

release - Oct 2009.

column → field  
row → documents

classmate

table → collections

Date \_\_\_\_\_

Page \_\_\_\_\_

# MongoDB

- MongoDB is document ~~base~~ database
- It stores data in JSON format → BSON.
- Stores data in key-value pair.
- Collection → Group of documents which are related
- Records → Documents
- Field values can have number, string, boolean, array or nested documents
- Mongo → Shell on which MongoDB works
- Mongo Atlas → cloud database platform
  - Mongosh → mongo shell in atlas

## MongoDB Query API. User

- Adhoc queries with mongosh, compass
- data transformation
- document join, from diff collection,
- Graph

- Document can be in format of JSON, BSON, XML

## \* Create collection

1. Using `createCollection()` → `db.createCollection("post")`
- `db.insertOne()`

## \* Insert document

- `insertOne()`
- `insertMany()`

`db.posts.insertOne (`

```
{ title : "Post Title 1",  
  body : "Body of post,"  
  category : "News" }
```

)

35 | 3 | 14 -

## \* ~~Insert~~

- insert Or

## \* ~~Find~~

- find() will show all data if not given any input.  
db.posts.find()

- findOne()  
in this if left empty gives 1<sup>st</sup> document.

- It supports projection if

```
db.posts.find( { } , { title: 1 , date: 1 } )
```

1 → include  
0 → exclude

## \* ~~Update~~

- UpdateOne() , updateMany()

- if want to insert document if it is not found, then  
use { upsert:true }

```
db.collection.updateOne(
```

<filter>,

<update>,

<options>

)

filter - selection criteria to update document / where condition is

update - modifications to be made like setting

new value, incrementing value or reset field

Option :- upsert → not found then add document

multi → update multiple if filter matches more than one

## \* Delete.

- Syntax      `db.collection.deleteOne( <filter>, <options> )`

- delete is permanent operation, no undo.

## \* Query Operation.

- Use to retrieve documents from collection that match specific criteria.
- find method is primary way to perform queries.

Syntax      `db.collection.find( <filter>, <projection> )`

### Comparison

<code>\$eq</code> → =	<code>\$gt</code> → <u>a</u> > b	<code>\$lt</code> → <u>a</u> < b
<code>\$ne</code> → ≠	<code>\$gte</code> → <u>a</u> ≥ b	<code>\$lte</code> → <u>a</u> ≤ b

`$in` → matched within array

### Logical

`$and` → both match  
`$or` → either match

`$nor` → both fails  
`$not` → query doesn't match

### Evaluation

- `$regex` → all use of regular expression when evaluating field value
- ~~`$perform $text`~~ → Perform text search
- `$where` → use Javascript expression to match

\* `$addFields()` → it will add one field into existing document.

### \* Index Types

- Single field
- Compound Index
- Multikey Index
- Geospatial Index
- Text Indexes
- Hashed Indexes.

## \* Features of MongoDB

- Open Source document oriented

- Stores JSON.

- Highly Scalable (full Index support)

- & Flexible

- distributed NoSQL database

- High Performance

- High availability

- Map/Reduce for aggregation

- BSON → Binary representation of JSON

## \* Replication.

- It supports replication, replica sets, which are groups of MongoDB servers that maintain same dataset.

- Replication provides high availability by auto-electing new primary node if primary becomes unavailable.

- Secondary nodes replicate data from primary to ensure data redundancy.

- Replica set is a group of mongod instances that maintain same data set.

## \*

## \* Sharding

- It mechanism for horizontal scaling in MongoDB.

- It distributes data across multiple servers (shards) based on shard keys.

- It allows MongoDB to handle large datasets & high-throughput workloads.

- mongos acts as query router, provide interface between client & sharded cluster.

## \* Date

- date data type is 64-bit int. which represent number of millisecond.
- BSON data type generally support UTC datetime

## Commands

### \* db.serverStatus() →

- various stats & metrics related to MongoDB server.

### \* mongodump -h → backup files

### \* db.stats()

- It provide all info & stats of current database

### \* ~~Current~~ use name-database

- create database if not present or return existing database

### \* db.dropDatabase() → drop. db

### \* \$unset → operator use to delete particular field

## \* Config Servers

- It stored meta data & config setting for cluster
- As of MongoDB 3.4, config servers must be deployed as replica set (CSRS).

## \* Capped Collection

- Collections with fixed-size, circular order
- When creating collection, we must fix max size of collection & max. no. of documents that it can store
- It works like circular buffers.
- Once it's filled then new documents is stored overwriting oldest documents in collection.
- It preserves inserted order. As result, don't need index. to return documents. in insertion order
- It support automatic removal of oldest documents
- It have datatype - boolean
- Capped don't allow single document removal  
Can't delete document
- no default indexers present

## \* Shard Keys

- It uses shard keys to distribute collection's documents across shards
- Shard key → fields that exist in every document in target collection
- shard key can't be change after sharding
- Shard collection can have only one-shard key

chunk :

- id = unique id (Object Id)

files\_id :- Object Id of that file (-id & parent)

n : sequence number of chunk

- Insert file more than 16 MB JSON default MP3, MP4.

- It stores files in two collections

chunks → binary chunks (255KB) (fs.chunks)

files → file's metadata (fs.files)

## \* Embedded document

- It's document that is nested within another document

- It way of modeling complex & hierarchical data structures within single document

- It allow to store related data together, makes retrieval & manipulation easier

```
{
  id: 124,
  title: "Intro. to Mongo",
  author: {
    username: "John-doe",
    email: "john@gmail.com"
  }
}
```

date : 23-07-2023

- denormalization allow applications to retrieve & manipulate related data in single database operation

- find And Modify for update

## \* Index.

- Collection can have 64 indexes.
- index name → 125 characters
- Compound index can have 314 fields
- Index are stored in RAM
- Index can be applied to field of sub-document

## \* -id

- 12 bytes :-
  - 4 bytes → seconds
  - 3 bytes → machine identifier
  - 2 bytes → process id
  - 3 → random counter value

- NoSQL is best suited for Larger data Set
- NoSQL is used to store large datasets.
- Apache Cassandra developed by Twitter.  
Facebook, July 2008
- In NoSQL Redis is implemented by store data in Key-Value pair in-memory with optional persistence
- NoSQL structures → Key/Value based, Graph based, Column based
- Cassandra minimizes Disk I/O operation
- NoSQL supports → Horizontal Scaling, Simple design, fine controlled availability
- NoSQL database type → document database
- Cassandra is wide-column store
- Sharding a database across many server instances can be achieved with SAN [Storage Area Network]
- NoSQL database have automatic replication which provide high availability & disaster recovery
- Graph database are used to store info. about network such as social ~~or~~ connections

- MongoDB are written in JavaScript, C, C++
- MongoDB is cross-platform
- Primary classification for NoSQL architecture
  - Document database
  - Graph Database
  - Key-Value database
- MongoDB is adopted as backend software by many major websites & services.
- MongoDB dynamic schema makes polymorphism easier for application.
- MMS [MongoDB Management Services] it supports complete backup soln & full monitoring deployment
- In mongoDB can run over multiple servers, balancing load. Embedded documents & array reduce needs for joins.
- MongoDB can be used as AMS [Application Mng. Services] because of load balancing & data replication.
- MongoDB uses readers-writer lock that allow concurrent read access to database but exclusive write access to single write operation.

- mongotop provides statistics on per-collection level, it give amount of time instance spend.
  - \* on reading & writing data.
- mongoStats give stats & status for currently running instance. provide info. how many inserts, update, remove, queries & commands perform. shall time & memory they use.
- mongostats is same as linux/unix vmstat utility.
- mongoSniff requires Libpcap network library & only available for unix-like system.
- mongoSniff provide low level tracing view into database activity by monitoring network.
- for install mongo mongooplog is simple tool.
- Wire shark is cross platform, open-source sniffer which supports mongo. as network analyzer.
- Mongo import is command to import JSON, CSV or TSV
- Mongo Hub is native OS-X application for MongoDB management.

- opricot is browser based MongoDB shell implement in PHP
- mongodb-enterprise → metapackage
- query shape (query structure) consist of query, sort, projection specification
- Each index in mongodb requires at least 8kB of data space
- To interact with embedded document use dot(.)
- TTL (Time-to-live) is used which allow to auto ~~to~~ remove document from collection after certain time period.  
we mention expireAfterSeconds : 3600.
- files-id is used for datatype of -id for documents in file collections
- Gridfs index allows efficient retrieval of chunk using files-id & n values

- mongo stores time in UTC format
- index of BinData type are fast to retrieval
- it can be use to iterate cursor of document return by `db.collection.find()`
- Covered query is query in which all fields are part of an index & return of document are in same index
  - in this it will not look inside documents
- \$explain() will return query stats.
- \$hint will force to use specified index.