Introduction to MongoDB

MongoDB is one of the top most popular modern databases which is now widely used for building modern day applications. It is different from traditional RDBMS (Relational Database Management Systems) and it is categorized as NoSQL. database because of the flexibility of schema which it provides. MongoDB is easy to use and the data is stored in the JSON (JavaScript Object Notation) like format which stores data in key-value pair and it is very easy for developers while they work on database related queries

Definition

MongoDB is a leading NoSQL (non-relational) database system that offers a unique approach to data storage and retrieval. Unlike traditional relational databases (RDBMS) structured in tables and rows, MongoDB stores data in flexible documents, similar to JSON objects.

MangoDB is a cross-platform, document oriented database that provides high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

Database

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema.

Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

The following table shows the relationship of RDBMS terminology with MongoDB.

RDMS	MangoDB
Databases	Databases
Table	Collections
Tuple/Rows	Documents
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key -id provided by mongodb itself)

Database Server and Client

mysqld/Oracle	mongod
mysqld/sqlplus	mongo

Sample Document

```
{
    id : ObjectId("5099803df3f4948bd2f98391"),
    name : {first: "Alan", last: "Turing" },

    birth:new Date('Jun 23, 1912'),

    death : new Date('Jun 07, 1954'),

    contribs: [ "Turing machine", "Turing test", "Turingery" ],

    view : NumberLong(1250000)

}
```

Features of MongoDB

* 1/0 operations are lesser compare to RDBMS due to support of embedded documents.

*select queries are faster due to faster indexing support

*rich query language

*Auto-replication feature leads to high availability

* Support of Automatic failover

*Horizontal scalability due to Sharding feature

* Support of multiple storage engine

Advantages of MongoDB

*Schema-less database

* Dynamic query by document query language.

* Scalable

* No complex joins are needed

*SQL injection is not possible.

*Search by REGEX and fields

* No need of mapping application objects to data objects

*Index on any attribute

* Fast in-place update

Organizations that use MongoDB

* Adobe

* LinkedIn

* McAfee

* FourSquare

* eBay

* MetLife

* SAP

_id (primary key)

*_id is a 12 bytes hexadecimal number

*assures the uniqueness of every document

*If you don't provide then MongoDB provides a unique id for every document

* These 12 bytes

- first 4 bytes for the current timestamp

- next 3 bytes for machine id

- next 2 bytes for process id of MongoDB server and

- remaining 3 bytes are simple incremental VALUE.

MongoDB Data Types

String - String in MongoDB must be UTF-8 valid

Integer - 32/64 bit depending upon your server

Boolean - stores a boolean (true/false) value

Double - This type is used to store floating point values

Min/Max keys - used to compare a value against the lowest and highest BSON elements

Arrays - stores arrays/list/multiple values into one key

Timestamp - ctimestamp

Object - used for embedded documents

Null - used to store a Null value

Symbol - identical to a string; however, it's generally reserved for languages that use a specific symbol type

Date - stores the current date-time in UNIX time format

Object ID - used to store the document's ID

Binary data - This datatype is used to store binary data

Code - stores JavaScript code into the document

Regular expression - stores regular expression

MongoDB Installation Process

Download and Install Mongodb on windows using https://www.mongodb.com/download-center/community

Or

Refer :- https://www.geeksforgeeks.org/how-to-install-mongodb-on-windows/

Download Mongodb Shell using https://www.mongodb.com/try/download/shell

MongoDB Shell Commands

- To clear the screen:

```
cls
```

- To view existing databases:

```
show dbs
```

- To create/connect to an existing database:

```
use <db_name>
```

- To check currently selected database:

```
db
```

- To drop database, first select the database and then drop it by:

```
db.dropDatabase()
```

- To view collections:

```
show collections
```

- To create students collection:

```
db.createCollection("students")
```

- To insert a document to students collection:

```
db.students.insert( {name: "Viren" })
```

If you insert a document in the collection and if, the collection doesn't exist, it will be created automatically

- To drop students collection:

```
db.students.drop()
```

- To query data from collection:

```
db.students.find()
```

- To display the results in a formatted way:

```
db.students.find().pretty()
```

- To query the document on the basis of some condition:

```
db.students.find( {name: "Viren" })

db.students.find( (roll_no: ($gt: 10) })

db.students.find( (roll_no: ($gte: 10} })

db.students.find( [roll_no: ($gte: 10))).limit(5)

db.students.find((roll_no: ($gte: 10))).sort({roll_no: -1})
```

- To query the document on the basis of some condition:

```
db.students.find( (
                Sand (roll no: 10}, {name: "Viren") ]
 })
 db.students.find(
                $or: (roll_no: 10), (name: "Viren"]]
                })
```

- To update the document on the basis of some condition:

```
db.<collection_name>.update(SELECTION_CRITERIA, UPDATED_DATA)

                db.students.update(
                        {roll_no: 5),
                {$set: (mobile: "9876512345"}}
                )
```

It will update only single document

- To update multiple document on the basis of some condition:

```
                db.students.update(
                      {roll_no: 5},
                      {$set: (mobile: "9876512345"}},
                      {multi: true)
                )
```

- To create an index (descending order) on field 'name':

```
                db.students.createIndex({name:-1})
```

- Save a New Document without Specifying an id Field:

```
                db.students.save( (roll_no: 7))
```

It will insert a new student document with a new_id field with a unique Objectid value

- Save a New Document Specifying an id Field:

```
                 db.students.save({_id: 10, roll_no: 8))
```

it will insert a new document if a document with_id=10 doesn't exist, otherwise update the document with_id=10

- To remove all documents from a collection:

```
db.students.remove({})
```

- To remove all documents that match a condition:

```
db.students.remove({roll_no: {$gt>10}})
```

- To remove a single document that match a condition:

```
db.students.remove({roll_no: {$gt > 10)), true)
```