02/10/25 - Document Databases and MongoDB

Document Database

A non-relational database that stores data as structured documents, usually in JSON

Designed to be simple, flexible, and scalable

JSON (JavaScript Object Notation)

A lightweight data-interchange format

- Easy for humans to read and write
- Easy for machines to parse and generate

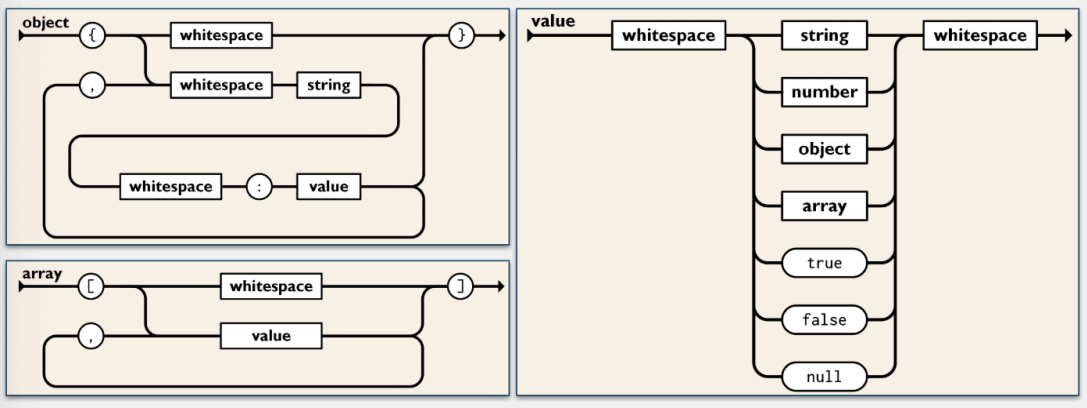
Built on two structures

- A collection of name/value pairs
- An ordered list of values

These are two universal data structures

- Thus, JSON makes a great data interchange format

JSON Syntax



BSON (Binary JSON)

Binary-encoded serialization of a JSON-like document structure

XML (eXtensible Markup Language)

Precursor to JSON as data exchange format

XML + CSS → web pages that separated content and formatting

Structurally similar to HTML, but tag set is extensible (can be defined)

XML-Related Tools/Technologies

There are various tools that were written for XML

Why Document Databases?

Document databases address the impedance mismatch problem between object persistence in OO systems and how relational DBs structure data

- OO Programming → Inheritance and Composition of types

Amazon example

- When you search something on Amazon (e.g., hard drives), you will get filters on the left (e.g., speed, size)
- If you search for something else (e.g. toothbrushes), you will get different filters (e.g., electric/non-electric, bristle stiffness)

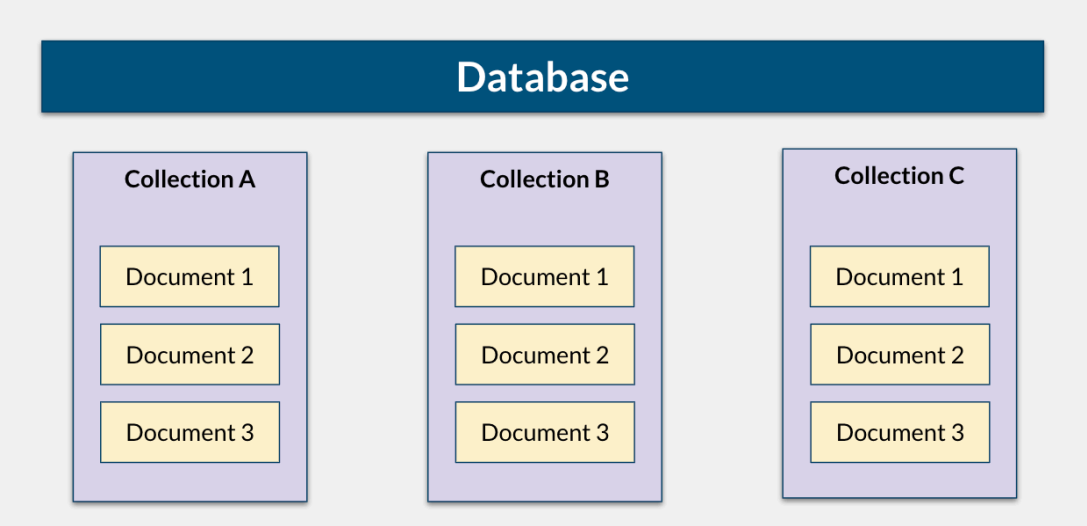
MongoDB

Started in 2007 after Doubleclick was acquired by Google

- Three of its veterans realized the limitations of relational databases for serving >400,000 ads per second

Short for Humongous Database

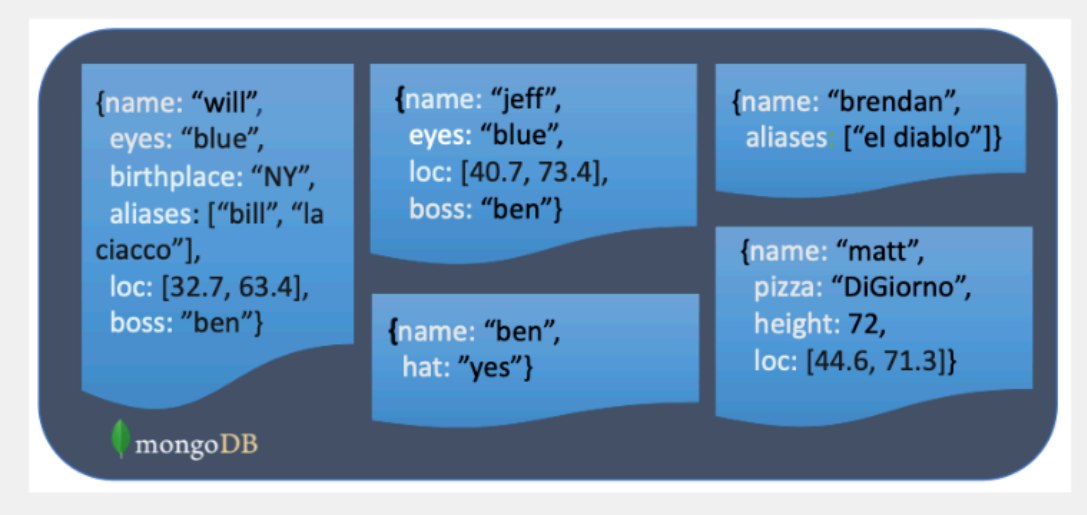
Structure



Documents

No predefined schema for documents is needed

Every document in a collection could have different data/schema



Relational vs Mongo/Document DB

RDBMS	MongoDB
Database	Database
Table/View	Collection
Row	Document

Column	Field
Index	Index
Join	Embedded Document
Foreign Key	Reference

Features

Rich Query Support

- Robust support for all CRUD operations

Indexing

Replication

Load balancing

Versions

MongoDB Atlast

MongoDB Enterprise

MongoDB Community

Interacting with MongoDB

mongosh → MongoDB Shell

- CLI tool for interacting with a MongoDB instance

MongoDB Compass: free, open-source GUI to work with a MongoDB database

DataGrip and other third party tools

Every major language has a library to interface with MongoDB

- PyMongo (Python), Mongoose (JavaScript/node), ...

mongosh - Mongo Shell

find(...) is like SELECT

```
collection.find({ filters }, { projections })
```

```
SELECT * FROM users;
```

```
use mflix
db.users.find()
```

```
SELECT *
FROM users
WHERE name = "Davos Seaworth";
```

```
db.users.find({"name": "Davos Seaworth"})
```

```
SELECT *
FROM movies
WHERE rated in ("PG", "PG-13")
```

```
db.movies.find({rated: {"$in":[ "PG", "PG-13" ]}})
```

Comparison Operators

Name	Description
`$eq`	Matches values that are equal to a specified value.
`$gt`	Matches values that are greater than a specified value.
`$gte`	Matches values that are greater than or equal to a specified value.
`$in`	Matches any of the values specified in an array.
`$lt`	Matches values that are less than a specified value.
`$lte`	Matches values that are less than or equal to a specified value.
`$ne`	Matches all values that are not equal to a specified value.
`$nin`	Matches none of the values specified in an array.