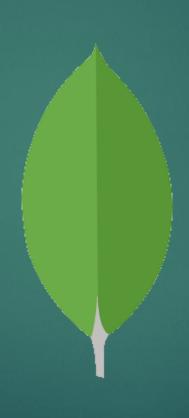


# MongoDB

HOSSEIN FORGHANI MAKTAB SHARIF



#### contents

- Querying documents
- Update documents
- Delete documents
- Limiting results
- Sorting

#### Query Document

▶ To get all the documents:

```
db.COLLECTION_NAME.find([query])
```

▶ To display the results in a formatted way:

```
db.COLLECTION_NAME.find([query]).pretty()
```

► To get only one document:

```
db.COLLECTIONNAME.findOne([query])
```

#### Query Documents – cont.

```
> db.mycol.findOne({title: "MongoDB Overview"})
      " id" : ObjectId("5dd6542170fb13eec3963bf0"),
      "title": "MongoDB Overview",
      "description": "MongoDB is no SQL database",
      "by" : "tutorials point",
      "url" : "http://www.tutorialspoint.com",
      "tags" : [
            "mongodb",
            "database",
            "NoSQL" ],
      "likes" : 100
```

#### Comparison

▶ To get all documents where their "likes" are more than 10:

```
db.mycol.find({"likes": {$gt:10}})
```

▶ To get all documents where their "likes" are equal to 10:

```
db.mycol.find({"likes": {$eq:10}})
```

# All Conditions

Operation	Syntax
Equality	{ <key>:{\$eg;<value>}}</value></key>
Less Than	{ <key>:{\$It:<value>}}</value></key>
Less Than Equals	{ <key>:{\$Ite:<value>}}</value></key>
Greater Than	{ <key>:{\$gt:<value>}}</value></key>
Greater Than Equals	{ <key>:{\$gte:<value>}}</value></key>
Not Equals	{ <key>:{\$ne:<value>}}</value></key>
Value in an array	{ <key>:{\$in:[<value1>, <value2>,<valuen>]}}</valuen></value2></value1></key>
Value not in an array	{ <key>:{\$nin:[<value1>, <value2>,<valuen>]}}</valuen></value2></value1></key>
Array contains all values of an array	{ <key>:{\$all:[<value1>, <value2>,<valuen>]}}</valuen></value2></value1></key>

#### AND, OR

```
AND:
            db.mycol.find({ key1:value1, key2:value2 } )
    db.mycol.find({ $and: [ {key1:value1}, { key2:value2} ] } )
 ▶ OR: db.mycol.find( { $or: [ {key1: value1}, {key2:value2} ] } )
 Example:
db.mycol.find(
{$or:[{gender: "female"}, {age:{$lt: 18}}, {status: "exempt"}]}
```

#### AND & OR Together

```
db.mycol.find( {
    "likes": {$gt:10},
    $or: [{"by": "tutorials point"}, {"title": "MongoDB Overview"}]})
```

#### NOT, NOR

```
{ field: { $not: { <operator-
expression> } }
```

▶ To get documents that fail all the query expressions:

```
{ $nor: [
{ <expression1> }, { <expression2> }, ...
{<expressionN> }
] }
```

## Update Document

```
db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)
```

Example:

# Update Options

- upsert: If set to true, creates a new document when no document matches the query criteria
- multi: If set to true, updates multiple documents that meet the query criteria, otherwise just one! The default value is false
- ► For more options refer to the references

```
db.mycol.update(
    {'title':'MongoDB Overview'},
    {$set:{'title':'New MongoDB Tutorial'}},
    {multi:true}
)
```

#### Save Function

```
db.COLLECTION_NAME.save(<document>)
```

- If the document does not contain an \_id field, then the save() method is like an insert
- ▶ If the document contains an \_id field, then the save() method is equivalent to an update with the query predicate on the \_id field

#### Update Document – cont.

▶ Update and return the updated document:

```
db.COLLECTION_NAME.findOneAndUpdate(<filter>, <update>)
```

Update the first document and return a document containing matchedCount and modifiedCount:

```
db.COLLECTION_NAME.updateOne(<filter>, <update>)
```

Update all document and return a document containing matchedCount and modifiedCount:

```
db.COLLECTION NAME.updateMany(<filter>, <update>)
```

#### Delete Document

▶ To remove documents based on a criteria:

```
db.COLLECTION_NAME.remove(DELLETION_CRITTERIA)
```

▶ To remove just one document based on a criteria:

```
db.COLLECTION_NAME.remove(DELETION_CRITERIA, true)
```

▶ To remove all documents:

```
db.COLLECTION NAME.remove({})
```

#### Projection

- Specify which fields to get?
- ▶ Use a document:

```
> db.mycol.find({},{"title":1,_id:0})
{"title":"MongoDB Overview"}
{"title":"NoSQL Overview"}
{"title":"Tutorials Point Overview"}
```

\_id is included by default

# Projection – cont.

Deep in embedded documents:

# Limiting Records

```
db.COLLECTION_NAME.find().limit(NUMBER)
```

db.COLLECTION NAME.find().limit(NUMBER).skip(NUMBER)

## Sorting Records

▶ Set 1 for ascending and -1 for descending

```
db.COLLECTION_NAME.find().sort({KEY:1})
```

```
> db.mycol.find({},{"title":1,_id:0}).sort({"title":-1})
{"title":"Tutorials Point Overview"}
{"title":"NoSQL Overview"}
{"title":"MongoDB Overview"}
```

You can set multiple keys with different sort modes

#### References

- https://www.tutorialspoint.com/mongodb
- https://docs.mongodb.com/manual/reference

# Any Question?