mongo DB
CheatSheet

CODE HELP

In this Cheatsheet, we will cover the basics of mongo DB. We will provide examples to help you understand how mongo DB work and how to use them in your own web development projects. Whether you are a beginner or an experienced developer, this PDF can serve as a useful reference guide.

MongoDB is a document-oriented, open-source database program that is used for storing and retrieving data. It is a NoSQL database, which means that it does not use a fixed schema and does not rely on tables to organize data, unlike a traditional relational database. Instead, MongoDB stores data in a format called BSON (binary JSON), which allows for more flexibility and scalability when working with large amounts of data. MongoDB is often used for web and mobile applications, real-time analytics, and big data processing.

Most Commonly Used MongoDB terms:

Common DB Terms	MongoDB Terms
Database	Database
Tables	Collections
Rows	Document
Columns	Fields

NOTE: In MongoDB, data are store in **BSON** not in **JSON** format.

ATOMIC/UPDATE OPERATOR

Atomic Operator	Use of the Operator
$set	Updating the stored document without changing the prev. state of the document.
$inc	To increment and decrement on the numerical type variable.
$push	Push new field but it must be of ARRAY type field.
$eq	Matches values that are equal to the given value.
$gt	Matches if values are greater than the given value.
$lt	Matches if values are less than the given value.
$gte	Matches if values are greater or equal to the given value.
$lte	Matches if values are less or equal to the given value.
$in	Matches any of the values in an array.
$ne	Matches values that are not equal to the given value.
$nin	Matches none of the values specified in an array.

MongoDB COMMANDS

- For invoking Mongo Daemon:

```
mongod
```

- For starting the mongo shell:

```
mongo
```

- To show all the database present in the MongoDB:

```
show dbs
```

- To create a new database:

```
use students
```

- To see the current database:

```
db
```

- To delete the current database which we are in:

```
db.dropDatbase()
```

- To show all the collection in the database:

```
show collections
```

- To show all the collection in the database:

```
show collections
```

- To create a new collections:

```
db.createCollection('studentData')
```

- To delete a collection:

```
db.studentData.drop()
```

- To insert a document in the collections:

```
db.studentData.insert({name : "Jack", age : 20})
```

- For inserting only one document into the collections (same work as " .insert() "):

```
db.studentData.insertOne({
        name : "Travour",
        email : "travor458@gmail.com",
        age : 24
})
```

- For inserting multiple document into the collections:

```
db.studentData.insertMany([
   {name : "Harry", email : "marry234@gmail.com", age : 13},
   {name : "Ronit", email : "ronit@outlook.com", age : 45},
   {name : "Adesh", email : "sadhuadesh@gmail.com", age : 18}
])
```

- For displaying the documents present inside the collections:

```
db.studentData.find()
```

- For displaying the documents present inside the collections in a prettyer format:

```
db.studentData.find().pretty()
```

- For displaying the limited number of document present in the collection:

```
db.studentData.find().limit(3).pretty()
```

- For counting the number of document present in the collections:

```
db.studentData.find().count()
```

- For sorting the document in ascending order on the basis of age:

```
db.studentData.find().sort({age : 1})
```

- For sorting the document in descending order on the basis of age:

```
db.studentData.find().sort({age : -1})
```

- For chaining multiple function together:

```
db.studentData.find().limit(2).sort({age : -1}).pretty()
```

- For finding only one document:

```
db.studentData.findOne({age : 18})
```

- For finding a group of documents based on some condition:

```
db.studentData.findOne({age : {$lte : 18}})
```

- For updating a data based upon certain condition or filter: (This will change the entire documents and fields by removing the previous state of the document)

```
db.studentData.update({name : "Harry"} , {fblogged : "Yes" , age : 45})
```

- For updating a data based upon certain condition or filter: (This will change the documents and fields without changing the previous state of the document)

```
db.studentData.updateOne({name : "Harry"} , {$set : {fblogged : "Yes" , age : 45}})
```

- For updating a group of document based on certain condition:

```
db.studentData.updateMany({age : 24} , {$set : {courseCount : 2}})
```

- For adding a new field to all the documents:

```
db.studentData.updateMany({} , {$set : {registered : "Yes"}})
```

- For renaming a field name:

```
db.studentData.updateMany({} , {$rename : {age : "student_age"}})
```

- For incrementing a numeric field in a document:

```
db.studentData.updateMany({} , {$inc : {age : 1}})
```

- For decrementing a numeric field in a document:

```
db.studentData.updateMany(
        {age : {$lt : 19}} ,
        {
                $inc : {courseCount : -1},
                $set {hobby : ["Painting", "Football"]}
        })
```

- To append new data to the array type field:

```
db.studentData.updateMany(
    {age : {$lt : 19}} ,
    {
        $push : {hobby : "Swimming"}
    })
```

- To delete One document using condition or filter:

```
db.studentData.deleteOne({age : 17})
```

- To delete the first document from the table:

```
db.studentData.deleteOne({}})
```

- To delete the first document from the table:

```
db.studentData.deleteMany({fblogged : "Yes"})
```

- To delete the all the documents of the table:

```
db.studentData.deleteMany({})
```

- To group specific field together:
  (This will display the email and _id)

```
db.studentData.find({}, {email : 1})
```

- To group specific field together:
  (_id by default is 1)

```
db.studentData.find({name : "Harry"}, {email : 1, _id : 0, name : 1})
```

- Converting the group of data into an Array:

```
db.studentData.find({name : "Harry"}, {email : 1, _id : 0}).toArray()
```

- To get a particular data from a document:

```
db.studentData.findOne({name : "Adesh"}).email
db.studentData.findOne({name : "Adesh"}).age
db.studentData.findOne({name : "Adesh"}).hobby[1]
```

- How to store a data into a variable:

```
var studentid = db.studentData.findOne({name : "Travour"})._id
studentid
```

- How to fetch data from the variable (studentid):

```
db.studentData.find(_id : studentid)
db.studentData.findOne(_id : studentid).email
```

- How to print all the documents using ForEach Loop
  (If in your collection there is more than 20 documents then '**.find()**' function will display only the first 20 documents and to display the next 20 documents you have type 'it' to display more So, by using for each loop you will see all the documents present in the collections.)

```
db.studentData.find().forEach((student) => {printjson(student)})
```

- How to print specific field from all the documents:

```
db.studentData.find().forEach((student) =>
```