

dealing w/ lots of data

- 20+ billion web pages x 20KB = 400+ terabytes (to just store it -- not to do sth.)
- One computer can read ~50MB/sec from disk
- so 3 months to read from web
- Solution?

Solution

- get 1000's of computers in cluster
- take 2.5 hrs to process the web



cluster of 1000 computers

4,500 Terabytes (4.5 Petabytes)
8 racks of 64 quad core servers



The only thing needed are network connections, a chilled water supply, and electricity

University of Mary Washington



[PROSPECTIVE STUDENTS](#)

[CURRENT STUDENTS](#)

[FACULTY & STAFF](#)

[ALUMNI](#)

[VISITORS](#)

[JAMES MONROE CENTER](#)

[VISITING THE
MWC CAMPUS...](#)

[COLLEGE CALENDAR](#)

[ACADEMIC CATALOGUE](#)

[NEWS](#)

[EVENTS](#)

[A TO Z INDEX](#)

[PEOPLE SEARCH](#)

[CAMPUS ADVISORIES](#)

[PROFILE OF MWC](#)

[VIRTUAL TOUR](#)

[EMPLOYMENT](#)

SEARCH UMW

go!

Our Most Visited Sites

go!

[Moving to University Status
Presidential Center](#)

[MWC Listed as one of Top 77 Best Values
in Education](#)

[Students Receive Excellence Awards at
Commencement Ceremony](#)

[Board of Visitors Approves 2004-05
Tuition and Fee Increase](#)

[Awards Presented to JMC Faculty Member
and Top Graduates](#)

["Sunday Tea at Belmont" to be Held
MCSE Certification w/Security Emphasis
to be Offered at JMC](#)

[ACADEMICS](#)

[ADMISSIONS](#)

[ATHLETICS](#)

[GALLERIES AND MUSEUMS](#)

[GRADUATE AND PROFESSIONAL](#)

[LIBRARIES AND RESEARCH](#)

[MAKE A GIFT](#)

[TECHNOLOGY](#)

top story

MWC GRADUATES ITS LARGEST CLASS EVER

*Childhood friends Brittany
Hammelman and Jamie
Oliver started and ended
their education the same
way--together. read more*

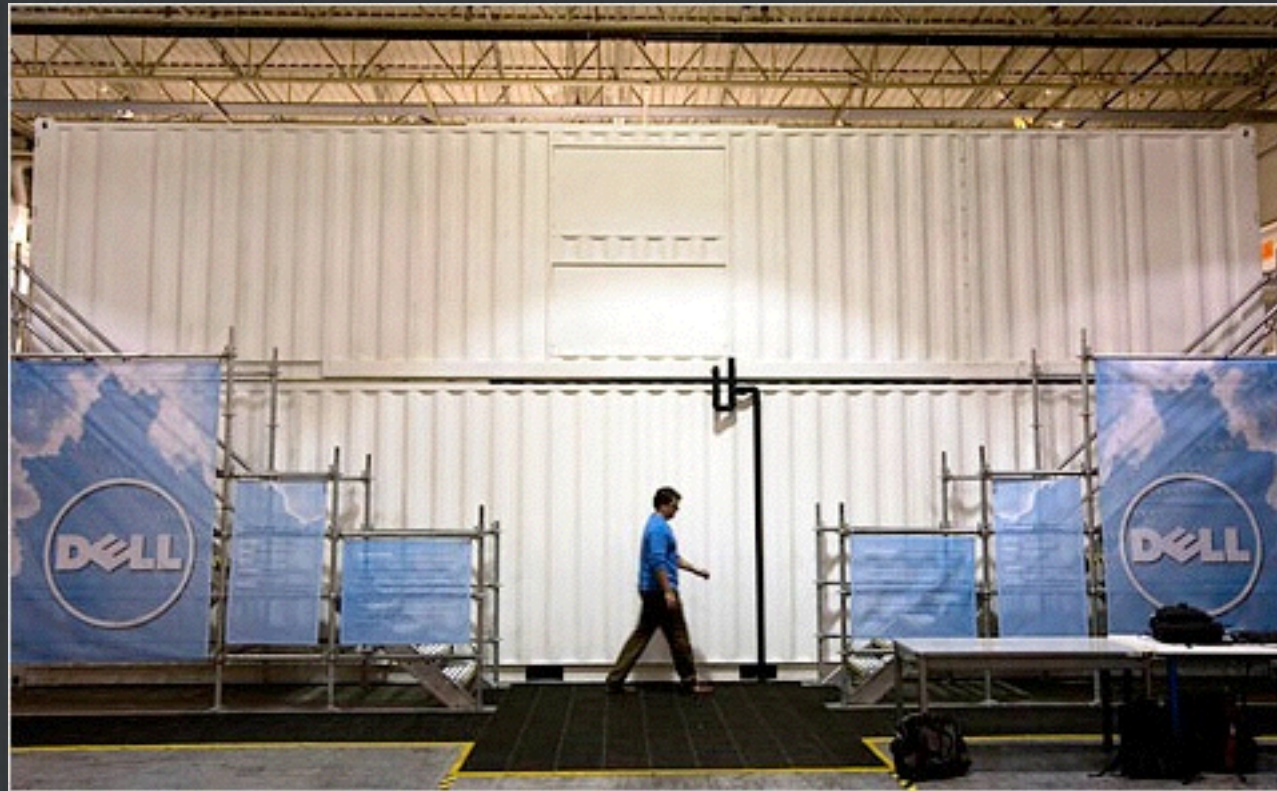
>>>

Mary Washington College
1301 College Avenue, Fredericksburg, VA 22401
Information: 540.654.1000 | [TTY Lines](#)
[Comments or Questions?](#)

[©2003 MWC](#) | [Disclaimer](#) | [Privacy](#) | [Text Site](#) | [James Monroe Center](#)



Nice Sun Server Box



You: a graduate of UMW



You: a graduate of UMW

**I hire you to write a program to handle lots
and lots of data.**

-
- You: a graduate of UMW**
 - I hire you to write a program to handle lots and lots of data.**
 - 3 months run time unacceptable.**
 -

-
- You: a graduate of UMW**
 - I hire you to write a program to handle lots and lots of data.**
 - 3 months run time unacceptable.**
 - I buy you 2 nice Sun 20ft server containers**

-
- You: a graduate of UMW**
 - I hire you to write a program to handle lots and lots of data.**
 - 3 months run time unacceptable.**
 - I buy you 2 nice Sun 20ft server containers**
 - How do you program it?**

Solutions

- Map Reduce**
(easy way to write distributed programs)
- Sharding**
- node.js**



MongoDB

The Definitive Guide

mongo

- **document oriented database**
 - **replaces concept of 'row' with 'document'**


```
{
  "_id" : ObjectId("4bddfd5bd7a20d38d26e34e7"),
  "rating" : "R",
  "release_year" : 2010,
  "rotten_id" : ObjectId("4d24029da9797eb6a7cfc dab"),
  "title" : "Black Swan",

  "actors" : [
    {
      "name" : "Barbara Hershey",
      "person_id" : ObjectId("4c453f78b6b4c16e051cf250")
    },
    {
      "name" : "Benjamin Millepied",
      "person_id" : ObjectId("4c45b16fb6b4c16e16c6f450")
    }
  ],
  "genres" : [
    "Suspense",
    "Thriller",
    "Psychological Thrillers"
  ]
}
```


mongoDB

Quickstart

Installation

- On Mac using Homebrew package manager:
 - brew install mongo
- On Linux, see mongodb.org.
 - should install the mongodb-10gen package
- On Windows, download from mongodb.org/downloads

nitrous.io

- parts install mongodb
- parts start mongodb

mongod - the server

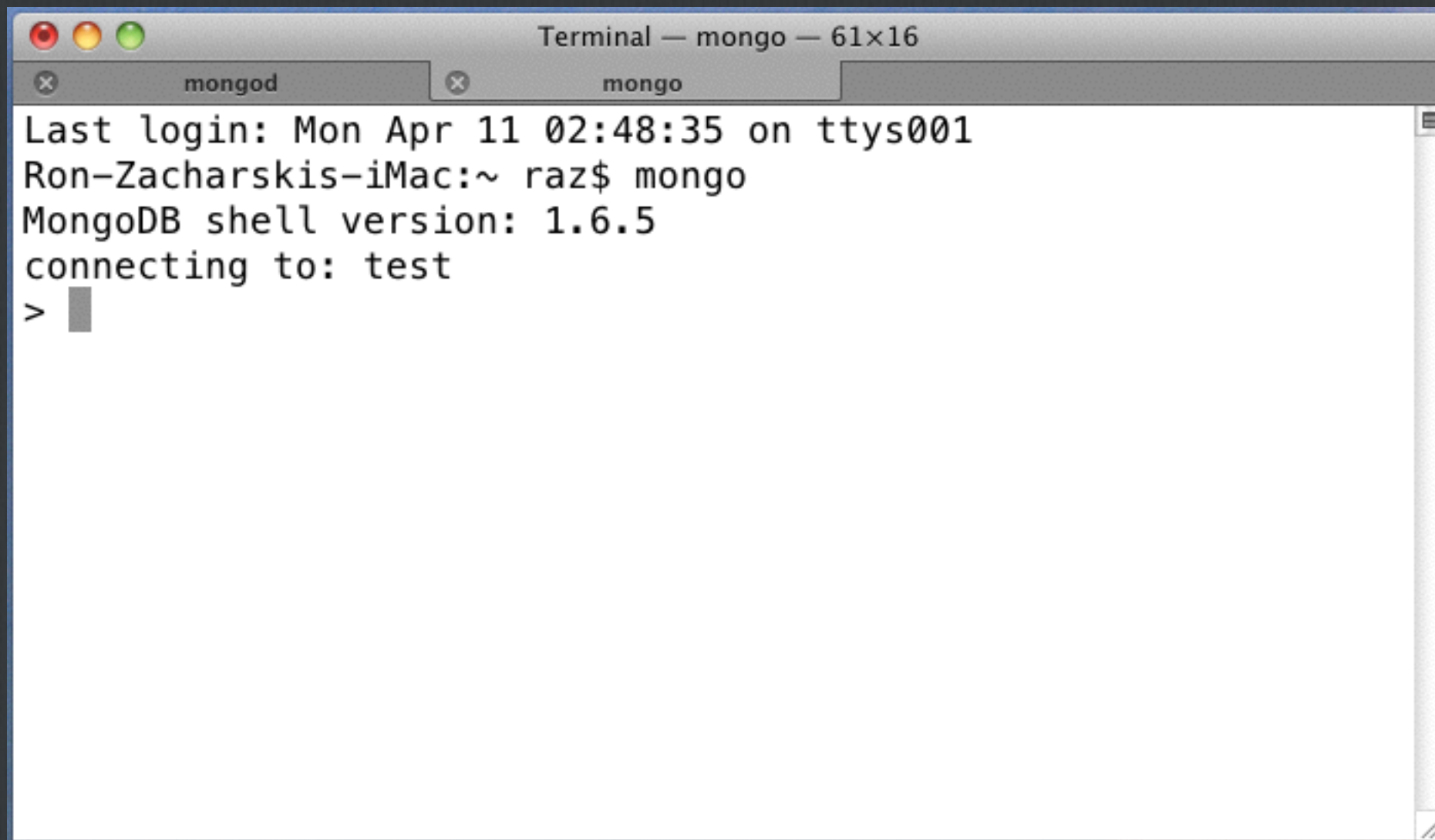
```
Terminal — mongod — 69x22
Last login: Sat Apr  9 22:18:48 on ttys000
Ron-Zacharskis-iMac:~ raz$ mongod
mongod --help for help and startup options
Mon Apr 11 02:42:36 MongoDB starting : pid=48116 port=27017 dbpath=/data/db/ 64-bit
Mon Apr 11 02:42:36 db version v1.6.5, pdfile version 4.5
Mon Apr 11 02:42:36 git version: 0eb017e9b2828155a67c5612183337b89e12e291
Mon Apr 11 02:42:36 sys info: Darwin erh2.10gen.cc 9.6.0 Darwin Kernel Version 9.6.0: Mon Nov 24 17:37:00 PST 2008; root:xnu-1228.9.59~1/RELEASE_I386 i386 BOOST_LIB_VERSION=1_40
Mon Apr 11 02:42:36 [initandlisten] waiting for connections on port 27017
Mon Apr 11 02:42:36 [websvr] web admin interface listening on port 28017
█
```


mongod - the server

```
Terminal — mongod — 69x22
Last login: Sat Apr  9 22:18:48 on ttys000
Ron-Zacharskis-iMac:~ raz$ mongod
mongod --help for help and startup options
Mon Apr 11 02:42:36 MongoDB starting : pid=48116 port=27017 dbpath=/data/db/ 64-bit
Mon Apr 11 02:42:36 db version v1.6.5, pdfile version 4.5
Mon Apr 11 02:42:36 git version: 0eb017e9b2828155a67c5612183337b89e12e291
Mon Apr 11 02:42:36 sys info: Darwin erh2.10gen.cc 9.6.0 Darwin Kernel Version 9.6.0: Mon Nov 24 17:37:00 PST 2008; root:xnu-1228.9.59~1/RELEASE_I386 i386 BOOST_LIB_VERSION=1_40
Mon Apr 11 02:42:36 [initandlisten] waiting for connections on port 27017
Mon Apr 11 02:42:36 [websvr] web admin interface listening on port 28017
```

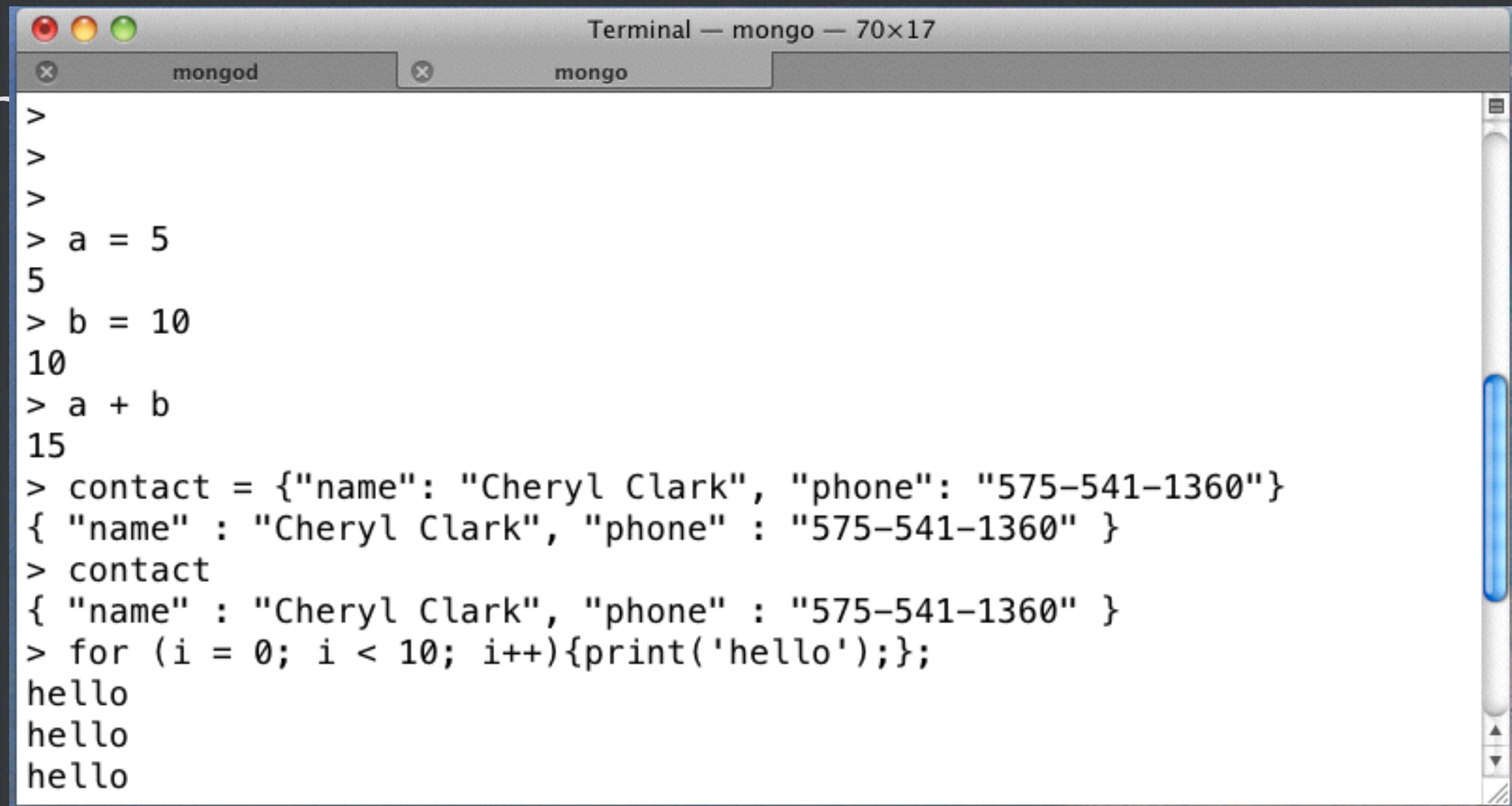

mongo

command line interpreter



```
Terminal — mongo — 61x16
mongod
mongo
Last login: Mon Apr 11 02:48:35 on ttys001
Ron-Zacharskis-iMac:~ raz$ mongo
MongoDB shell version: 1.6.5
connecting to: test
> █
```

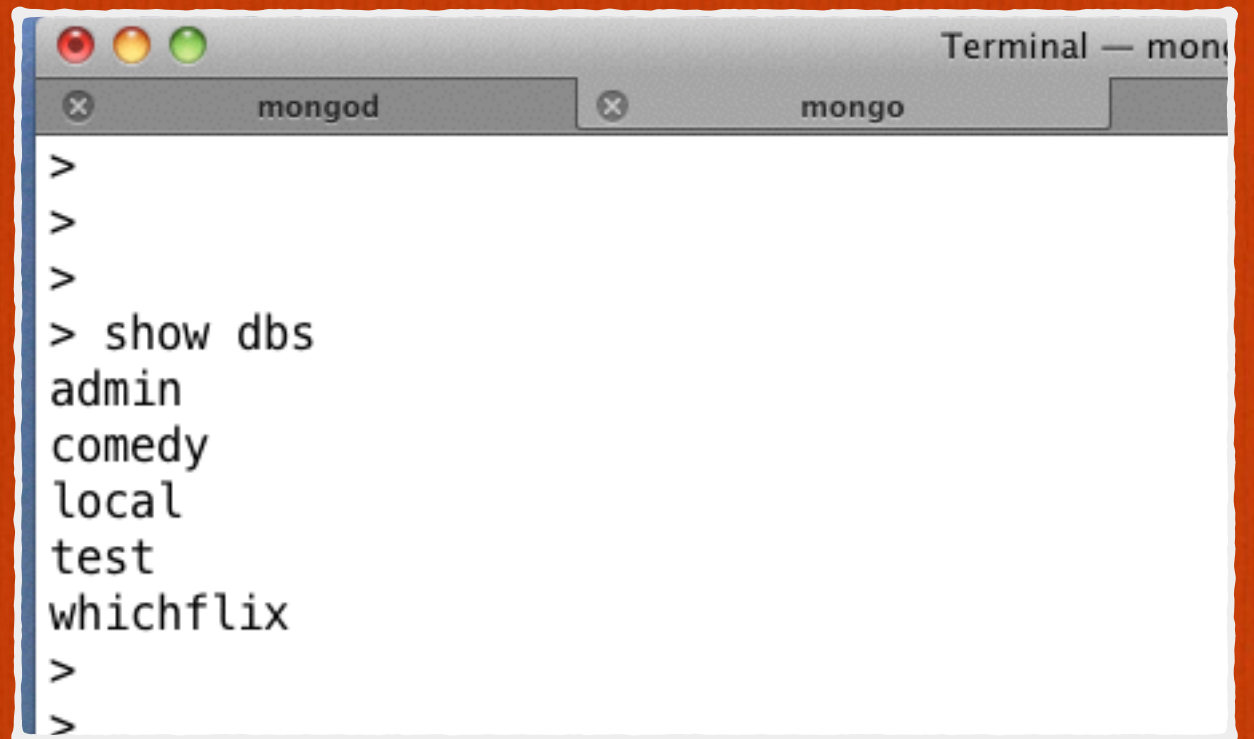

JavaScript Shell



```
Terminal — mongo — 70x17
mongod  mongo
>
>
>
> a = 5
5
> b = 10
10
> a + b
15
> contact = {"name": "Cheryl Clark", "phone": "575-541-1360"}
{ "name" : "Cheryl Clark", "phone" : "575-541-1360" }
> contact
{ "name" : "Cheryl Clark", "phone" : "575-541-1360" }
> for (i = 0; i < 10; i++){print('hello');};
hello
hello
hello
```

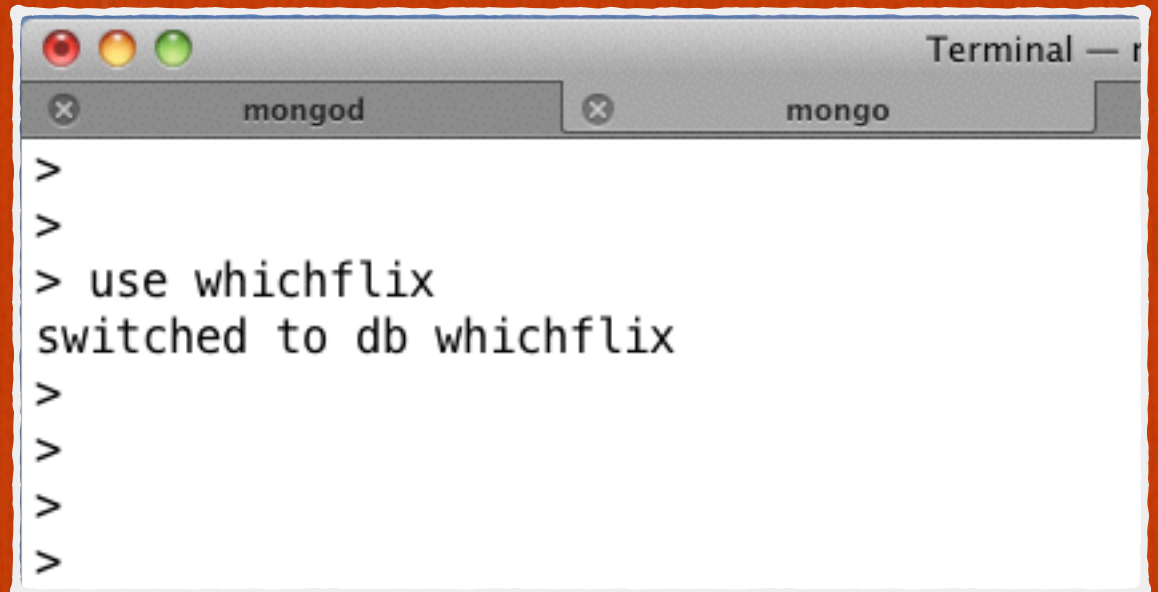

**commands similar to
MySQL**

show databases
-> show dbs

A screenshot of a macOS Terminal window titled "Terminal - mongod". The window has two tabs: "mongod" and "mongo". The "mongo" tab is active. The terminal shows a series of prompt characters ">" followed by the command "show dbs" and its output: "admin", "comedy", "local", "test", and "whichflix".

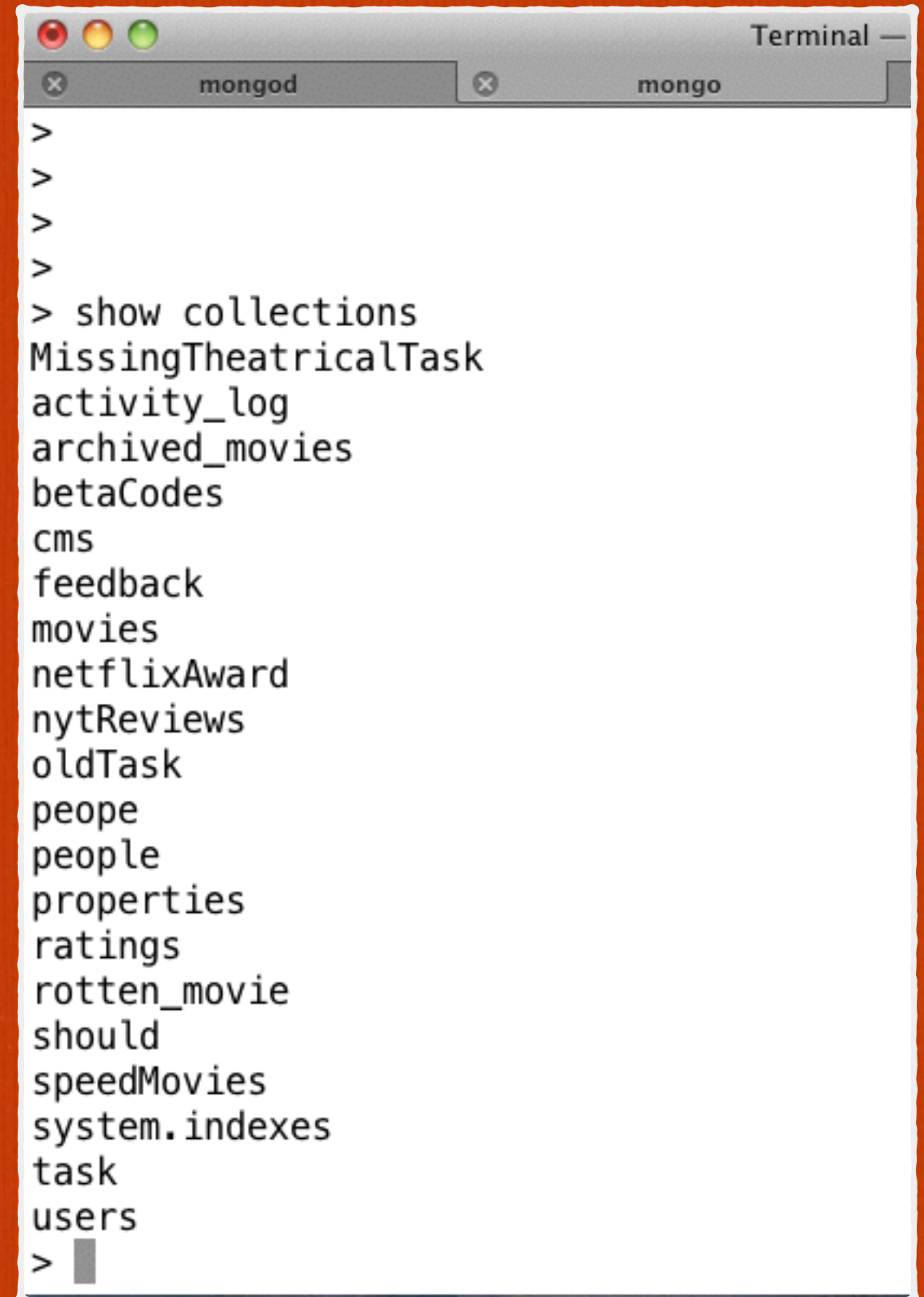
```
Terminal - mongod
mongod
mongo
>
>
>
> show dbs
admin
comedy
local
test
whichflix
>
>
```


use xxx

A screenshot of a macOS Terminal window with two tabs: 'mongod' and 'mongo'. The 'mongo' tab is active. The terminal shows a series of prompt characters '>' and the command 'use whichflix', followed by the output 'switched to db whichflix'.

```
Terminal - r
mongod mongo
>
>
> use whichflix
switched to db whichflix
>
>
>
>
```


show tables ->
show
collections



```
Terminal —
mongod x mongo x
>
>
>
>
> show collections
MissingTheatricalTask
activity_log
archived_movies
betaCodes
cms
feedback
movies
netflixAward
nytReviews
oldTask
people
people
properties
ratings
rotten_movie
should
speedMovies
system.indexes
task
users
> █
```


Queries

name of collection (i.e., table)



```
db.movies.findOne({})
```

What to match -- like where ...

```
Terminal — mongo — 84x26
mongod
mongo
>
>
>
> db.movies.findOne({})
{
  "_id" : ObjectId("4b9081f449ae805e00627343"),
  "actors" : [
    {
      "name" : "Helena Bonham Carter",
      "person_id" : ObjectId("4c453f57b6b4c16efb0bf250")
    },
    {
      "name" : "Mickey Rourke",
      "person_id" : ObjectId("4c453f57b6b4c16efc0bf250")
    }
  ],
  "amazon" : {
    "updated" : "Thu Aug 26 2010 11:47:18 GMT-0400 (EDT)"
  },
  "directors" : [
    {
      "name" : "Liliana Cavani",
      "person_id" : ObjectId("4c453f57b6b4c16efd0bf250")
    }
  ],
  "genres" : [
```



```
Terminal — mongo — 84x26
mongod
mongo
>
>
>
>
> db.movies.findOne({"title": "Black Swan"})
{
  "_id" : ObjectId("4bddfd5bd7a20d38d26e34e7"),
  "actors" : [
    {
      "name" : "Barbara Hershey",
      "person_id" : ObjectId("4c453f78b6b4c16e051cf250")
    },
    {
      "name" : "Benjamin Millepied",
      "person_id" : ObjectId("4c45b16fb6b4c16e16c6f450")
    },
    {
      "name" : "Janet Montgomery",
      "person_id" : ObjectId("4c45a31eb6b4c16e6c93f450")
    },
    {
      "name" : "Kristina Anapau",
      "person_id" : ObjectId("4c454272b6b4c16e878cf250")
    },
    {
      "name" : "Ksenia Solo",
```

```
Terminal — mongo — 84x26
mongod  mongo
>
> db.movies.findOne({"actors.name": "Natalie Portman"})
{
  "_id" : ObjectId("4b9081f649ae805e30677343"),
  "actors" : [
    {
      "name" : "Ahmed Best",
      "person_id" : ObjectId("4c453f89b6b4c16eca24f250")
    },
    {
      "name" : "Christopher Lee",
      "person_id" : ObjectId("4c453f73b6b4c16e051bf250")
    },
    {
      "name" : "Ewan McGregor",
      "person_id" : ObjectId("4c453f89b6b4c16ecb24f250")
    },
    {
      "name" : "Frank Oz",
      "person_id" : ObjectId("4c453f55b6b4c16e150bf250")
    }
  ],
  "title" : "Star Wars: Episode II: Attack of the Clones",
  "tmdb" : {
    "updated" : "Thu Sep 09 2010 08:16:35 GMT-0400 (EDT)"
  }
}
> 
```


Comparison Operators

\$lt \$lte \$gt \$gte \$ne

- `db.users.find({"age": {"$gte": 18, "$lte": 30}})`

- `db.users.find("name": {"$ne": "Ann"})`

or

```
db.movies.find({"actors.name": {"$in": ["Angelina Jolie", "Natalie Portman"]})
```


Regular Expressions

Joe, joe, Joey, joey

```
db.users.find({"name": /joey?/i})
```


query demo

save & insert: the insert equivalents

Collection

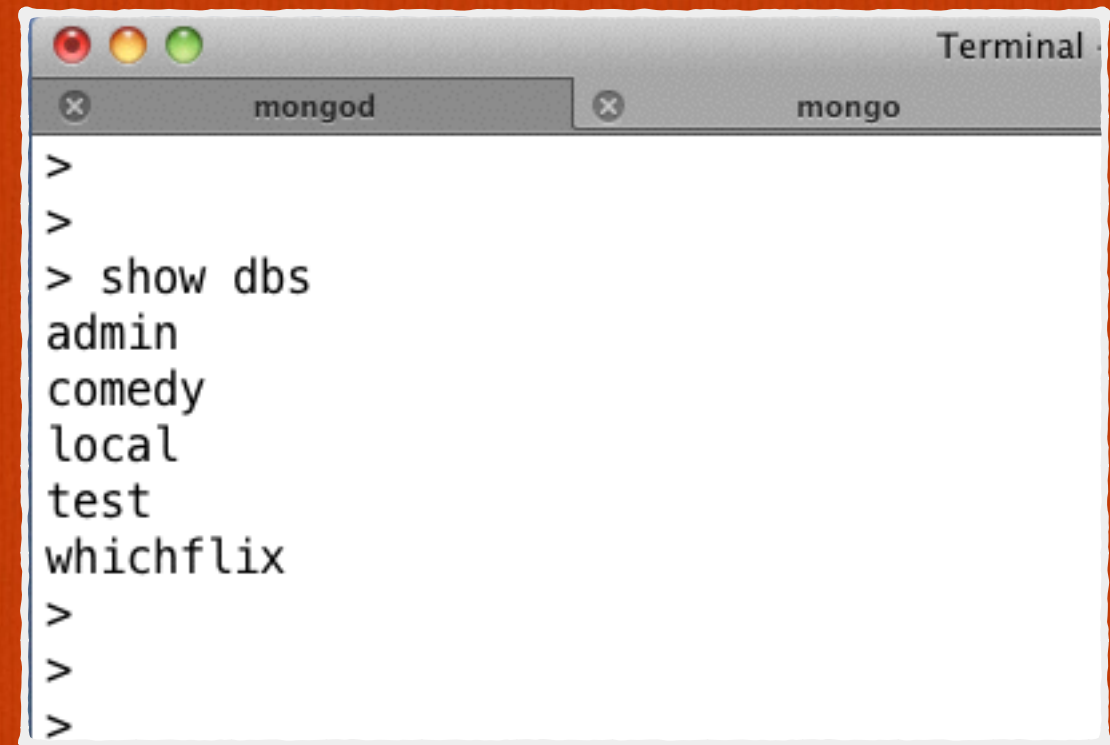


```
db.contacts.save({"name": "Cheryl Clark",  
                 "cell": "575-541-1360"})
```



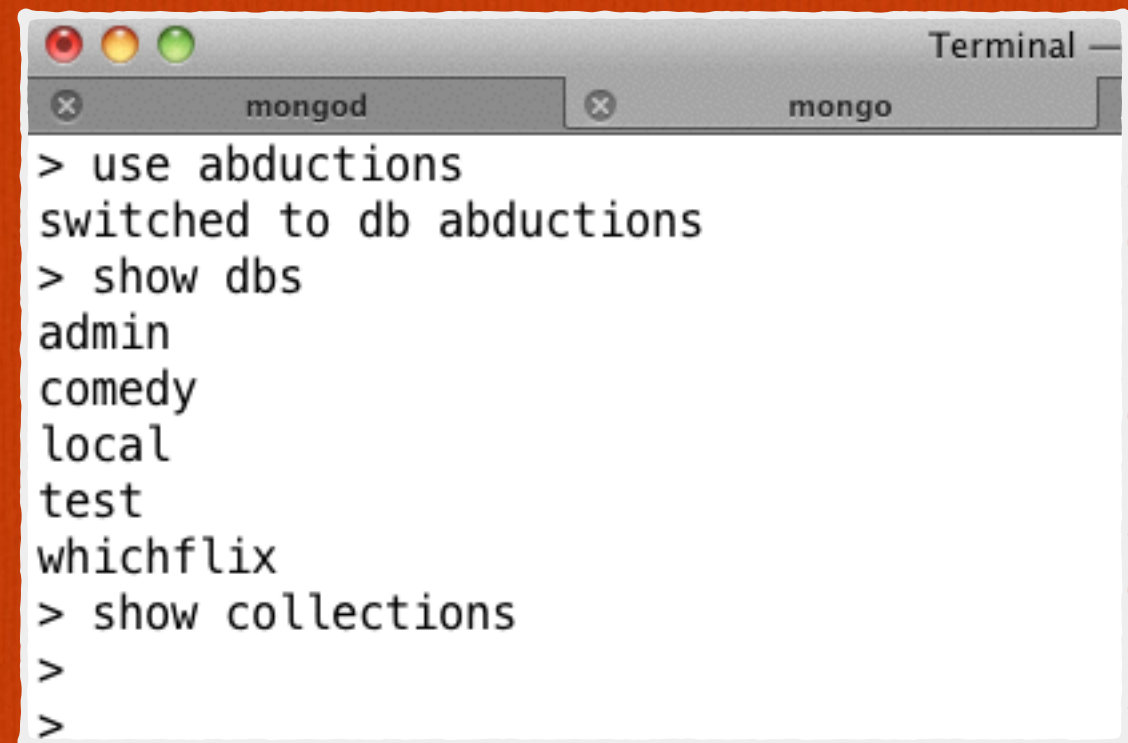
What to save

**Databases and
collections are
created
automagically**

A screenshot of a macOS Terminal window. The window title is "Terminal" and it has two tabs: "mongod" and "mongo". The "mongo" tab is active. The terminal shows a prompt ">" followed by the command "show dbs". The output lists five databases: "admin", "comedy", "local", "test", and "whichflix". The prompt ">" appears again at the end of the output.

```
>  
>  
> show dbs  
admin  
comedy  
local  
test  
whichflix  
>  
>  
>
```


**Databases and
collections are
created
automagically**

A screenshot of a macOS Terminal window. The window title is "Terminal" and it has two tabs: "mongod" and "mongo". The "mongo" tab is active. The terminal shows the following commands and output:

```
> use abductions
switched to db abductions
> show dbs
admin
comedy
local
test
whichflix
> show collections
>
>
```



```
Terminal — mongo — 84x26
mongod  mongo
>
>
> a = {"name": "Dana Scully", "implant": "metal cylinder"}
{ "name" : "Dana Scully", "implant" : "metal cylinder" }
> db.implants.save(a)
> show collections
implants
system.indexes
> show dbs
abductions
admin
comedy
local
test
whichflix
>
>
```

**Databases and collections are created
automagically**

How to save and restore MongoDB databases

mongodump / mongorestore

for ex.,

mongodump -d whichflix

saves to a folder called dump

mongoDB & PHP

the equivalent of `mysqli_connect`

need to install mongoDB library

pip install pymongo

then ...

```
>>> from pymongo import MongoClient

>>> client = MongoClient()

>>> db = client.abductions

>>> db.implants.find_one()
{u'implant': u'metal cylinder', u'_id':
ObjectId('53409e18fed08dc8b0001034'), u'name': u'Dana Scully'}

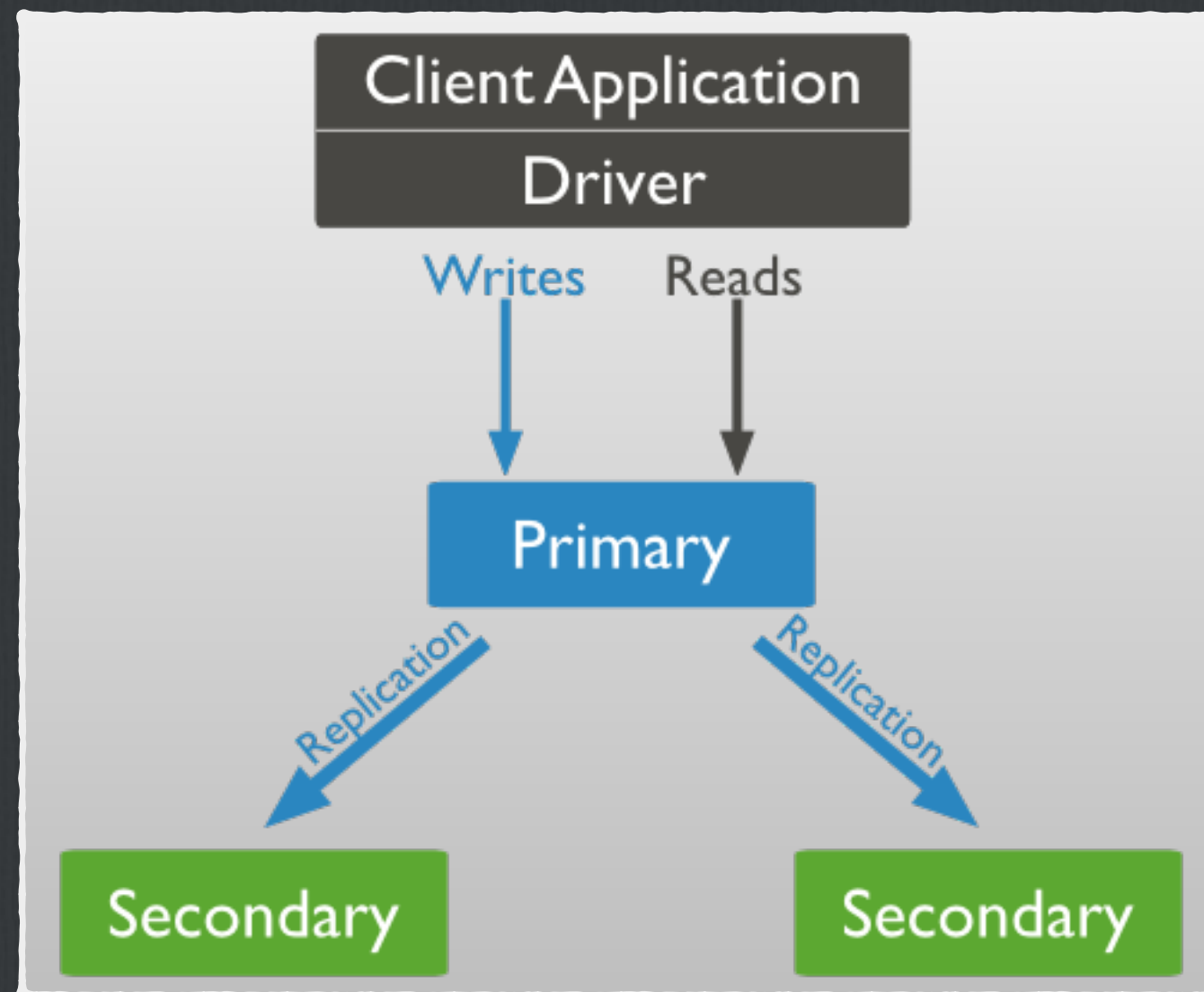
>>>
```


more later

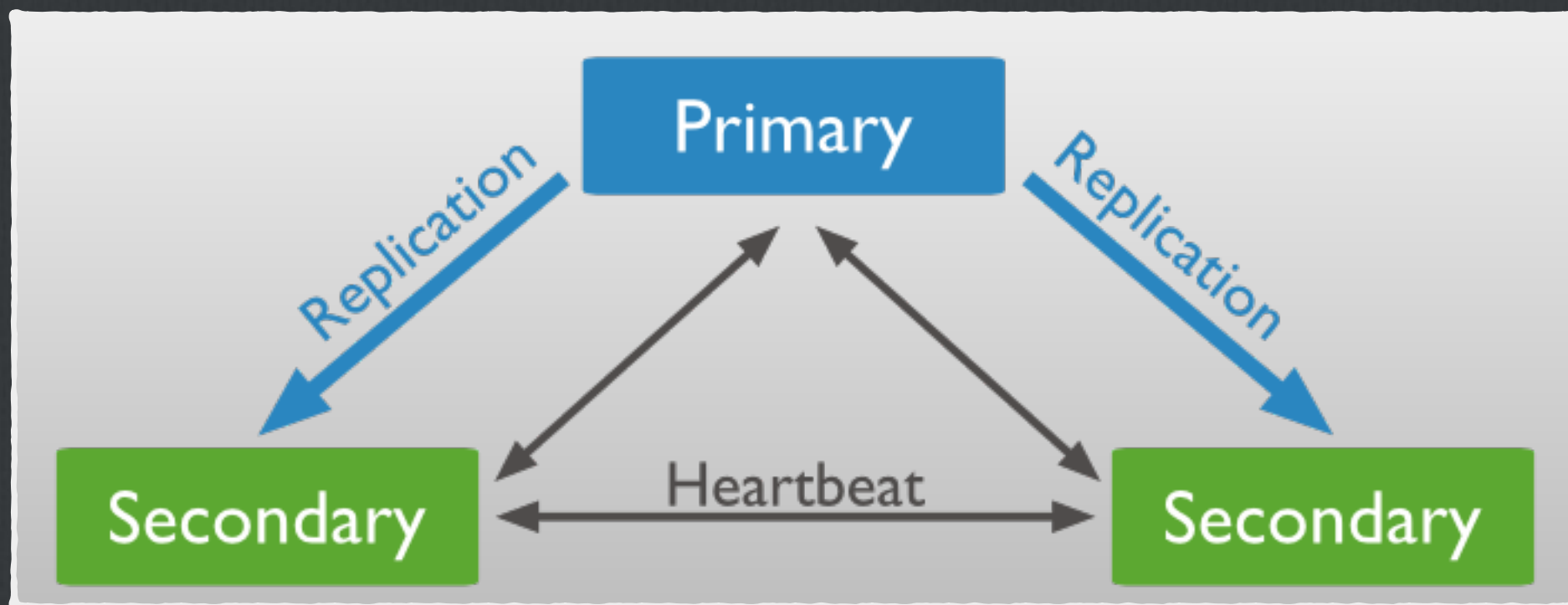
replication

(historically master/slave)

**writes are only to the
primary**



secondaries



master slave

- master performs reads and writes
 - `mongod --master` (default port 27017)
- slaves perform reads not writes
 - `mongod --slave --dbpath /data/db/slave --port 10001 --source localhost:27017`