooooodles"},
          {'$set' => {"loves" => ["apple", "sugar", "carrot"]}})

Roooooodles m 575 99 1979-08-18T08:44:00 apple, sugar, carrot
```

```
db.unicorns.update({name: "Roooooodles"},
                                  {"$pull": {loves: "sugar"}});

my $doc = $coll->update({"name" => "Roooooodles"},
                               {'$pull' => {"loves" => "sugar"}})

Roooooodles   m  575 99  1979-08-18T08:44:00 apple, carrot
```

```
db.unicorns.update({name: "Roooooodles"},
                           {"$addToSet": {loves: "watermelon"}});

my $doc = $coll->update({"name" => "Roooooodles"},
                    {'$addToSet' => {"loves" => "watermelon"}});

Roooooodles m 575 99 1979-08-18T08:44:00 apple,carrot,watermelon
```

### 4.4 *Remove the existing key of a document*

```
db.unicorns.update({name: "Roooooodles"},
                   {"$unset": {loves: 1}});

my $doc = $coll->update({"name" => "Roooooodles"},
                        {'$unset' => {"loves" => 1}});

Roooooodles   m  575  99  1979-08-18T08:44:00
```

### 4.5 *Add more than one element to a key*

```
db.unicorns.update({name: "Roooooodles"},
   {"$addToSet": {loves: {"$each": ["carrot", "watermelon"]}}});

my $doc = $coll->update({"name" => "Roooooodles"},
       {'$addToSet' => {"loves" => {'$each' =>
                        ["carrot", "watermelon"]}}});

Roooooodles m 575  99  1979-08-18T08:44:00  apple,carrot,watermelon
```

## 5 Delete

### 5.1 *Remove a document*

```
db.unicorns.remove({name: 'Aurora'})

my $doc = $coll->remove({'name' => 'Aurora'});
```