Updating, Indexing, and Deleting Documents in NoSQL

Updating Documents

updateOne / updateMany

- Change a single document

db.CollectionName.updateOne({field: value}, {$set : {“field”: NewValue}}

- Change multiple documents

db.CollectionName.updateMany({field: value}, {$set : {“field”:NewValue}}

upsert

- Update + insert
- Adds a new value if the old one doesn’t exist

- Boolean toggle of true or false

Update Operators

- $currentDate : Uses today’s date
- $inc : Moves a value up or down
- $min : Update if a value is less than
- $max : Update if a value is more than
- $mul : Multiply a value
- $set : Replace with a new value
- $unset : Delete a field
- $rename : Change the name of a field

$inc

- Changes a value by a certain number
- Only for numbers

db.CollectionName.update({field: value}, {$inc: {field: inc#}})

$mul

- Multiplies a value
- Only for numbers

db.CollectionName.update({field: value}, {$mul: {field: mul#}})

$set

- Replaces a value
- Use with both numbers and strings

db.CollectionName.update({field: oldValue}, {$set: {field: ‘newValue’}})

$unset

- Delete a field

db.CollectionName.update({field: value}, {$unset: {field: ‘ ’}})

$rename

- Change the name of a field

db.CollectionName.updateMany({}, {$rename: {oldFieldName: ‘newFieldName’}})

Deleting

deleteOne / deleteMany

- Delete a single document

db.CollectionName.deleteOne({field: value})

- Delete multiple documents

db.CollectionName.deleteMany({field: value})

Delete a Collection

- Remove the entire set of documents

db.collectionName.drop()

Indexing

createIndex()

- Indexes make it easier and quicker to find documents

- 1 = ascending order
- -1 = descending order

db.collectionName.createIndex({field: indexValue})

Remove Index

```
db.collectionName.dropIndex({field: indexValue})
```