

Topic 5: Document Databases

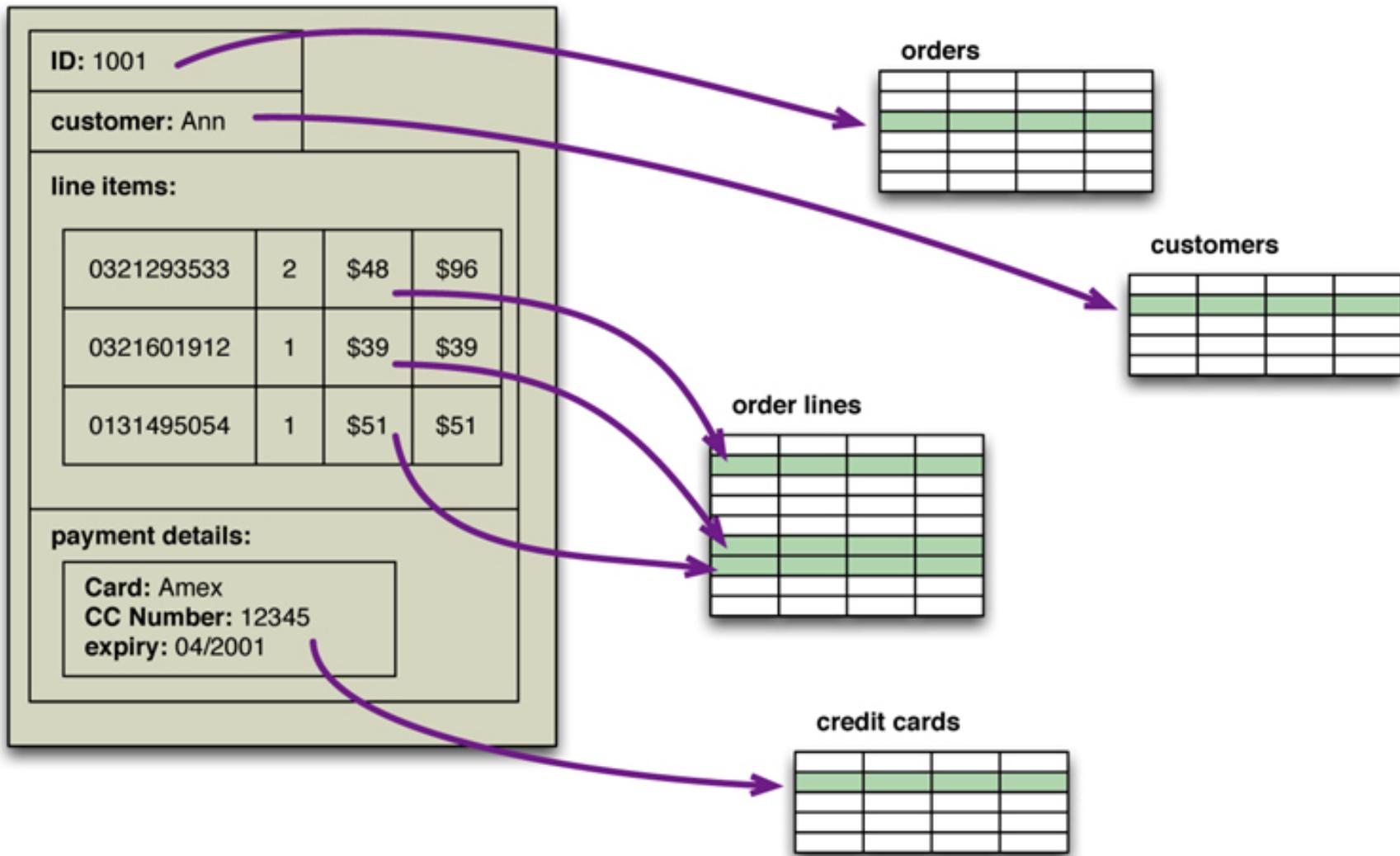


341 systems in ranking, February 2018

Rank			DBMS	Database Model	Score		
Feb 2018	Jan 2018	Feb 2017			Feb 2018	Jan 2018	Feb 2017
1.	1.	1.	Oracle 	Relational DBMS	1303.28	-38.66	-100.55
2.	2.	2.	MySQL 	Relational DBMS	1252.47	-47.24	-127.83
3.	3.	3.	Microsoft SQL Server 	Relational DBMS	1122.04	-26.03	-81.42
4.	4.	4.	PostgreSQL 	Relational DBMS	388.38	+2.19	+34.70
5.	5.	5.	MongoDB 	Document store	336.42	+5.47	+0.92
6.	6.	6.	DB2 	Relational DBMS	189.97	-0.30	+2.07
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	130.07	+3.37	-3.32
8.	↑ 9.	↑ 10.	Redis 	Key-value store	127.02	+3.88	+12.98
9.	↑ 10.	↑ 11.	Elasticsearch 	Search engine	125.32	+2.76	+17.01
10.	↓ 8.	↓ 7.	Cassandra 	Wide column store	122.78	-1.10	-11.60



Impedance mismatch problem



Object vs. Relational Databases

A relational database is like a garage that forces you to take your car apart and store the pieces in little drawers.

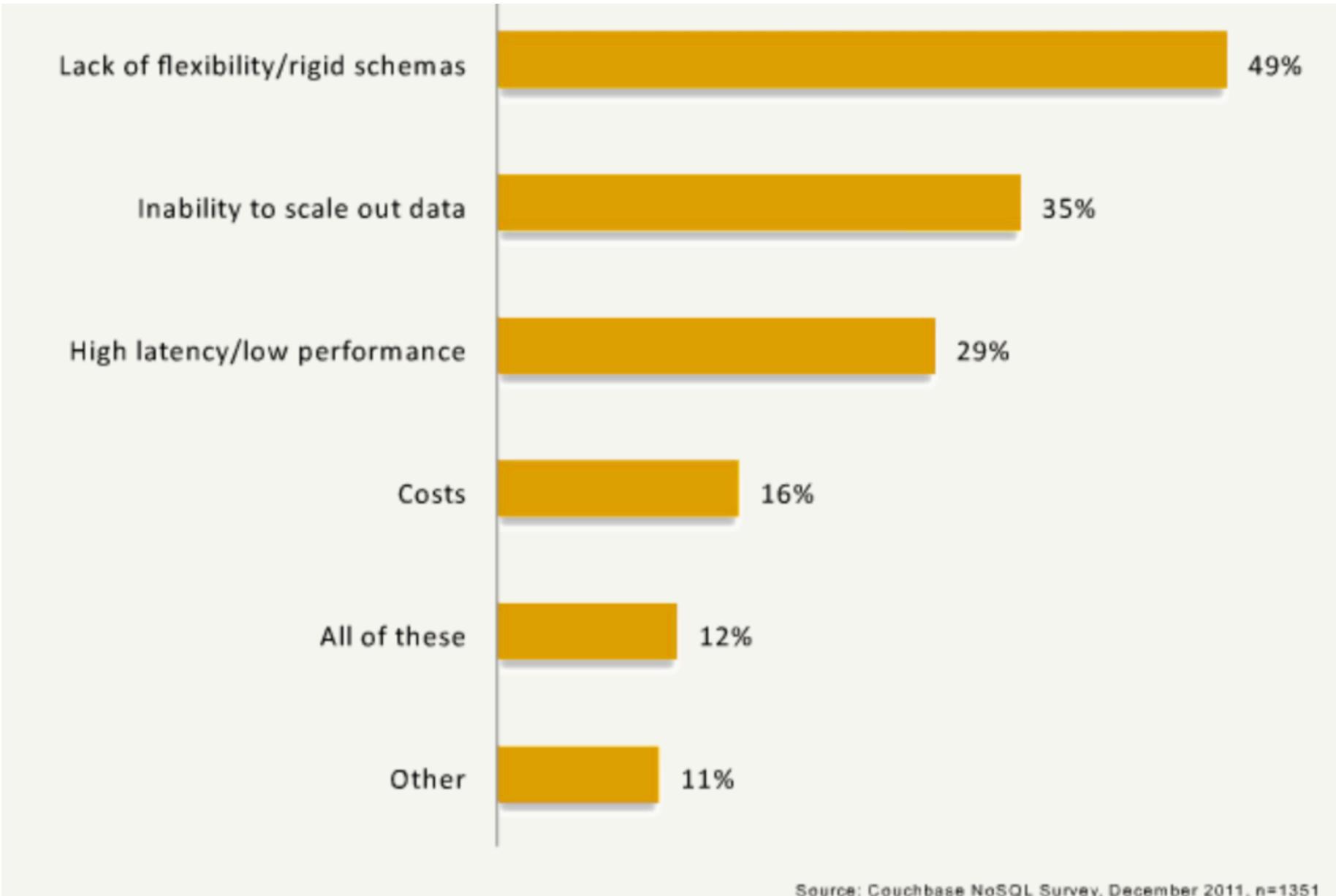
-- Object-oriented data community, mid-1990s

An object database is like a closet which requires that you hang up your suit, with tie, underwear, belt, socks and shoes all attached.

-- David Ensor, same period.



Drivers of NoSQL



Document Databases

A **document database** is a non-relational database that stores data as structured documents, usually XML or JSON formats.

Designed to be *simple*, *flexible*, and *scalable*

```
{ "users": [ {  
    "firstName": "Ray",  
    "lastName": "Villalobos",  
    "joined": {  
        "month": "January",  
        "day": 12,  
        "year": 2012  
    },  
    {  
        "firstName": "John",  
        "lastName": "Jones",  
        "joined": {  
            "month": "April",  
            "day": 28,  
            "year": 2010  
        }  
    }  
]
```



XML (eXtensible Markup Language)

- Used as an information storage standard
- XML + CSS (Cascading Style Sheets) enables development of websites that separate data and format
- The foundation for web-service specifications such as SOAP
- A rich ecosystem of tools and standards that includes:

Xpath – Retrieving specific elements of an XML document

Xquery – Interrogating XML documents

XML schema – Describes XML documents within a namespace and used for document validation

XSLT – for transforming XML into other formats such as HTML

DOM (Document Object Model) – An OO API for navigating XML docs.

```
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
</CATALOG>
```

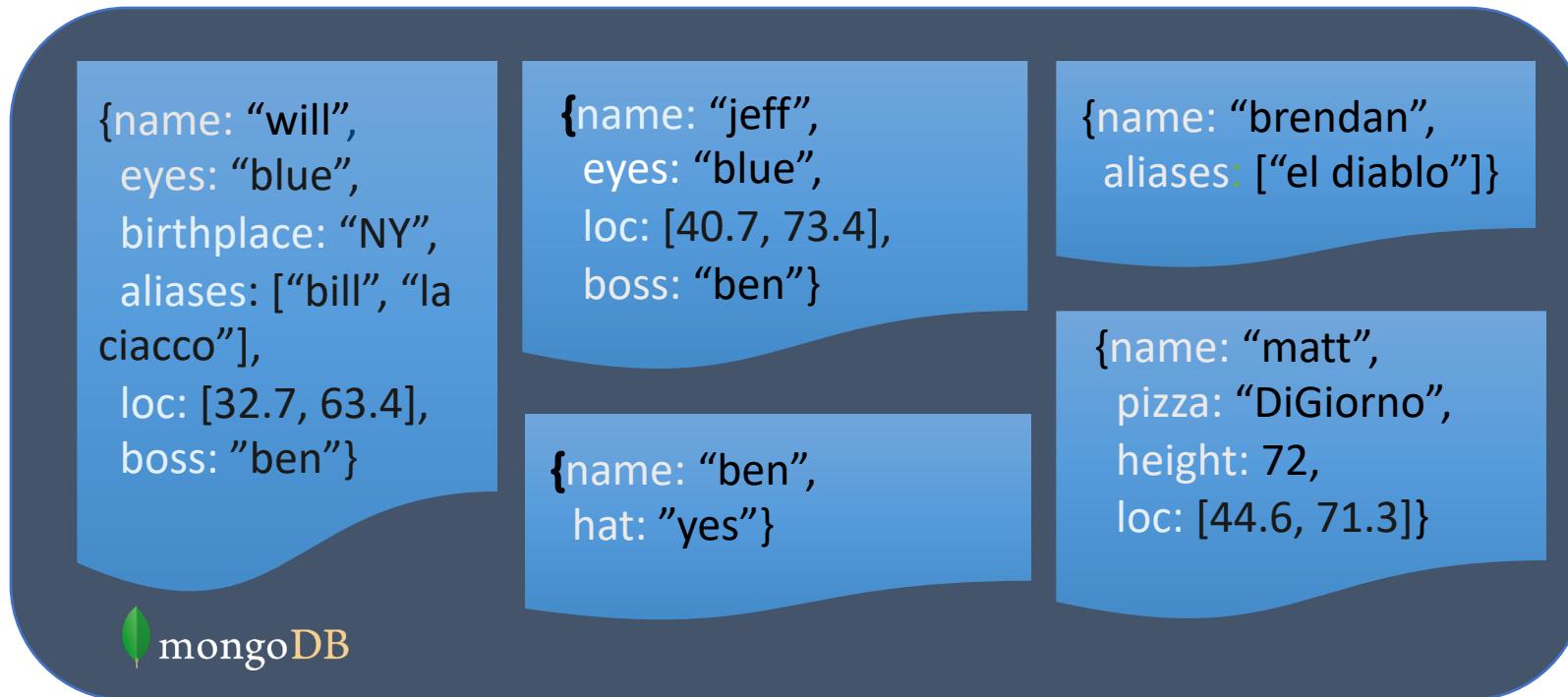
Why have document databases flourished since 2008?

- They address the *impedance mismatch* problem associated with object-oriented developers needing to persist their data to relational databases. (Object-Relational Mapping)
- Self-describing formats support *ad hoc* query by value absent in pure key-value stores.
- Aligned with web-service based programming models using JSON / XML as a transport layer.

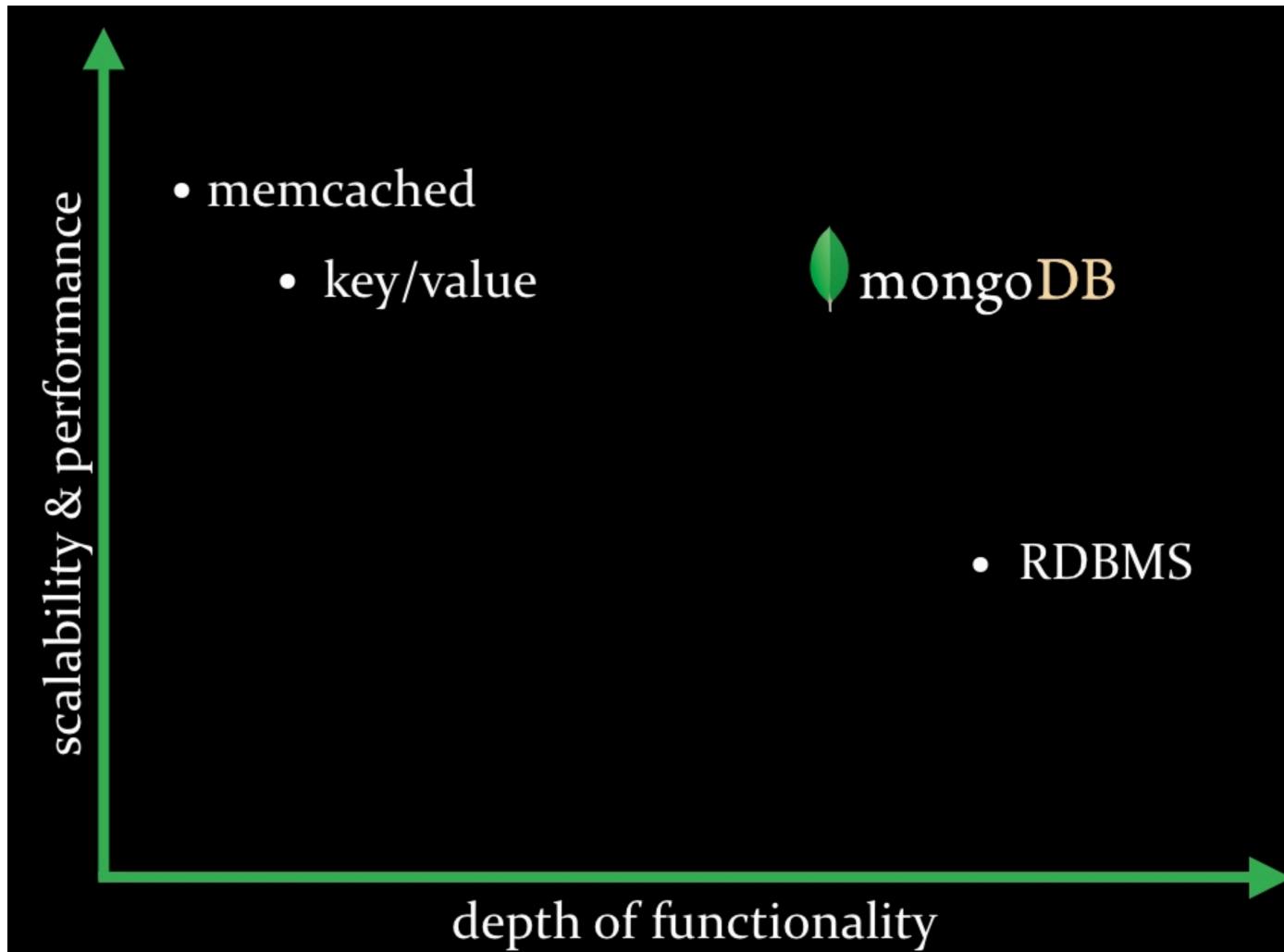


Schema-less

- MongoDB does not need any pre-defined data schema
- Every document in a collection could have different data



Document databases: A happy medium

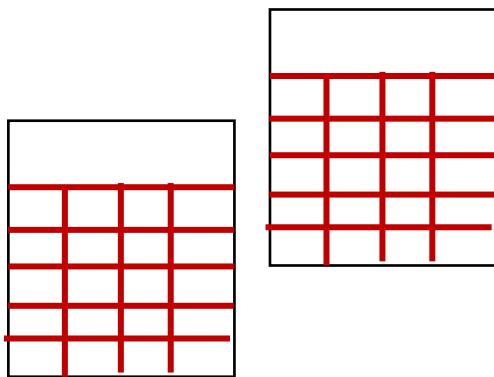


Document databases: a happy medium

Relational Databases

Pre-defined / rigid schema

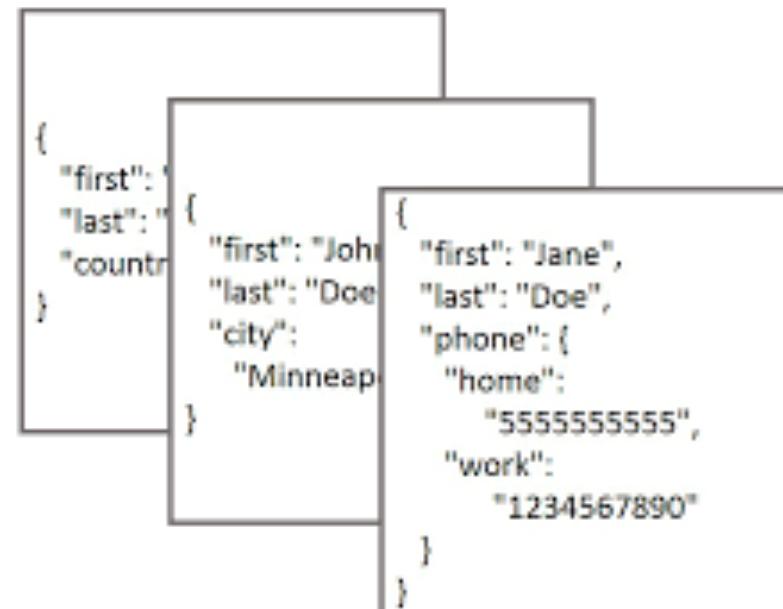
SQL: Structured Query Language



Document Databases

Data is self-describing and flexible

Query by content



Key-Value Stores

Data is simple hash tables

Query by key only (with exceptions)

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623



Relational DB vs. Document DB

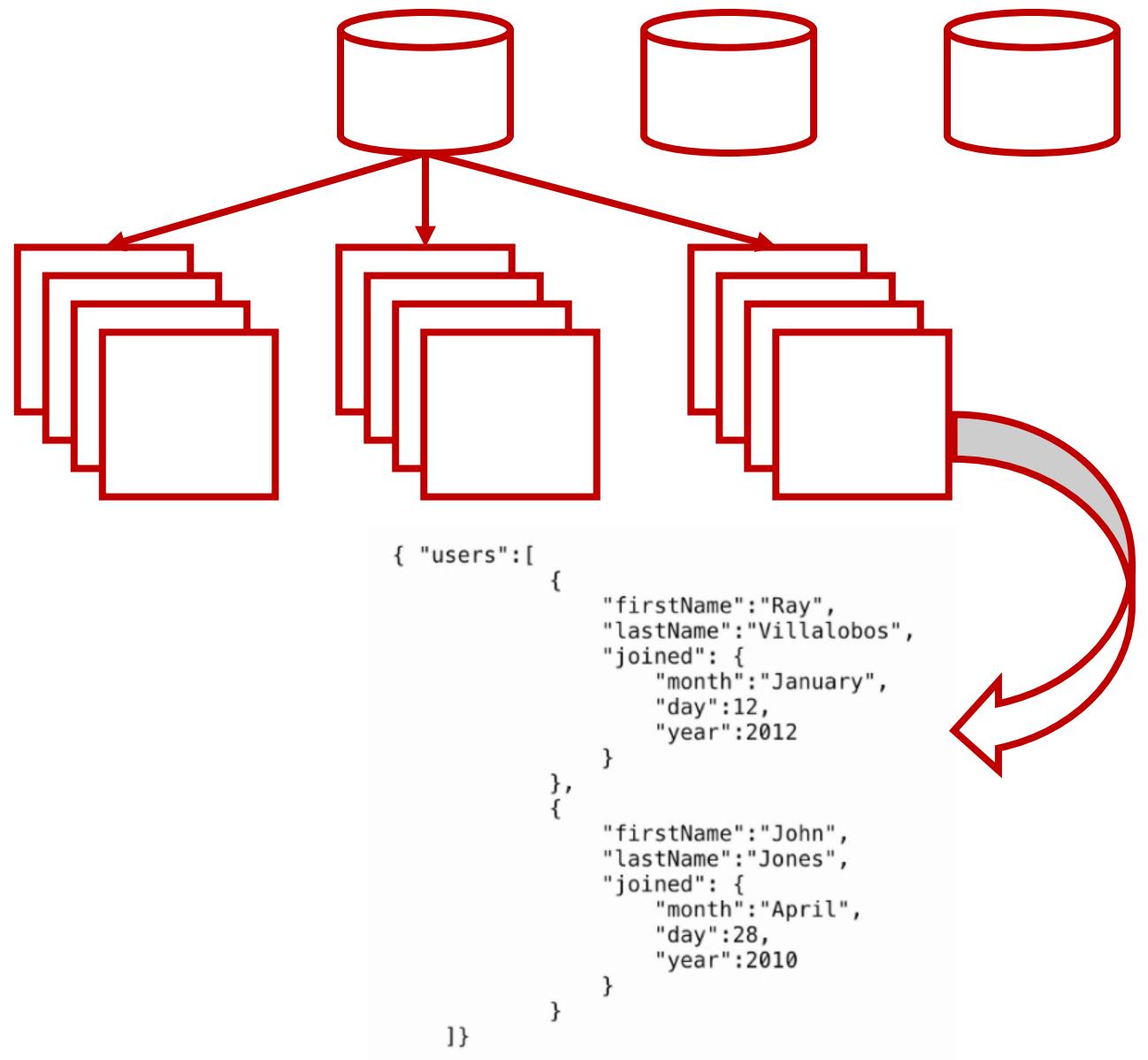
RDBMS		MongoDB
Database	→	Database
Table, View	→	Collection
Row	→	Document (BSON)
Column	→	Field
Index	→	Index
Join	→	Embedded Document
Foreign Key	→	Reference
Partition	→	Shard

- A collection is not strict about what it stores (Schema-less)
- Hierarchy is evident in the design
- Embedded Document to represent relation.

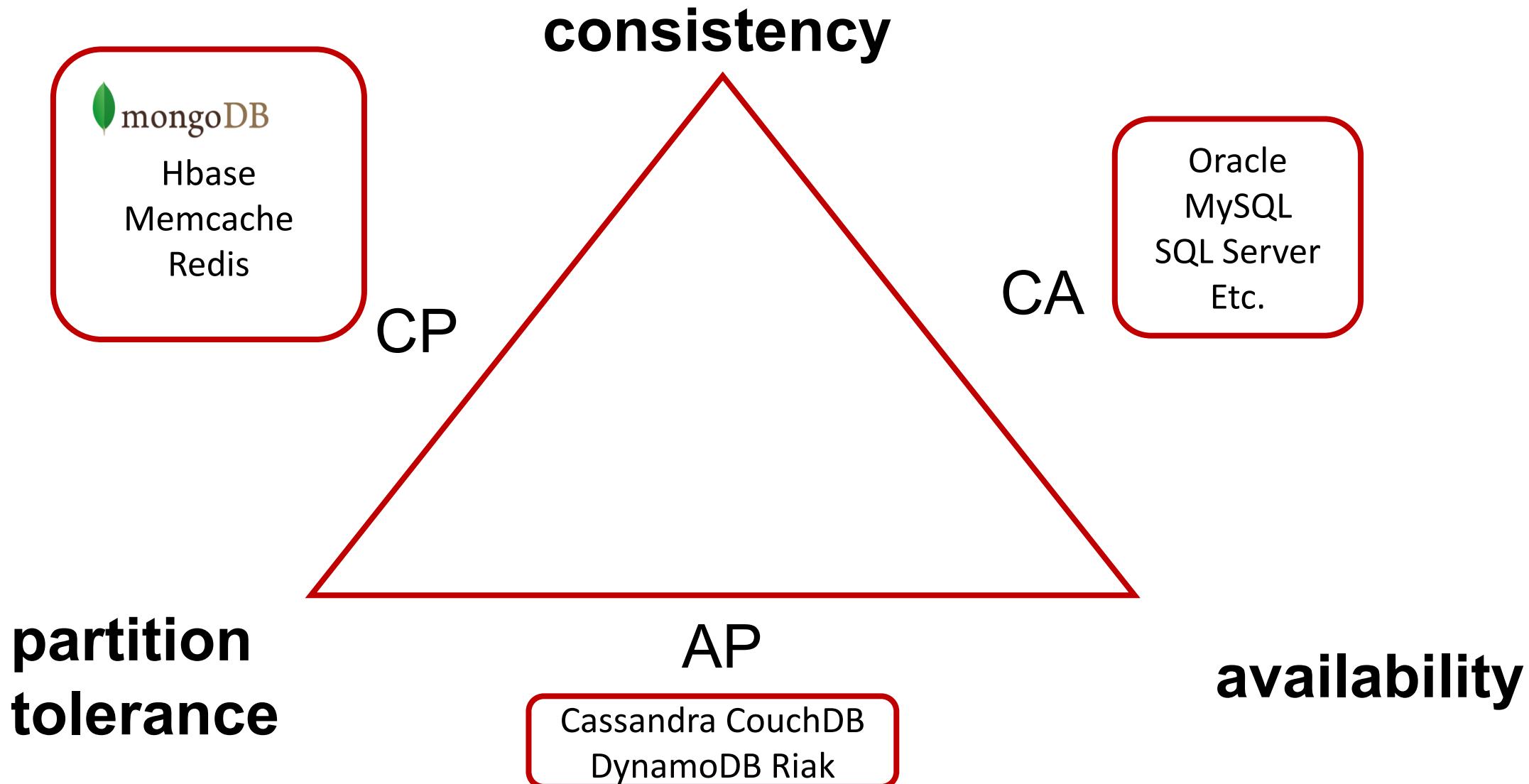


MongoDB: Hierarchical Objects

- A MongoDB instance may have zero or more ‘databases’
- A database may have zero or more ‘collections’.
- A collection may have zero or more ‘documents’.
- A document may have one or more ‘fields’.
- MongoDB ‘Indexes’ function much like their RDBMS counterparts.



The CAP Triangle Revisited



What is MongoDB?

- Developed by MongoDB, Inc. (Formerly 10gen Inc.)
 - Founded in 2007
 - IPO October 2017
- A document-oriented NoSQL database
 - Hash-based, *schema-less database*
 - No Data Definition Language
 - Uses BSON (Binary JSON) format
- Written in C++
- Supports APIs (drivers) in many computer languages:
JavaScript, Python, Ruby, Perl, Java, Java Scala, C#, C++, Haskell, Erlang



MongoDB Features

- **Rich query support:** We can query the database as we do with SQL databases. It has a large query set that supports insert, update, delete and select operations. MongoDB supports fields, range queries, and regular expressions. Queries also support the projection where they return a value for specific keys.
- **Indexing:** MongoDB supports primary and secondary indices in its fields.
- **Replication:** Replication means providing more than one copy of data. MongoDB provides multiple copies of data with multiple servers. It provides fault tolerance, if one database server goes down, the application uses other database servers.



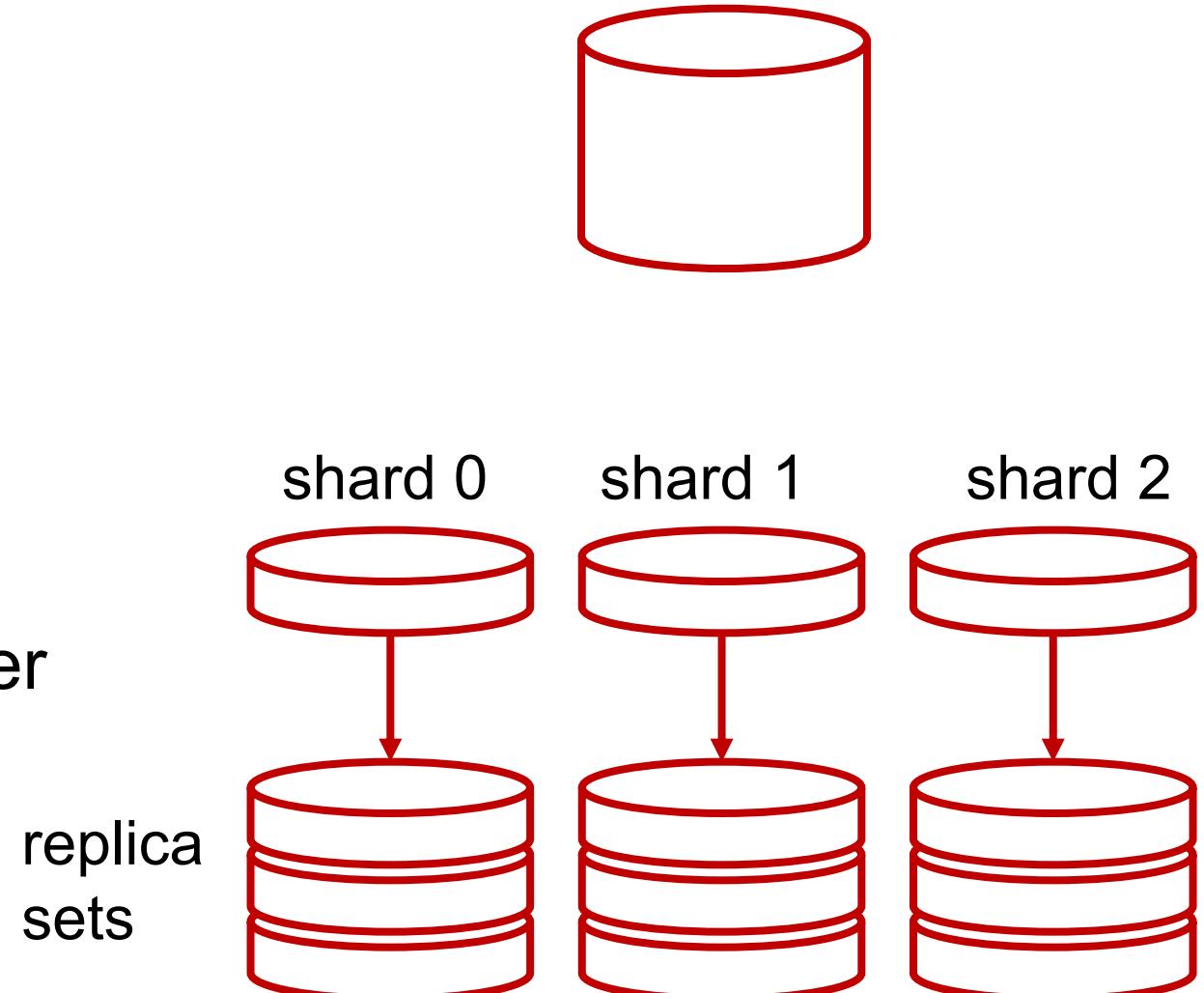
MongoDB Features

- **Load balancing:** Replica sets provide multiple copies of data. MongoDB can scale read operation by client request directly to the secondary node. This divides loads across multiple servers.
- **File storage:** We can store documents up to 6 MB directly to the MongoDB JSON field. For documents exceeding the size limit of 16 MB, MongoDB provides GridFS to store in chunks.
- **Aggregation:** The aggregate function takes a number of records and calculates single results like sum, min, and max. MongoDB provides a data pipeline and multistage pipeline to move large data to the aggregate function which improves performance.



MongoDB Features

- Rich query language supporting search by field, range, regular expressions
- Primary, secondary, composite indexing
- Replica sets with automatic failover
- Load balancing via sharding
- Joins via \$lookup operator



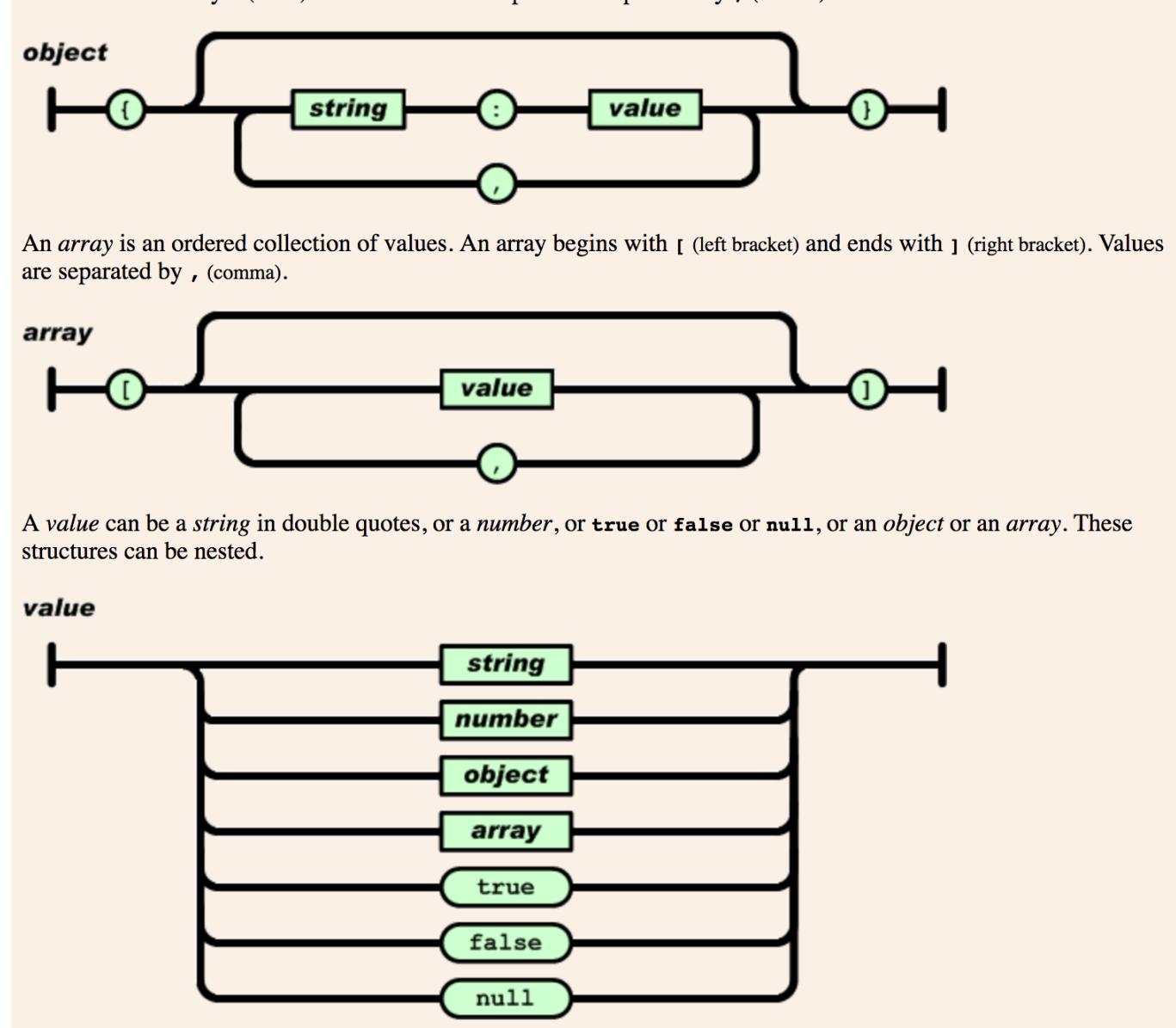
JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.



BSON (Binary JSON)



BSON [*bee · sahn*], short for Binary [JSON](#), is a binary-encoded serialization of JSON-like documents. Like JSON, BSON supports the embedding of documents and arrays within other documents and arrays. BSON also contains extensions that allow representation of data types that are not part of the JSON spec. For example, BSON has a Date type and a BinData type.

BSON can be compared to binary interchange formats, like [Protocol Buffers](#). BSON is more "schema-less" than Protocol Buffers, which can give it an advantage in flexibility but also a slight disadvantage in space efficiency (BSON has overhead for field names within the serialized data).

BSON was designed to have the following three characteristics:

1. **Lightweight**

Keeping spatial overhead to a minimum is important for any data representation format, especially when used over the network.

2. **Traversable**

BSON is designed to be traversed easily. This is a vital property in its role as the primary data representation for [MongoDB](#).

3. **Efficient**

Encoding data to BSON and decoding from BSON can be performed very quickly in most languages due to the use of C data types.



BSON Specification 1.1

Specification Version 1.1

BSON is a binary format in which zero or more ordered key/value pairs are stored as a single entity. We call this entity a *document*.

The following grammar specifies version 1.1 of the BSON standard. We've written the grammar using a pseudo-[BNF](#) syntax. Valid BSON data is represented by the `document` non-terminal.

Basic Types

The following basic types are used as terminals in the rest of the grammar. Each type must be serialized in little-endian format.

<code>byte</code>	1 byte (8-bits)
<code>int32</code>	4 bytes (32-bit signed integer, two's complement)
<code>int64</code>	8 bytes (64-bit signed integer, two's complement)
<code>uint64</code>	8 bytes (64-bit unsigned integer)
<code>double</code>	8 bytes (64-bit IEEE 754-2008 binary floating point)
<code>decimal128</code>	16 bytes (128-bit IEEE 754-2008 decimal floating point)

Non-terminals

The following specifies the rest of the BSON grammar. Note that quoted strings represent terminals, and should be interpreted with C semantics (e.g. "\x01" represents the byte 0000 0001). Also note that we use the * operator as shorthand for repetition (e.g. (""\x01"*2) is "\x01\x01"). When used as a unary operator, * means that the repetition can occur 0 or more times.

```
document ::= int32 e_list "\x00"
e_list  ::= element e_list
```

BSON Document. `int32` is the total number of bytes comprising the document.



JSON → BSON

The following are some example documents (in JavaScript / Python style syntax) and their corresponding BSON representations.

{ "hello": "world" }	→	\x16\x00\x00\x00 \x02 hello\x00 \x06\x00\x00\x00world\x00 \x00	// total document size // 0x02 = type String // field name // field value // 0x00 = type EOO ('end of object')
{ "BSON": ["awesome", 5.05, 1986] }	→	\x31\x00\x00\x00 \x04BSON\x00 \x26\x00\x00\x00 \x02\x30\x00\x08\x00\x00\x00awesome\x00 \x01\x31\x00\x33\x33\x33\x33\x33\x33\x14\x40 \x10\x32\x00\xc2\x07\x00\x00 \x00 \x00	



Embedded vs. References Data Models

```
{  
    "_id" : ObjectId("59769440c5160d8a42a02358"),  
    "firstName" : "John",  
    "lastName" : "Cent",  
    "age" : 45,  
    "email" : "john.cent@abc.com",  
    "address" : [  
        {  
            "city" : "NY",  
            "state" : "NY",  
            "country" : "USA"  
        }  
    ]  
}
```

```
{  
    "_id" : ObjectId("59769578c5160d8a42a0235a"),  
    "firstName" : "John",  
    "lastName" : "Cent",  
    "age" : 45,  
    "email" : "john.cent@abc.com",  
    "address" : [  
        ObjectId("597694f5c5160d8a42a02359")  
    ]  
}
```



```
{  
    "_id" : ObjectId("597694f5c5160d8a42a02359"),  
    "city" : "NY",  
    "state" : "NY",  
    "country" : "USA"  
}
```



Product Catalogs / e-commerce websites

The screenshot shows a product page for an Italian Shoemakers Napa Sandal. The main image is a brown leather sandal with blue and orange straps. Below it are smaller thumbnail images of the shoe from different angles. A green callout bubble labeled "Product images" points to these thumbnails.

General Information (green callout): Item # 295144, UPC # 885655921272. Description: When the weather gets hot, stay cool in the Napa platform sandal from Italian Shoemakers. This simple slide is the perfect match for your favorite sundress!

Localized Description (green callout):

- Leather upper
- Square open toe
- ¾" platform, ¾" wood cutout heel
- Synthetic sole
- Imported

[View more Women's Italian Shoemakers Shoes](#)

Average Overall Rating: ★★★★☆

Product Details:
Runs Short: Runs Long:
Runs Narrow: Runs Wide:
Not Comfy: Very Comfy:

[Read Reviews](#) [Write a Review](#) [Printable Reviews](#)

External Information (green callout): [Find It In Store](#) [Share:](#) [Facebook](#) [Pinterest](#) [Twitter](#)

Italian Shoemakers Napa Sandal
\$49.95
Compare at \$72.00

1. Select Size: 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9
2. Select Width: M
3. Select Color: Brown, Black, Grey
4. Select Quantity: 1

size chart

ADD TO BAG **ADD TO WISH LIST**

NEED HELP? Chat with a Shoe Lover →



Flexible product descriptions

➤ db.item.findOne()

```
{ _id: "301671", // main item id
  department: "Shoes",
  category: "Shoes/Women/Pumps",
  brand: "Guess",
  thumbnail: "http://cdn.../pump.jpg",
  image: "http://cdn.../pump1.jpg", // larger version of thumbnail
  title: "Evening Platform Pumps",
  description: "Those evening platform pumps put the perfect
finishing touches on your most glamourous night-on-the-town
outfit",
  shortDescription: "Evening Platform Pumps",
  style: "Designer",
  type: "Platform",
  rating: 4.5, // user rating
  lastUpdated: Date("2014/04/01"), // last update time
  ...
}
```

Product Search

- **Get item by id**

```
db.item.findOne( { _id: "301671" } )
```

- **Get item from Product Ids**

```
db.item.findOne( { _id: { $in: ["301671", "301672"] } } )
```

- **Get items by department**

```
db.item.find({ department: "Shoes" })
```

- **Get items by category prefix**

```
db.item.find( { category: /(^Shoes|Women)/ } )
```

- **Indices**

productId, department, category, lastUpdated



Compound Libraries

- Store Chemical Compounds – Fingerprints
- Want to find compounds which are “close” to a given compound
- Need to return quickly a small set of reasonable candidates
- Few researchers working concurrently
- Use Tanimoto association coefficient to compare two compounds based on their common fingerprints

```
{  
    "compound_cid" : "46200001" ,  
    "smiles" :  
        "CCC1C(C(C(C(=NOCC=CCN2CCCCC2)C(CC(C(C(C(C(=O)O1)C)OC3CC(C(C(O3)C)O)  
            (C)OC)C)OC4C(C(CC(04)C)N(C)CC5=CC=CC=C5)O)(C)O)C)C)O)(C)O" ,  
    "fingerprint_count" : 120 ,  
    "fingerprints" : [  
        "0[N]1[C O]2[C C C]" ,  
        "0[N]1[C O]2[C C C]3[C C C C C]" ,  
        "0[C]1[C C C]2[C C N O]3[C C C C O O]" ,  
        "0[C]1[C C C]2[C C C C O]3[C C N O]" ,  
        "0[O]1[C]2[C O]3[C C C]" ,  
        "0[C]1[C O O]2[C C C O]" ,  
        "0[C]1[C C C]2[C C]" ,  
        "0[C]1[C]2[C]3[C O]" ,  
        "0[C]1[C C N]2[C C C C O]3[C C C O]" ,  
        ... ] ,  
}
```

$$T = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$



Genomics

- Very large base of DNA sample sequences
 - Origin, collection method, sequence, date, ...
- Enumeration of mutations relative to reference sequence
 - Positions, mutation type, base
- Need to retrieve efficiently all sequences showing a particular mutation
- Similar to Content Management System pattern
- Add tag array in sequence document with mutation names
- Index tag array
- Queries looking for affected sequences are indexed and very fast
- Easy to setup, flexible representation and details for sequences, flexible evolution
- Can scale to massive volumes

	1	2	3	4	5	6	7	9	10
reference	T	T	C	C	T	A	-	T	T
sample	T	G	C	C	-	A	G	T	T

mongoDB



MongoDB Installation

Atlas

Community Server

Enterprise Server

Ops Manager

Compass

Connector for BI

[Current Release](#) | [Previous Releases](#) | [Development Releases](#)

Current Stable Release (3.6.2)

01/10/2018: [Release Notes](#) | [Changelog](#)
Download Source: [tgz](#) | [zip](#)



Windows



Linux



OSX

Version:

OS X 10.7+ 64-bit w/SSL x64 ▾

Package Manager:

[Instructions for installing with Homebrew](#)

Binary: [Installation Instructions](#) | [All Version Binaries](#)

DOWNLOAD (tgz)

https://fastdl.mongodb.org/osx/mongodb-osx-ssl-x86_64-3.6.2.tgz



Northeastern University

Or on Mac OSX: \$ brew install mongodb

MongoDB Detailed Instructions

<https://docs.mongodb.com/manual/administration/install-community/>

[Install MongoDB](#) > [Install MongoDB Community Edition](#)



Install MongoDB Community Edition

These documents provide instructions to install MongoDB Community Edition.

[Install on Linux](#)

Install MongoDB Community Edition and required dependencies on Linux.

[Install on macOS](#)

Install MongoDB Community Edition on macOS systems from Homebrew packages or from MongoDB archives.

[Install on Windows](#)

Install MongoDB Community Edition on Windows systems and optionally start MongoDB as a Windows service.



MongoDB Compass



SOLUTIONS CLOUD CUSTOMERS RESOURCES ABOUT US

MongoDB Compass

As the GUI for MongoDB, MongoDB Compass allows you to make smarter decisions about document structure, querying, indexing, document validation, and more. Compass enables you to:

- View, add, and delete databases and collections
- Monitor performance in real time and identify slow queries
- View and interact with documents with full CRUD functionality
- Build and run ad hoc queries
- Visually explore the structure of data in your database
- View and optimize query performance with visual explain plans
- Manage indexes: view stats, create, and delete
- Create and modify data validation rules

Compass is available as part of the MongoDB Professional, MongoDB Enterprise Advanced, and MongoDB Atlas Professional subscriptions; [contact us](#) if you're interested in learning more.

Versions:

1.11.2 (Community Edition Stable)

Platforms:

OS X 64-bit (10.10+)

[Download](#)

[Documentation](#)
[License Agreement](#)



Northeastern University

Running MongoDB

Add \$MONGO_HOME to your path

```
export APPS=/Users/rachlin/apps  
export MONGO_HOME=$APPS/mongo/mongodb-4.0.7  
export PATH=$MONGO_HOME/bin:$PATH
```

Set up a directory for the database storage and run server

```
$ mkdir -p $HOME/data/mongodb  
$ mongod --dbpath $HOME/data/mongodb
```

Run the command-line client (or launch Compass)

```
$ mongo
```



Mongo Demo

- The basics of databases, collections, and documents
- Searching with `find()`

