About the Tutorial

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++.

This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

Audience

This tutorial is designed for Software Professionals who are willing to learn MongoDB Database in simple and easy steps. It will throw light on MongoDB concepts and after completing this tutorial you will be at an intermediate level of expertise, from where you can take yourself at higher level of expertise.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of database, text editor and execution of programs, etc. Because we are going to develop high performance database, so it will be good if you have an understanding on the basic concepts of Database (RDBMS).

Copyright & Disclaimer

© Copyright 2018 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.
# Table of Contents

About the Tutorial .................................................................................. i  
Audience ............................................................................................... i  
Prerequisites ......................................................................................... i  
Copyright & Disclaimer ....................................................................... i  
Table of Contents .................................................................................. ii  

MONGODB ......................................................................................... 1  
1. MongoDB – Overview ...................................................................... 2  
2. MongoDB – Advantages ................................................................... 4  
3. MongoDB – Environment .................................................................. 5  
4. MongoDB – Data Modelling ............................................................... 10  
5. MongoDB – Create Database ............................................................ 12  
6. MongoDB – Drop Database ............................................................... 13  
7. MongoDB – Create Collection ......................................................... 14  
8. MongoDB – Drop Collection ............................................................. 16  
9. MongoDB – Datatypes ....................................................................... 17  
10. MongoDB – Insert Document ............................................................. 18  
11. MongoDB – Query Document ............................................................ 20  
12. MongoDB – Update Document ............................................................ 24  
14. MongoDB – Projection ...................................................................... 28  
15. MongoDB – Limit Records ................................................................. 29  
16. MongoDB – Sort Records ................................................................. 31  
17. MongoDB – Indexing ................................................................------- 32  
18. MongoDB – Aggregation ................................................................. 34  
19. MongoDB – Replication ................................................................. 38  
20. MongoDB – Sharding ..................................................................... 41  
21. MongoDB – Create Backup ............................................................. 43
1. MongoDB – Overview

MongoDB 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.

<table>
<thead>
<tr>
<th>RDBMS</th>
<th>MongoDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Database</td>
</tr>
<tr>
<td>Table</td>
<td>Collection</td>
</tr>
<tr>
<td>Tuple/Row</td>
<td>Document</td>
</tr>
<tr>
<td>column</td>
<td>Field</td>
</tr>
<tr>
<td>Table Join</td>
<td>Embedded Documents</td>
</tr>
<tr>
<td>Primary Key</td>
<td>Primary Key (Default key _id provided by mongodb itself)</td>
</tr>
</tbody>
</table>

**Database Server and Client**

| Mysql/Oracle   | mongod                              |
Sample Document
Following example shows the document structure of a blog site, which is simply a comma separated key value pair.

```json
{
   _id: ObjectId(7df78ad8902c)
   title: 'MongoDB Overview',
   description: 'MongoDB is no sql database',
   by: 'tutorials point',
   url: 'http://www.tutorialspoint.com',
   tags: ['mongodb', 'database', 'NoSQL'],
   likes: 100,
   comments: [
      {
         user: 'user1',
         message: 'My first comment',
         dateCreated: new Date(2011,1,20,2,15),
         like: 0
      },
      {
         user: 'user2',
         message: 'My second comments',
         dateCreated: new Date(2011,1,25,7,45),
         like: 5
      }
   ]
}
```

_id is a 12 bytes hexadecimal number which assures the uniqueness of every document. You can provide _id while inserting the 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.
Any relational database has a typical schema design that shows number of tables and the relationship between these tables. While in MongoDB, there is no concept of relationship.

**Advantages of MongoDB over RDBMS**

- **Schema less**: MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Structure of a single object is clear.
- No complex joins.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that’s nearly as powerful as SQL.
- Tuning.
- Ease of scale-out: MongoDB is easy to scale.
- Conversion/mapping of application objects to database objects not needed.
- Uses internal memory for storing the (windowed) working set, enabling faster access of data.

**Why Use MongoDB?**

- Document Oriented Storage: Data is stored in the form of JSON style documents.
- Index on any attribute
- Replication and high availability
- Auto-sharding
- Rich queries
- Fast in-place updates
- Professional support by MongoDB

**Where to Use MongoDB?**

- Big Data
- Content Management and Delivery
- Mobile and Social Infrastructure
- User Data Management
- Data Hub
Let us now see how to install MongoDB on Windows.

**Install MongoDB on Windows**

To install MongoDB on Windows, first download the latest release of MongoDB from [http://www.mongodb.org/downloads](http://www.mongodb.org/downloads). Make sure you get correct version of MongoDB depending upon your Windows version. To get your Windows version, open command prompt and execute the following command.

```
C:\>wmic os get osarchitecture
OSArchitecture
64-bit
C:\>
```

32-bit versions of MongoDB only support databases smaller than 2GB and suitable only for testing and evaluation purposes.

Now extract your downloaded file to c:\ drive or any other location. Make sure the name of the extracted folder is mongodb-win32-i386-[version] or mongodb-win32-x86_64-[version]. Here [version] is the version of MongoDB download.

Next, open the command prompt and run the following command.

```
C:\>move mongodb-win64-* mongodb
1 dir(s) moved.
C:\>
```

In case you have extracted the MongoDB at different location, then go to that path by using command `cd FOOLDER/DIR` and now run the above given process.

MongoDB requires a data folder to store its files. The default location for the MongoDB data directory is c:\data\db. So you need to create this folder using the Command Prompt. Execute the following command sequence.

```
C:\>md data
C:\>md data\db
```

If you have to install the MongoDB at a different location, then you need to specify an alternate path for `\data\db` by setting the path `dbpath` in `mongod.exe`. For the same, issue the following commands.
In the command prompt, navigate to the bin directory present in the MongoDB installation folder. Suppose my installation folder is `D:\set up\mongodb`

```plaintext
C:\Users\XYZ>d:
D:\>cd "set up"
D:\set up>cd mongodb
D:\set up\mongodb>cd bin
D:\set up\mongodb\bin>mongod.exe --dbpath "d:\set up\mongodb\data"
```

This will show **waiting for connections** message on the console output, which indicates that the mongod.exe process is running successfully.

Now to run the MongoDB, you need to open another command prompt and issue the following command.

```plaintext
D:\set up\mongodb\bin>mongo.exe
MongoDB shell version: 2.4.6
connecting to: test
>db.test.save( { a: 1 } )
>db.test.find()
{ 
  "_id" : ObjectId(5879b0f65a56a454), 
  "a" : 1 
}
>
```

This will show that MongoDB is installed and run successfully. Next time when you run MongoDB, you need to issue only commands.

```plaintext
D:\set up\mongodb\bin>mongod.exe --dbpath "d:\set up\mongodb\data"
D:\set up\mongodb\bin>mongo.exe
```

**Install MongoDB on Ubuntu**

Run the following command to import the MongoDB public GPG key –

```plaintext
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
```

Create a `/etc/apt/sources.list.d/mongodb.list` file using the following command.

```plaintext
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen'
| sudo tee /etc/apt/sources.list.d/mongodb.list
```

Now issue the following command to update the repository –

```plaintext
sudo apt-get update
```
Next install the MongoDB by using the following command –

```
apt-get install mongodb-10gen=2.2.3
```

In the above installation, 2.2.3 is currently released MongoDB version. Make sure to install the latest version always. Now MongoDB is installed successfully.

**Start MongoDB**

```
sudo service mongodb start
```

**Stop MongoDB**

```
sudo service mongodb stop
```

**Restart MongoDB**

```
sudo service mongodb restart
```

To use MongoDB run the following command.

```
mongo
```

This will connect you to running MongoDB instance.

**MongoDB Help**

To get a list of commands, type `db.help()` in MongoDB client. This will give you a list of commands as shown in the following screenshot.
MongoDB Statistics

To get stats about MongoDB server, type the command `db.stats()` in MongoDB client. This will show the database name, number of collection and documents in the database. Output of the command is shown in the following screenshot.

![MongoDB Statistics Screenshot](image-url)
4. MongoDB – Data Modelling

Data in MongoDB has a flexible schema. 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.

Some considerations while designing Schema in MongoDB

- Design your schema according to user requirements.
- Combine objects into one document if you will use them together. Otherwise separate them (but make sure there should not be need of joins).
- Duplicate the data (but limited) because disk space is cheap as compare to compute time.
- Do joins while write, not on read.
- Optimize your schema for most frequent use cases.
- Do complex aggregation in the schema.

Example

Suppose a client needs a database design for his blog/website and see the differences between RDBMS and MongoDB schema design. Website has the following requirements.

- Every post has the unique title, description and url.
- Every post can have one or more tags.
- Every post has the name of its publisher and total number of likes.
- Every post has comments given by users along with their name, message, data-time and likes.
- On each post, there can be zero or more comments.

In RDBMS schema, design for above requirements will have minimum three tables.
While in MongoDB schema, design will have one collection post and the following structure:

```json
{
    _id: POST_ID,
    title: TITLE_OF_POST,
    description: POST_DESCRIPTION,
    by: POST_BY,
    url: URL_OF_POST,
    tags: [TAG1, TAG2, TAG3],
    likes: TOTAL_LIKES,
    comments: [
        {
            user: 'COMMENT_BY',
            message: TEXT,
            dateCreated: DATE_TIME,
            like: LIKES
        },
        {
            user: 'COMMENT_BY',
            message: TEXT,
            dateCreated: DATE_TIME,
            like: LIKES
        }
    ]
}
```
So while showing the data, in RDBMS you need to join three tables and in MongoDB, data will be shown from one collection only.
In this chapter, we will see how to create a database in MongoDB.

**The use Command**

MongoDB **use DATABASE_NAME** is used to create database. The command will create a new database if it doesn't exist, otherwise it will return the existing database.

**Syntax**

Basic syntax of **use DATABASE** statement is as follows:

```
use DATABASE_NAME
```

**Example**

If you want to create a database with name `<mydb>`, then **use DATABASE** statement would be as follows:

```
>use mydb
switched to db mydb
```

To check your currently selected database, use the command `db`

```
>db
mydb
```

If you want to check your databases list, use the command `show dbs`.

```
>show dbs
local  0.78125GB
test   0.23012GB
```

Your created database (mydb) is not present in list. To display database, you need to insert at least one document into it.

```
>db.movie.insert({"name":"tutorials point"})
>show dbs
local  0.78125GB
mydb   0.23012GB
```
In MongoDB default database is test. If you didn't create any database, then collections will be stored in test database.

<table>
<thead>
<tr>
<th>Database</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>0.23012GB</td>
</tr>
</tbody>
</table>
In this chapter, we will see how to drop a database using MongoDB command.

**The dropDatabase() Method**

MongoDB **db.dropDatabase()** command is used to drop a existing database.

**Syntax**

Basic syntax of **dropDatabase()** command is as follows:

```
db.dropDatabase()
```

This will delete the selected database. If you have not selected any database, then it will delete default 'test' database.

**Example**

First, check the list of available databases by using the command, **show dbs**.

```
> show dbs
local  0.78125GB
mydb   0.23012GB
test   0.23012GB
>
```

If you want to delete new database `<mydb>`, then **dropDatabase()** command would be as follows:

```
> use mydb
switched to db mydb
> db.dropDatabase()
>{ "dropped" : "mydb", "ok" : 1 }
>
```

Now check list of databases.

```
> show dbs
local  0.78125GB
test   0.23012GB>
```
In this chapter, we will see how to create a collection using MongoDB.

**The createCollection() Method**

MongoDB `db.createCollection(name, options)` is used to create collection.

**Syntax**

Basic syntax of `createCollection()` command is as follows:

```javascript
db.createCollection(name, options)
```

In the command, `name` is name of collection to be created. `Options` is a document and is used to specify configuration of collection.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Name of the collection to be created</td>
</tr>
<tr>
<td>Options</td>
<td>Document</td>
<td>(Optional) Specify options about memory size and indexing</td>
</tr>
</tbody>
</table>

Options parameter is optional, so you need to specify only the name of the collection. Following is the list of options you can use:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>capped</td>
<td>Boolean</td>
<td>(Optional) If true, enables a capped collection. Capped collection is a fixed size collection that automatically overwrites its oldest entries when it reaches its maximum size. <strong>If you specify true, you need to specify size parameter also.</strong></td>
</tr>
<tr>
<td>autoIndexID</td>
<td>Boolean</td>
<td>(Optional) If true, automatically create index on _id field. Default value is false.</td>
</tr>
<tr>
<td>size</td>
<td>number</td>
<td>(Optional) Specifies a maximum size in bytes for a capped collection. <strong>If capped is true, then you need to specify this field also.</strong></td>
</tr>
</tbody>
</table>
max number (Optional) Specifies the maximum number of documents allowed in the capped collection.

While inserting the document, MongoDB first checks size field of capped collection, then it checks max field.

**Examples**

Basic syntax of `createCollection()` method without options is as follows:

```plaintext
>use test
switched to db test
>db.createCollection("mycollection")
{ "ok" : 1 }
>
```

You can check the created collection by using the command `show collections`.

```plaintext
>show collections
mycollection
system.indexes
```

The following example shows the syntax of `createCollection()` method with few important options:

```plaintext
>db.createCollection("mycol", { capped : true, autoIndexID : true, size : 6142800, max : 10000 } )
{ "ok" : 1 }
>
```

In MongoDB, you don't need to create collection. MongoDB creates collection automatically, when you insert some document.

```plaintext
>db.tutorialspoint.insert({"name" : "tutorialspoint"})
>show collections
tutorialspoint
```

21
End of ebook preview
If you liked what you saw...
Buy it from our store @ https://store.tutorialspoint.com