mongoDB®

MongoDB Overview

- Developed by 10gen in 2007
- Publicly available in 2009
- Open-source database which is controlled by 10gen
- Document oriented database → stores JSON documents
- Stores data in binary JSON

Install MongoDB

- Install MongoDB by downloading community edition
  - (https://www.mongodb.com/download-center/community)
- Linux and Mac Users
  - Extract the downloaded file somewhere in the disk.
  - Set the environment path to use the tools without going to the bin directory in the ~/.bash_profile or ~/.basrc file.
- Ubuntu (20.04) Mongo installation
  - terminal> wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
  - terminal> echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
  - terminal> sudo apt-get update
  - terminal> sudo apt-get install -y mongodb-org
- Windows Users
  - Install the MongoDB by following all the steps in the installation wizard
  - Set the the environment path to include the <path>/bin

JSON

- Java Script Object Notation
- Hierarchical way of organizing data
- Defined as part of the JS language by JavaScript creator Douglas Crockford (2000).
- JavaScript objects are associative containers, wherein a string key is mapped to a value
- JSON shows up in many different cases.
  - APIs
  - Configuration files
  - Log messages
  - Database storage
- JSON is not ideal for usage inside of a database.
  - JSON is a text-based format, and text parsing is very slow
  - JSON's readable format is far from space-efficient, another database concern
  - JSON only supports a limited number of basic data types
- Mongo stores JSON data into Binary form.

JSON have

- String -> “hello”   “name”
- Numbers -> 30   1.5   -40   1.2e10
- Booleans -> true   false
- Array   -> [“Jan”, “feb”, “Mar”]   [10 ,20 ,30]
- Object -> {“key”,”value”}   {“age”: 20}

Key is always a string

At top level typically have an array or object

BSON

- BSON simply stands for “Binary JSON”
- Binary structure encodes type and length information, which allows it to be parsed much more quickly
- It has been extended to add some optional non-JSON-native data types
- It allows for comparisons and calculations to happen directly on data
- MongoDB stores data in BSON format both internally, and over the network
- Anything you can represent in JSON can be natively stored in MongoDB

	JSON	BSON
Encoding	UTF-8 String	Binary
Data Support	• String • Boolean • Number • Array	• String • Boolean • Number • Integer • Float • Long • Decimal • Array • Date • Raw Binary
	Human and Machine	Machine Only

MongoDb: Data Types

data	bson	values
null	10	
boolean	8	true, false
number	1 / 16 / 18	123, 456.78, NumberInt("24"), NumberLong("28")
string	2	"...."
date	9	new Date(), ISODate("yyyy-mm-ddThh:mm:ss")
array	4	[..., ..., ..., ...]
object	3	{ ... }

JSON

```
[ {
        "_id": 1,
        "name" : { "first" : “Sanjay", "last" : “Pawar" },
        “Language" : [ “C++", “JAVA", “Python", “Kotlin“,”Go” ],    ← array of string
        "awards" : [
           { “Winner" : "Best developer", "year" : 1998 },
           { “2nd Runner-Up" : "Best Programmer", "year" : 2000 }   ← array of JSON object
        ]
  } ]
```

Mongo Server and Client

- MongoDb server (mongod) is developed in C, C++ and JS.
- MongoDb data is accessed via multiple client tools
  - mongo : client shell (JS).
  - mongofiles : stores larger files in GridFS. ✓
  - mongoimport / mongoexport : tools for data import / export. ✓
  - mongodump / mongorestore : tools for backup / restore. ✓
- MongoDb data can be accessed in application through client drivers available for all major programming languages e.g. Java, Python, Ruby, PHP, Perl, …
- Mongo shell is follows JS syntax and allow to execute JS scripts.

MongoDB Terminology

- **Database** $\longrightarrow$ Database
  - This is a container for collections like in RDMS wherein it is a container for tables
  - Each database gets its own set of files on the file system
  - A MongoDB server can store multiple databases
- **Collection** $\longrightarrow$ table
  - This is a grouping of MongoDB documents
  - A collection is the equivalent of a table which is created in any other RDB MS such as Oracle or MS SQL
  - Collections don't enforce any sort of structure
- **Document** $\longrightarrow$ row
  - A record in a MongoDB collection is basically called a document
  - The document, in turn, will consist of field name and values
- **Field** $\longrightarrow$ column
  - Field names are strings
  - A name-value pair in a document
  - A document has zero or more fields
  - Fields are analogous to columns in relational databases

```
{
"name":"ab",
"course":"de",
}
```

	mongoDB	sql
1:	database	database
2:	collection	table
3:	document	row
4:	fields	column

Document

- MongoDB stores data records as BSON documents
- Maximum size of document is 16MB
- Restrictions
  - ✓ The field name _id is reserved for use as a primary key ✓
  - Field names **cannot** contain the null character
  - Top-level field names **cannot** start with the dollar sign ($) character

_id field

key value

- Each document requires a unique _id field that acts as a primary key
- If an inserted document omits the _id field, the MongoDB driver automatically generates an ObjectId for the _id field
- Behaviors
  - By default, MongoDB creates a unique index on the _id field during the creation of a collection
  - The _id field is always the first field in the documents. If the server receives a document that does not have the _id field first, then the server will move the field to the beginning.
  - The _id field may contain values of any BSON data type, other than an array
- Autogenerated _id (of type ObjectId) will be of 12 bytes which contains
  - Timestamp: 4 bytes
  - Machine Id: 3 bytes
  - Process Id: 2 bytes
  - Counter: 3 bytes

CRUD operations

Database Operations

- List existing databases

```
> show dbs
> show databases
```

- Create and use database

```
> use <db name>
```

- Get the selected database name

```
> db
```

- Show the database statistics

```
> db.stats()
```

Collection operations

- Get the list of collections
  > **show collections**

- Create Collection
  > db.createCollection(‘contacts’)

- Drop Collection
  > **db.contacts.drop()**

Create Document (Insert data)

- Create one document

```
> db.contacts.insert({ name: ‘ravi’, mobile: ‘7709859986’ })
```

- Create many documents

```
> db.contacts.insertMany([
        { name: ‘contact 1’, address: ‘pune’ },
        { name: ‘contact 2’, address: ‘mumbai’ }
])
```

- **Note: if you are passing the _id field, make sure that it is unique. If it is not unique, the document will not get inserted**

Read/Find Documents (Query data)

- Find documents

  > db.contacts.find()

- Returns cursor on which following operations allowed
  - **hasNext():** returns if cursor can iterate further
  - **next():** returns the next document
  - **skip(n):** skips first n documents
  - **limit(n):** limit the result to n
  - **count():** returns the count of result
  - **toArray():** returns an array of document
  - **forEach(fn):** Iterates the cursor to apply a JavaScript function to each document from the cursor
  - **pretty():** Configures the cursor to display results in an easy-to-read format
  - **sort():** sorts documents
- Shell by default returns 20 records. Press ”it” for more results