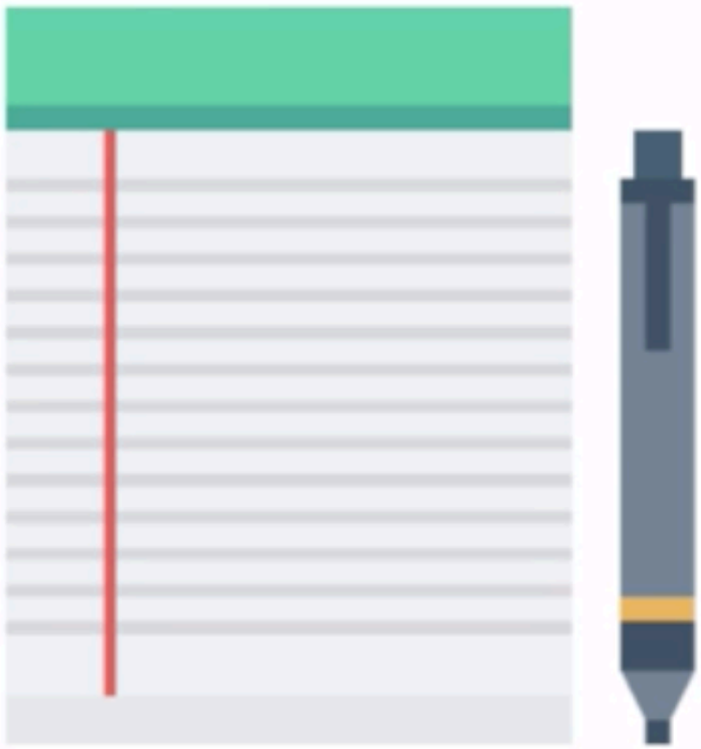
MongoDB



- What is MongoDB
- Installation
- Basic shell commands

Key Points

Documents

```
{
  "_id" : ObjectId("5501d48fb20c30e0548f"),
  "title" : "Article Title 1",
  "body" : "this is the article body"
}
```

BSON

NoSQL

non-Relational

SQL

Joins

NoSQL

ref = ‘User’

1

Main Advantages

Why Use MongoDB Over Traditional SQL?

Advantages

- Easy Schema Iteration
- Scalability & Performance
- Object-Oriented

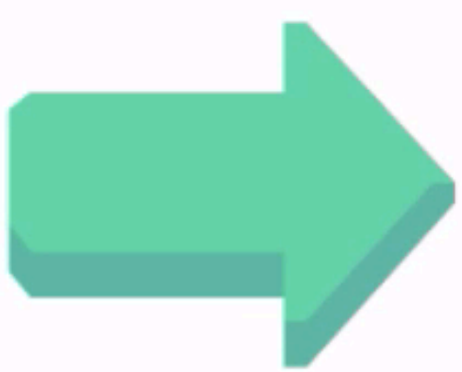
Agile Development

Accomodates large volumes of rapidly changing structured, semi-structured and unstructured data.

Add to Schema

User Schema

```
{
 first: “larry”,
 last: “david”
}
```



User Schema Updated

```
{
  first: “larry”,
  last: “david”,
  number: 555-9090
}
```

LEARN MORE

http://www.mongodb.com/nosql-explained

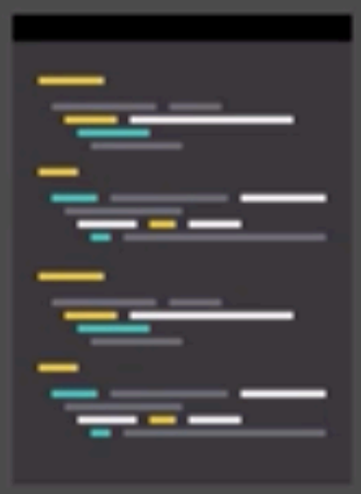
Document vs Collection

“

Document

A record in a MongoDB collection and the basic unit of data in MongoDB. Documents look like JSON objects but exist as BSON.

JSON = JavaScript Object Notation



Document Example

```
{
        "_id" : ObjectId("5591d39fb2933c30e0486f0b"),
        "title" : "Article two",
        "category" : "Education",
        "body" : "this is the body",
        "date" : ISODate("2015-06-29T23:24:15.212Z")
}
```

“

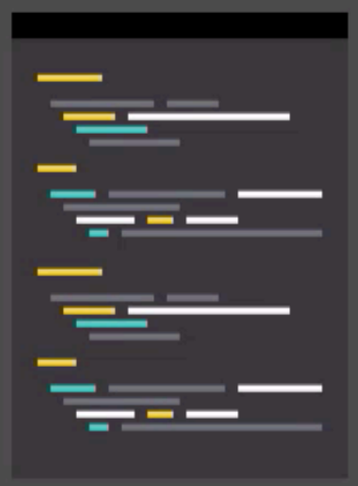
Collection

A grouping of MongoDB documents. Typically, all documents in a collection have a similar or related purpose.

Collection Example

```
{
        "_id" : ObjectId("5591d39fb2933c30e0486f0b"),
        "title" : "Article two",
        "category" : "Education",
        "body" : "this is the body",
        "date" : ISODate("2015-06-29T23:24:15.212Z")
}
{
        "_id" : ObjectId("5591d3ceb2933c30e0486f0c"),
        "title" : "Article three",
        "category" : "Health Care",
        "body" : "this is the body",
        "date" : ISODate("2015-06-29T23:25:02.226Z")
}
```

Schema = Collection



MongoDB Commands

Add, Modify, Find

Mongo Shell

DATA TYPES

- db.createCollection(“cars”)
- $upsert({ cars: ‘honda’ })
- Types of Data

DATA TYPES

STRING

name: String

```
{
  name: “John”
}
```

NUMBER

likes: Number

```
{
  likes: 5
}
```

DATE

timeStamp: Date

```
{
  timeStamp: ISODate(“...”)
}
```

ARRAY

tags: Array

OR

tags: []

```
{
  tags: [“tag1”, “tag2”]
}
```

BOOLEAN

published: Boolean

```
{
  published: true
}
```

ObjectId

_creator: Schema.ObjectId

```
{
  _creator: “41239878”
}
```

OTHER TYPES

Buffer

- Used for Video, Images, and Audio

Mixed

- Combines different types.

QUERY

find()

```
db.student.find({})
db.student.find({'name': 'Rachel'})
db.student.find({units: {$gt: 6}})
```

sort(), limit()

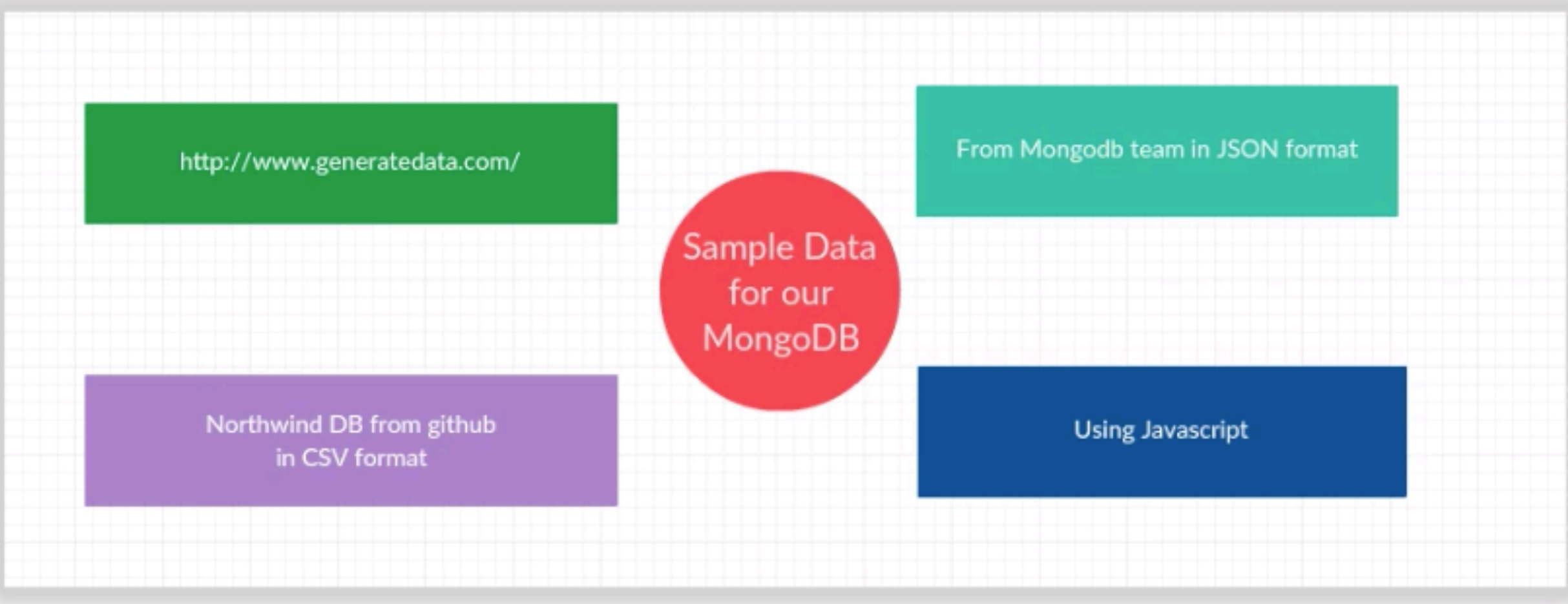
```
db.student.find({}).sort({name: 1}).limit(2)
```

```
db.student.find({})
db.student.find({'name': 'Rachel'})
db.student.find({units: {$gt: 6}})
db.student.find({units: {$lt: 7}})
db.student.find({classes: {$in: ['history']}})
```

```
db.student.find({classes: {$in: ['history']}}).sort({units: -1})    // ascending
db.student.find({}).sort({name: 1})    // descending

db.student.find({}).sort({name: 1}).limit(2)
```

https://docs.mongodb.com/manual/tutorial/query-documents/

Load Test Data in Mongodb



Mongodb Test Datasets

https://github.com/ozlerhakan/mongodb-json-files

https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json

http://media.mongodb.org/zips.json

https://github.com/cjlee/northwind

https://github.com/tmcnab/northwind-mongo

Northwind Database

Products

	Column Name	Condensed...
PK	ProductID	int
	ProductName	nvarchar(40)
	SupplierID	int
	CategoryID	int
	QuantityPerUnit	nvarchar(20)
	UnitPrice	money
	UnitsInStock	smallint
	UnitsOnOrder	smallint
	ReorderLevel	smallint
	Discontinued	bit

Order Details

	Column ...	Conde...
PK	OrderID	int
PK	ProductID	int
	UnitPrice	money
	Quantity	smallint
	Discount	real

EmployeeTerritories

	Column Name	Condensed T...
PK	EmployeeID	int
PK	TerritoryID	nvarchar(20)

Orders

	Column Name	Condensed...
PK	OrderID	int
	CustomerID	nchar(5)
	EmployeeID	int
	OrderDate	datetime
	RequiredDate	datetime
	ShippedDate	datetime
	ShipVia	int
	Freight	money
	ShipName	nvarchar(40)
	ShipAddress	nvarchar(60)
	ShipCity	nvarchar(15)
	ShipRegion	nvarchar(15)
	ShipPostalCode	nvarchar(10)
	ShipCountry	nvarchar(15)

Customers

	Column Name	Condensed...
PK	CustomerID	nchar(5)
	CompanyName	nvarchar(40)
	ContactName	nvarchar(30)
	ContactTitle	nvarchar(30)
	Address	nvarchar(60)
	City	nvarchar(15)
	Region	nvarchar(15)
	PostalCode	nvarchar(10)
	Country	nvarchar(15)
	Phone	nvarchar(24)
	Fax	nvarchar(24)

Categories

	Column Name	Condensed...
PK	CategoryID	int
	CategoryName	nvarchar(15)
	Description	ntext
	Picture	image

Employees

	Column Name	Condensed ...
PK	EmployeeID	int
	LastName	nvarchar(20)
	FirstName	nvarchar(10)
	Title	nvarchar(30)
	TitleOfCourtesy	nvarchar(25)
	BirthDate	datetime
	HireDate	datetime
	Address	nvarchar(60)
	City	nvarchar(15)
	Region	nvarchar(15)
	PostalCode	nvarchar(10)
	Country	nvarchar(15)
	HomePhone	nvarchar(24)
	Extension	nvarchar(4)
	Photo	image
	Notes	ntext
	ReportsTo	int
	PhotoPath	nvarchar(255)

CustomerCustomerDemo

	Column Name	Condensed T...
PK	CustomerID	nchar(5)
PK	Customer...	nchar(10)

Suppliers

	Column Name	Condensed...
PK	SupplierID	int
	CompanyName	nvarchar(40)
	ContactName	nvarchar(30)
	ContactTitle	nvarchar(30)
	Address	nvarchar(60)
	City	nvarchar(15)
	Region	nvarchar(15)
	PostalCode	nvarchar(10)
	Country	nvarchar(15)
	Phone	nvarchar(24)
	Fax	nvarchar(24)
	HomePage	ntext

Territories

	Column Name	Condensed...
PK	TerritoryID	nvarchar(20)
	TerritoryDescription	nchar(50)
	RegionID	int

Region

	Column Name	Conden...
PK	RegionID	int
	RegionDescription	nchar(50)

Shippers

	Column Name	Condensed...
PK	ShipperID	int
	CompanyName	nvarchar(40)
	Phone	nvarchar(24)

CustomerDemographics

	Column Name	Conden...
PK	CustomerTypeID	nchar(10)
	CustomerDesc	ntext

mongoimport command line tool

https://docs.mongodb.com/manual/reference/program/mongoimport/

Import JSON File

```
mongoimport --db users --collection contacts --file contacts.json
```

Import CSV File

```
mongoimport --db users --collection contacts --type csv --headerline --
file /opt/backups/contacts.csv
```

Generate Dummy Data

https://mockaroo.com/

http://generatedata.com/

Assignment

- Use the testdatagenerator available with MongoBooster to generate and Load sample data(for eg) Product details) to MongoDB. Refer here for More info https://mongobooster.com/features#TestDataGenerator

- Use https://www.mockaroo.com/ to generate user profile data a JSON output and load the data to MongoDB using MongoBooster and then using Mongoimport utility

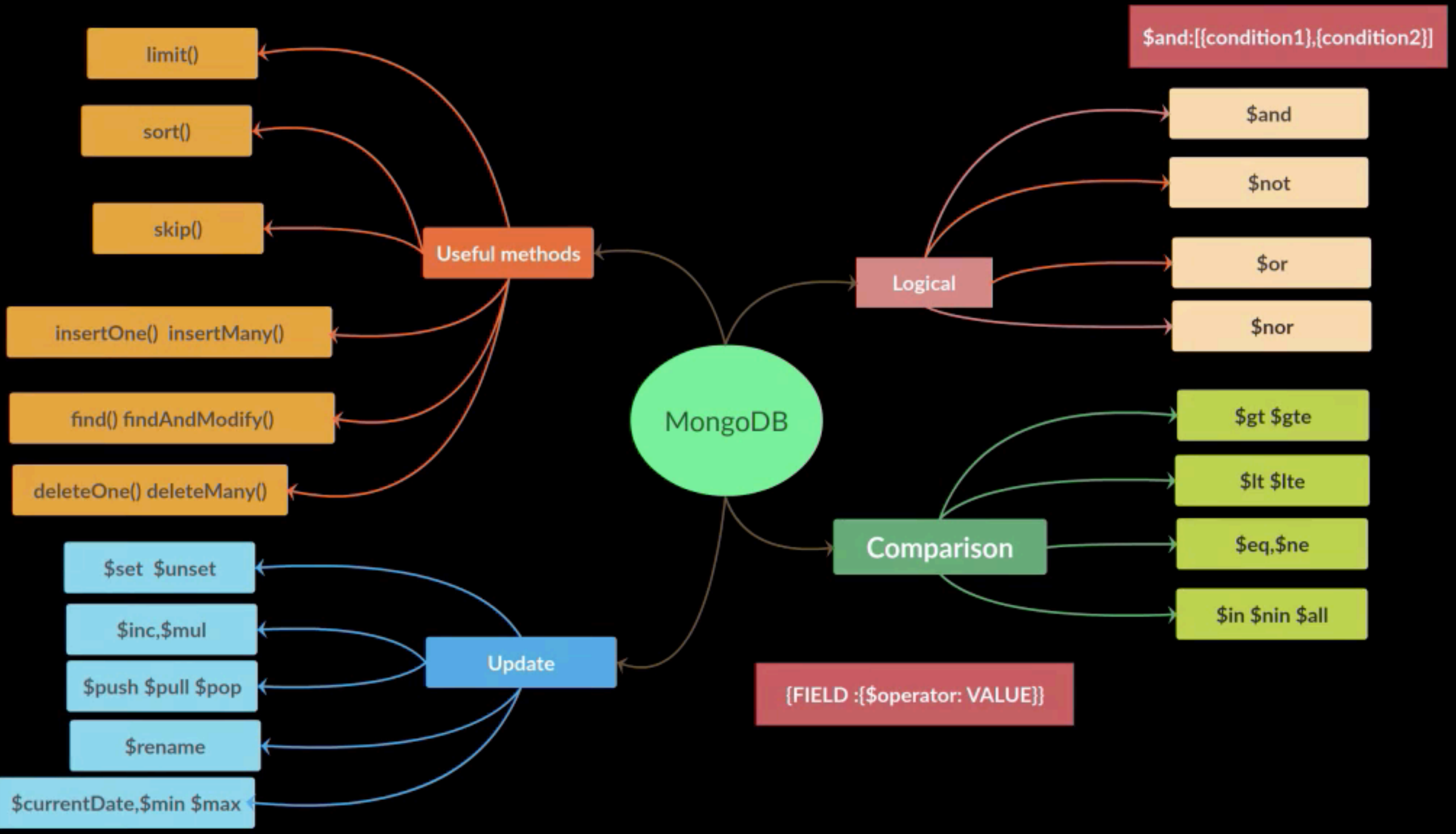
- Repeat the same Exercise using CSV format output from mockaroo.

- Load demo data provided by mongodb team using MongoBooster/MongoImport Tool. Demo Data URL: https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json

- Load again an useful demo dataset provided by mongodb team. The information is available here :https://docs.mongodb.com/manual/tutorial/aggregation-zip-code-data-set/

- The collection is available : http://media.mongodb.org/zips.json?_ga=2.261334925.1881078164.1503396731-778080937.1486117063

Queries


limit()
sort()
skip()
Useful methods
insertOne() insertMany()
find() findAndModify()
deleteOne() deleteMany()
$set $unset
$inc,$mul
$push $pull $pop
$rename
$currentDate,$min $max
Update
MongoDB
$and:[{condition1},{condition2}]
$and
$not
$or
$nor
Logical
$gt $gte
$lt $lte
$eq,$ne
$in $nin $all
Comparison
{FIELD :{$operator: VALUE}}

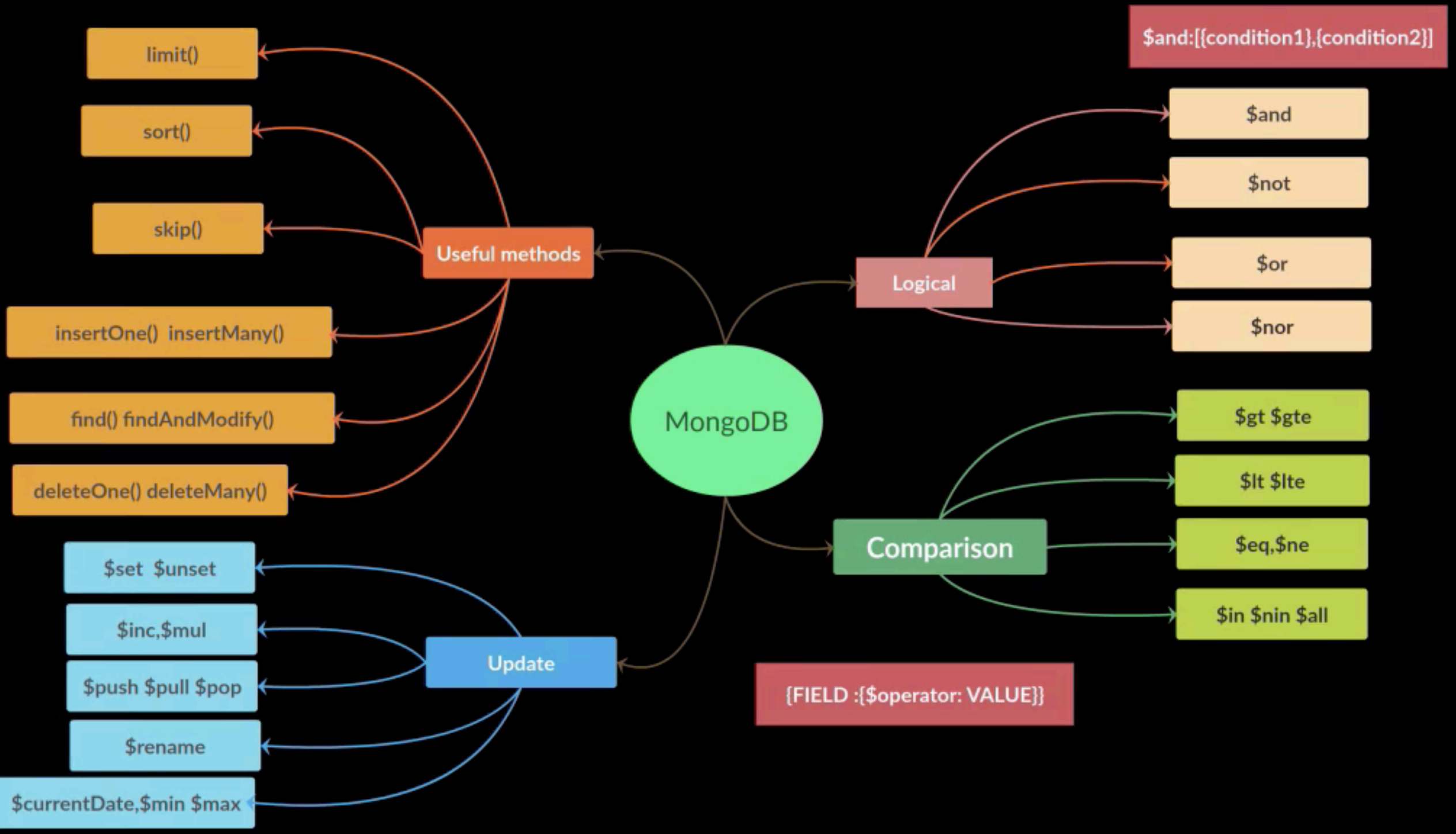
```
find(QUERY, FIELDS TO SELECT)
```

```
find(QUERY with Conditions, FIELDS TO SELECT)
```

```
db.getCollection('regions').find({QUERY},{FIELDS TO SELECT})

db.getCollection('regions').find({}).limit(2)

db.getCollection('regions').find({}).skip(2)

db.getCollection('regions').find().count()

db.getCollection('regions').find({RegionDescription:"Northern"})

db.getCollection('regions').find({RegionDescription:"Northern"})

db.getCollection('regions').find({RegionDescription:"Northern"},{RegionID:1})

db.getCollection('regions').find({RegionDescription:"Northern"},{RegionID:1,_id:0})

db.getCollection('regions').find({"RegionID" : 1,"RegionDescription" : "Eastern"})
```

count()
limit(<Number of documents>)
skip(<Number of documents>)
sort(1 for asc -1 for Desc order)

Now Logical and Comparison



Update Documents

db.collection.update(query, update, options)	upsert
Replace entire document	$inc $mul
update specific fields using $set	$unset
Multiple Updates	$currentDate
updateOne()	$min $max
updateMany()	update Arrays

Schema Design

Terminology

RDBMS		MongoDB
Database	→	Database
Table	→	Collection
Index	→	Index
Row	→	Document
Join	→	Embedding & Linking

What is a Document?

```
{
     _id: “123”,
     title: "MongoDB: The Definitive Guide",
     authors: [
        { _id: "kchodorow", name: "Kristina Chodorow“ },
        { _id: "mdirold", name: “Mike Dirolf“ }
     ],
     published_date: ISODate("2010-09-24"),
     pages: 216,
     language: "English",
     thumbnail: BinData(0,"AREhMQ…==")
     publisher: {
          name: "O’Reilly Media",
          founded: 1980,
          locations: ["CA”, "NY” ]
     }
}
```

Documents Map to Language Constructs

```
// Java: maps
DBObject query = new BasicDBObject("publisher.founded", 1980));
Map m = collection.findOne(query);
Date pubDate = (Date)m.get("published_date"); // java.util.Date


// Javascript: objects
m = collection.findOne({"publisher.founded" : 1980});
pubDate = m.published_date;     // ISODate
year = pubDate.getUTCFullYear();


# Python: dictionaries
m = coll.find_one({"publisher.founded" : 1980 });
pubDate = m["pubDate"].year    # datetime.datetime
```

Indexing

Indexing

```
Index are the base for high performance find Queries.

Without indexes mongodb queries through all the documents

Index increases storage amount but completely reduces the query execution time.

Indexes will affect insert and update operation speediness.

Default _id Index
MongoDB creates a unique index on the _id field during the creation of a collection.You can delete this.
```

```
There are around 5 types of indexes in mongodb

  1. Single field index
  2. Compound index  -- Multiple field indexes
  3. MultiKey index  -- indexig arrays of values
  4. Text indexes    -- for text search
  5. GeoSpatial indexes
```

```
getindexes()    -- Lists all the indexes
ensureindex() / createindex() --- create new index
dropindex() / dropindexes()   ---  remove index
reindex() --- Rebuild or modify indexes

What I love is Partial Indexes
```

List all the Indexes in a Collection

Search without indexing

Search with indexing

Check the default _id index

drop index

create a compound index

create a multikey index

Some Key Documentation

Mongodump & Restore

Tools

- Robomongo / Robo3T
- MongodbBooster / NoSQLBooster
- MongoChef / Studio 3T (Not Free)
- NoSql Manager / Mongodbmanager