Database Management Systems

- MongoDB

Doug Shook

Document Stores

- What is a document?

- “The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB”, David Hows; Peter Membrey; Eelco Plugge; Tim Hawkins, 2015

Documents

- Made up of key-value pairs
  - Each has a type
  - Order matters (kind of)
- Type sensitive and case sensitive
- No duplicate keys allowed

Documents

```
{
    "firstname": "Peter",
    "lastname": "Membrey",
    "phone_numbers": [
        "+852 1234 5678",
        "+44 1234 565 555"
    ]
}
```

Collections

- Collections are groups of documents
  - Is there a schema?
- Sub collections are also allowed

Indexing

- Automatically created on ID
- Can create your own
  - Embedded documents
  - Composite Indexes

Design

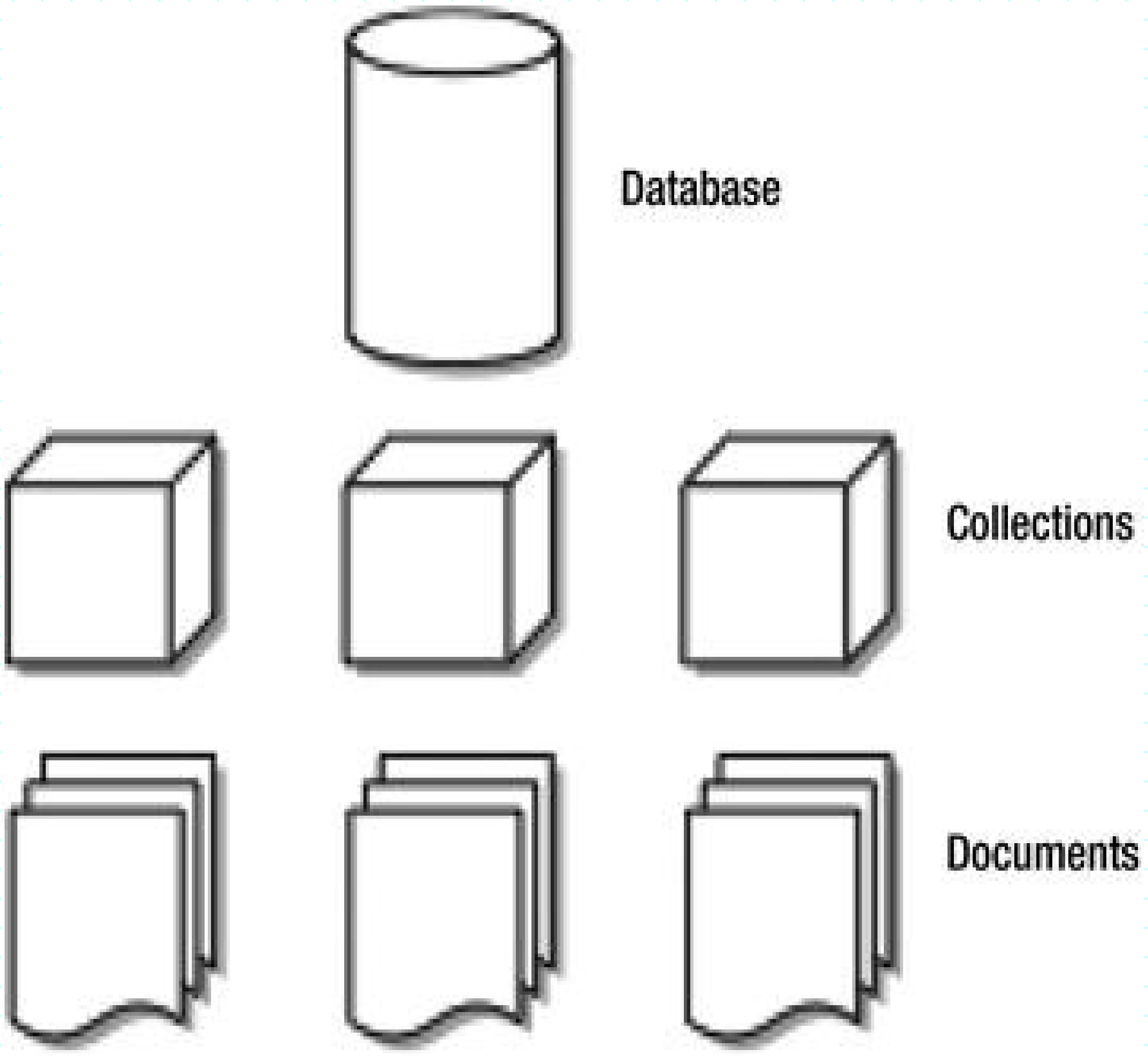
- The database is so flexible
  - How do we choose a good design?

```
  {
     "type": "Book",
     "Title": "Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB 3rd ed., The",
     "ISBN": "978-1-4842-1183-0",
     "Publisher": "Apress",
     "Author": [
         "Hows, David"
         "Plugge, Eelco",
         "Membrey, Peter",
         "Hawkins, Tim     ]
 }
```

Design

```
{
    "Type": "CD",
    "Artist": "Nirvana",
    "Title": "Nevermind",
    "Genre": "Grunge",
    "Releasedate": "1991.09.24",
    "Tracklist": [
        {
        "Track": "1",
        "Title": "Smells Like Teen Spirit",
        "Length": "5:02"
        },
        {
        "Track": "2",
        "Title": "In Bloom",
        "Length": "4:15"
        }
    ]
}
```

Database Structure



Data Types

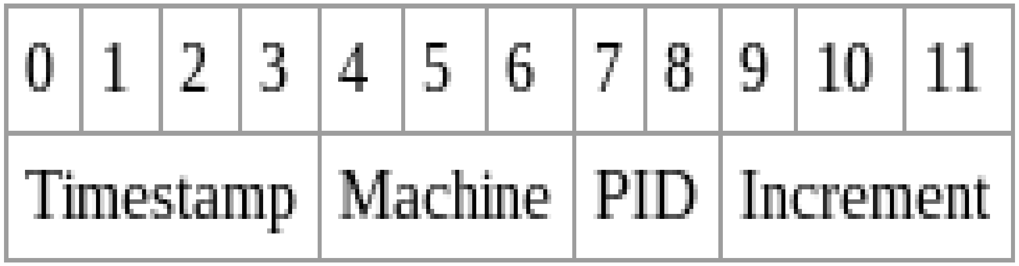
- Null
- Boolean
- Integer (careful!)
- Floating point
- String
- Date
- Regular Expression
- Javascript code
- Array
- Embedded Document

Embedding vs. Referencing

```
{
    "Type": "CD",
    "Artist": "Nirvana",
    "Title": "Nevermind",
    "Genre": "Grunge",
    "Releasedate": "1991.09.24",
    "Tracklist": [
        {
        "Track" : "1",
        "Title" : "Smells Like Teen Spirit",
        "Length" : "5:02"
        },
        {
        "Track" : "2",
        "Title" : "In Bloom",
        "Length" : "4:15"
        }
    ]
}
```

ObjectIds

- Special type that uniquely identifies each object within a collection



0	1	2	3	4	5	6	7	8	9	10	11
Timestamp				Machine			PID		Increment		

Navigation

```
use library

show dbs

show collections
```

Insertion

```
document = ({"Type": "Book", "Title" : "Definitive Guide to MongoDB 3rd ed., The", "ISBN" : "978-1-4842-1183-0", "Publisher" : "Apress", "Author" : ["Hows, David", "Plugge, Eelco", "Membrey, Peter", "Hawkins, Tim"] } )

db.media.insertOne(document)

db.media.insertOne( { "Type" : "CD", "Artist" : "Nirvana", "Title" : "Nevermind" })
```

Querying

```
db.media.find()

db.media.find ( { Artist : "Nirvana" } )

db.media.find ( {Artist : "Nirvana"}, {Title: 1} )

db.media.find( { "Author" : "Membrey, Peter" } )

db.media.find().sort( { Title: 1 })
```

Aggregates

```
db.media.count()

db.media.find( { Publisher : "Apress", Type: "Book" }
 ).count()

db.media.group (
{
    key: {Title : true},
    initial: {Total : 0},
    reduce : function (items,prev)
    {
        prev.Total += 1
    }
}
)
```

Conditionals

```
dvd = ( { "Type" : "DVD", "Title" : "Matrix, The", "Released" : 1999, "Cast" : ["Keanu Reeves","Carrie-Anne Moss","Laurence Fishburne","Hugo Weaving","Gloria Foster","Joe Pantoliano"] } )
db.media.insertOne(dvd)
dvd = ( { "Type" : "DVD", Title : "Blade Runner", Released : 1982 } )
db.media.insertOne(dvd)
dvd = ( { "Type" : "DVD", Title : "Toy Story 3", Released : 2010 } )
db.media.insertOne(dvd)
```

Conditionals

```
db.media.find ( { Released : {$gt : 2000} },
{ "Cast" : 0 } )

db.media.find ( { Released : {$gte : 1999 } },
{ "Cast" : 0 } )

db.media.find ( { Released : {$lt : 1999 } },
{ "Cast" : 0 } )

db.media.find( {Released : {$in :
[1999,2008,2009] } }, { "Cast" : 0 } )

db.media.find({ $or : [ { "Title" : "Toy Story 3" },
{ "ISBN" :"978-1-4842-1183-0" } ] } )
```

Updates

```
db.media.updateOne( { "Title" : "Matrix, The"}, {"Type" : "DVD", "Title" : "Matrix, The", "Released" : 1999, "Genre" : "Action"}, { upsert: true} )
```

```
db.media.updateMany( { "Title" : "Matrix, The"}, {$set: {"Type" : "DVD", "Title" : "Matrix, The", "Released" : 1999, "Genre" : "Action"} }, {upsert: true} )
```

```
db.media.update ( { "Title" : "Matrix, The" }, {$set : { Genre :"Sci-Fi" } } )
```

```
db.media.updateOne ( {"Title": "Matrix, The"}, {$unset : { "Genre" : 1 } } )
```

Deletion

```
db.newname.deleteOne( { "Title" : "Different Title" } )

db.newname.deleteMany({})

db.newname.drop()

db.dropDatabase()
```

Exercises

- Design a schema to be used to hold users and their reviews of various books. Create a few fake users and reviews and put them in the DB.

- Using your schema can you find:
  - The average review of a book?
  - The average review of a user?
  - The number of books in the system?