25-03-2017

# **General**

MondoDB is document database. Document is like JSON object. A collection can have multiple documents. Each document has one unique \_id. Document has name value pairs. Values in document can be arrays, another document or array of document.

Analogy with table database.

Database = Database

Collection = Table

Document = Row

Default port for MongoDB is 27017. By default, data is in /data/db

In mongo DB you may not require to create collection explicitly. Whenever a document is inserted in a collection it is auto created.

//good tutorial

<https://www.tutorialspoint.com/mongodb>

mongoDB maintains atomicity at single document level. Either all or none data of a document is updated in an update operation.

# **Installation**

I downloaded mongoDB from its official site and installed it. Create manually a folder c:\data\db. Give command mongod, will start the server. mongo command will open a interactive screen.

# **Theory**

**Map-Reduce** in MongoDB is answer to group by. Map collects documents satisfying certain criteria and reduce returns a new document out of all.

**Capped collection**

They are of fixed size. After the size is full the oldest data is overwritten. Capped collections produce results in natural order i.e the order in which the data was inserted. Unlike normal collections the capped collections must be explicitly created by using createCollection.

You can create an \_id of document in a collection. If you do not create an \_id then mongo DB automatically creates a 12 bytes \_id. Documents in a collection can have different schemas. Avoid need of joins between documents. Join while write and not while read.

## **Database and collections**

//Create database

use dbName

//check currently selected db

db

//show all databases

show dbs

To display a database you need to create at least one document to it. Default database is test. If you not create a database then collections will be stored in test database. db.dropDatabase() deleted the current database.

//Create a collection

db.createCollections(name, options)

//Shows all collections.

show collection

//drop collection

Db.collection\_name.drop()

//insert document

>db.COLLECTION\_NAME.insert(document)

>db.mycol.insert({

\_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

})

ObjectId() above will create an object id. To insert multiple documents insert an array of document objects.

//To query documents in a collection use find() method or find().pretty() to show nicely

>db.mycol.find().pretty()

To query the document on the basis of some condition, you can use following operations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Syntax** | **Example** | **RDBMS Equivalent** |
| Equality | {<key>:<value>} | db.mycol.find({"by":"tutorials point"}).pretty() | where by = 'tutorials point' |
| Less Than | {<key>:{$lt:<value>}} | db.mycol.find({"likes":{$lt:50}}).pretty() | where likes < 50 |
| Less Than Equals | {<key>:{$lte:<value>}} | db.mycol.find({"likes":{$lte:50}}).pretty() | where likes <= 50 |
| Greater Than | {<key>:{$gt:<value>}} | db.mycol.find({"likes":{$gt:50}}).pretty() | where likes > 50 |
| Greater Than Equals | {<key>:{$gte:<value>}} | db.mycol.find({"likes":{$gte:50}}).pretty() | where likes >= 50 |
| Not Equals | {<key>:{$ne:<value>}} | db.mycol.find({"likes":{$ne:50}}).pretty() | where likes != 50 |

## **Nodejs**

//url for nodejs client

<https://docs.mongodb.com>

I used

npm install mongodb

var mongoClient = require('mongodb').MongoClient;

var url = 'mongodb://localhost:27017/test';

let db1;

mongoClient.connect(url, function (err, db) {

    console.log("Connected mongodb");

    db1 = db;

});

router.post('/api/mongo', function (req, res, next) {

    try {

        let data = {

            "address": {

                "street": "2 Avenue",

                "zipcode": "10075",

                "building": "1480",

                "coord": [-73.9557413, 40.7720266]

        };

        db1

            .collection('restaurants')

            .insertOne(data, function (err, result) {

                console.log("Inserted a document into the restaurants collection:", result);

                res

                    .status(200)

                    .json({\_id: result.insertedId});

            });

    } catch (error) {

        let err = new def.NError(500, messages.errInternalServerError, error.message);

        next(err);

    }

});