```
/* Cheatsheet for MongoDB
Created by: Truly Mittal
Date: March 2019
*/

/* SHOW DATABASE */
show dbs

/* USE DATABASE */
use DATABASE_NAME

/* current database */
db

/* DROP(delete) DATABASE */
db.dropDatabase()

/* CREATE COLLECTION */
db.createCollection("COLLECTION_NAME", {OPTIONS})

/* DROP(delete) COLLECTION */
db.COLLECTION_NAME.drop()

/* INSERt document */
db.users.insert({
    name: "Truly Mittal",
    age: 29
})

/* gettings all documents */
db.COLLECTION_NAME.find()

/* pretty the output */
db.COLLECTION_NAME.find().pretty()

/* finding a document by _id */
db.users.find({ _id: ObjectId("5c4d7c52eb87a6e0e5fc8949") }).pretty()

/* Finding by a field, here it is name */
db.users.find({name: "Truly Mittal"}).pretty()

/* update document */
db.users.update(
    { name: "Truly Mittal" },
    { $set : { age: 29 } }
)

/* unset a field in document */
db.users.update(
    { name: "Truly Mittal" },
    { $unset : { age: "" } }
)

/* update and auto insert if document not present */
db.users.update(
    { name: "Truly Mittal" },
```

```
    { name: "Truly Mittal" },
    { upsert: true }
)

/* Remove a document */
db.COLLECTION_NAME.remove({query})
db.COLLECTION_NAME.deleteOne({query})

/* insert multiple documents */
db.users.insert(
    [
        {
            name: "Jhon Doe",
            age: 36,
            colors: ["red", "black", "blue"],
            address: {
                city: "paris",
                country: "france"
            }
        },
        {
            name: "Hans Down",
            age: 46,
            colors: ["yellow", "white"],
            address: {
                city: "london",
                country: "uk"
            }
        },
        {
            name: "Eric Widget",
            age: 31,
            colors: ["red", "green", "purple"],
            address: {
                city: "new delhi",
                country: "india"
            }
        },
        {
            name: "Indigo Violet",
            age: 29,
            colors: ["red", "black"],
            address: {
                city: "washington dc",
                country: "us"
            }
        }
    ]
)

/* count the number of documents */
db.users.count()    /* Number of documents */

/* limit the number of fetched records */
db.users.find().limit(2) /* limit the number of records */

/* skip the number of records */
```

```
db.users.find().skip(2)


/* sort the record by the key name (1 Asc, -1 DesC) */
db.users.find().sort({name: 1})     /* sorts the records in asc and desc */

/* Update Multiple docs */
db.users.updateMany(
    {}, //Filter
    {} //Updates
)

db.users.update(
    {}, //Filter
    {}, //Updates
    { multi: true } //required in options if using only update instead of updateMany
)


/* Change the Field Name */
db.users.updateMany(
    {}, //Filter
    { $rename: { "colours" : "colors" } } //Updates
)

/* AND query */
db.users.find({
    $and: [
        { name: "Indigo Violet" },
        { age: 29 }
    ]
})

/* OR query */
db.users.find({
    $or: [
        { name: "Indigo Violet" },
        { age: 29 }
    ]
})

/* OR with Less than equal to query */
db.users.find({
    $or: [
        { name: "John Doe" },
        { age: { $lte : 31 }}
    ]
})

/* not query */
db.users.find({
    age: {
        $not: {
            $lt: 40
        }
    }
```

```
}).pretty()

/* array operations
adds the whole array as an element inside an array */
db.users.update(
    { name: "Truly Mittal" },
    { $addToSet : { colors: ["yellow", "pink"] } }
)

/* adds each element of the array as an element inside an array works as a SET */
db.users.update(
    { name: "Truly Mittal" },
    {
        $addToSet : {
            colors: {
                $each: ["yellow", "pink"]
            }
        }
    }
)
/* repeat above also to show that NOTHING happens */

/* adds each element of the array as an element inside an array NOT works as a SET */
db.users.update(
    { name: "Truly Mittal" },
    {
        $push : {
            colors: {
                $each: ["yellow", "pink"]
            }
        }
    }
)

/* removes all entries from the array */
db.users.update(
    { name: "Truly Mittal" },
    {
        $pull : { colors: ["yellow", "pink"] }
    }
)


db.users.update(
    { name: "Truly Mittal" },
    {
        $pull : { colors: "yellow" }        //removes all enteries of "yellow"
    }
)

db.users.update(
    { name: "Truly Mittal" },
    {
        $pull : { colors: "pink" }
```

```
    }
)

/* The $in operator selects the documents where the value of a field equals any value in the specified array.  */
db.users.update(
    { },
    {
        $pull: {
            colors: {
                $in: [ "red", "black" ]
            }
        }
    },
    { multi: true }
)

/* Increment Operator */
db.users.update(
    { name: "Truly Mittal" },
    {
        $inc : { age: 2 }
    }
)

/* $pop */
db.users.update(
    {name: "Indigo Violet" },
    { $set : {
        scores: [ 3, 65, 2 ]
    }}
)

db.users.update(
    {name: "Indigo Violet" },
    { $pop : { scores: 1 } }
)

db.users.updateMany(
    { },
    {
        $push: {
            colors: {
                $each: [ "red", "black" ]
            }
        }
    }
)

/* Setting the current date in field called "lastModified" */
db.inventory.updateOne(
    { name: "Truly Mittal" },
    { $currentDate: { lastModified: true } }
)

/*
```

```
The $currentDate operator sets the value of a field to the current date, either as a Date or a timestamp. The default type is Date.
{ $currentDate: { <field1>: <typeSpecification1>, ... } }
    <typeSpecification> can be either:
1.  a boolean true to set the field value to the current date as a Date, or
2.  a document { $type: "timestamp" } or { $type: "date" } which explicitly specifies the type.
    The operator is case-sensitive and accepts only the lowercase "timestamp" or the lowercase "date". */


/* Exporting in docs using mongoexport */
mongoexport --db DB_NAME --collection COLLECTION_NAME -o FILENAME.json

/* Exporting in docs using mongoexport as JSON ARRAY */
mongoexport --db DB_NAME --collection COLLECTION_NAME -o FILENAME.json --jsonArray

/* Importing Docs from a file */
mongoimport --db DB_NAME --collection COLLECTION_NAME --fileo FILENAME.json

/* Importing Docs from a file if it is formatted as a JSON ARRAY */
mongoimport --db DB_NAME --collection COLLECTION_NAME --fileo FILENAME.json --jsonArray
```