**MongoDB**

**Create**

db.coll.insertOne({name: "Max"})

db.coll.insertMany([{name: "Max"}, {name:"Alex"}]) // ordered bulk insert

db.coll.insertMany([{name: "Max"}, {name:"Alex"}], {ordered: false}) // unordered bulk insert

db.coll.insertOne({date: ISODate()})

db.coll.insertOne({name: "Max"}, {"writeConcern": {"w": "majority", "wtimeout": 5000}})

**Read**

db.coll.findOne() // returns a single document

db.coll.find() // returns a cursor - show 20 results - "it" to display more

db.coll.find().pretty()

db.coll.find({name: "Max", age: 32}) // implicit logical "AND".

db.coll.find({date: ISODate("2020-09-25T13:57:17.180Z")})

db.coll.find({name: "Max", age: 32}).explain("executionStats") // or "queryPlanner" or "allPlansExecution"

db.coll.distinct("name")

// Count

db.coll.countDocuments({age: 32}) // alias for an aggregation pipeline - accurate count

db.coll.estimatedDocumentCount() // estimation based on collection metadata

// Comparison

db.coll.find({"year": {$gt: 1970}})

db.coll.find({"year": {$gte: 1970}})

db.coll.find({"year": {$lt: 1970}})

db.coll.find({"year": {$lte: 1970}})

db.coll.find({"year": {$ne: 1970}})

db.coll.find({"year": {$in: [1958, 1959]}})

db.coll.find({"year": {$nin: [1958, 1959]}})

// Logical

db.coll.find({name:{$not: {$eq: "Max"}}})

db.coll.find({$or: [{"year" : 1958}, {"year" : 1959}]})

db.coll.find({$nor: [{price: 1.99}, {sale: true}]})

db.coll.find({

$and: [

{$or: [{qty: {$lt :10}}, {qty :{$gt: 50}}]},

{$or: [{sale: true}, {price: {$lt: 5 }}]}

]

})

// Element

db.coll.find({name: {$exists: true}})

db.coll.find({"zipCode": {$type: 2 }})

db.coll.find({"zipCode": {$type: "string"}})

// Aggregation Pipeline

db.coll.aggregate([

{$match: {status: "A"}},

{$group: {\_id: "$cust\_id", total: {$sum: "$amount"}}},

{$sort: {total: -1}}

])

// Text search with a "text" index

db.coll.find({$text: {$search: "cake"}}, {score: {$meta: "textScore"}}).sort({score: {$meta: "textScore"}})

// Regex

db.coll.find({name: /^Max/}) // regex: starts by letter "M"

db.coll.find({name: /^Max$/i}) // regex case insensitive

// Array

db.coll.find({tags: {$all: ["Realm", "Charts"]}})

db.coll.find({field: {$size: 2}}) // impossible to index - prefer storing the size of the array & update it

db.coll.find({results: {$elemMatch: {product: "xyz", score: {$gte: 8}}}})

// Projections

db.coll.find({"x": 1}, {"actors": 1}) // actors + \_id

db.coll.find({"x": 1}, {"actors": 1, "\_id": 0}) // actors

db.coll.find({"x": 1}, {"actors": 0, "summary": 0}) // all but "actors" and "summary"

// Sort, skip, limit

db.coll.find({}).sort({"year": 1, "rating": -1}).skip(10).limit(3)

// Read Concern

db.coll.find().readConcern("majority")

**Update**

db.coll.updateOne({"\_id": 1}, {$set: {"year": 2016, name: "Max"}})

db.coll.updateOne({"\_id": 1}, {$unset: {"year": 1}})

db.coll.updateOne({"\_id": 1}, {$rename: {"year": "date"} })

db.coll.updateOne({"\_id": 1}, {$inc: {"year": 5}})

db.coll.updateOne({"\_id": 1}, {$mul: {price: NumberDecimal("1.25"), qty: 2}})

db.coll.updateOne({"\_id": 1}, {$min: {"imdb": 5}})

db.coll.updateOne({"\_id": 1}, {$max: {"imdb": 8}})

db.coll.updateOne({"\_id": 1}, {$currentDate: {"lastModified": true}})

db.coll.updateOne({"\_id": 1}, {$currentDate: {"lastModified": {$type: "timestamp"}}})

// Array

db.coll.updateOne({"\_id": 1}, {$push :{"array": 1}})

db.coll.updateOne({"\_id": 1}, {$pull :{"array": 1}})

db.coll.updateOne({"\_id": 1}, {$addToSet :{"array": 2}})

db.coll.updateOne({"\_id": 1}, {$pop: {"array": 1}}) // last element

db.coll.updateOne({"\_id": 1}, {$pop: {"array": -1}}) // first element

db.coll.updateOne({"\_id": 1}, {$pullAll: {"array" :[3, 4, 5]}})

db.coll.updateOne({"\_id": 1}, {$push: {"scores": {$each: [90, 92]}}})

db.coll.updateOne({"\_id": 2}, {$push: {"scores": {$each: [40, 60], $sort: 1}}}) // array sorted

db.coll.updateOne({"\_id": 1, "grades": 80}, {$set: {"grades.$": 82}})

db.coll.updateMany({}, {$inc: {"grades.$[]": 10}})

db.coll.updateMany({}, {$set: {"grades.$[element]": 100}}, {multi: true, arrayFilters: [{"element": {$gte: 100}}]})

// FindOneAndUpdate

db.coll.findOneAndUpdate({"name": "Max"}, {$inc: {"points": 5}}, {returnNewDocument: true})

// Upsert

db.coll.updateOne({"\_id": 1}, {$set: {item: "apple"}, $setOnInsert: {defaultQty: 100}}, {upsert: true})

// Replace

db.coll.replaceOne({"name": "Max"}, {"firstname": "Maxime", "surname": "Beugnet"})

// Write concern

db.coll.updateMany({}, {$set: {"x": 1}}, {"writeConcern": {"w": "majority", "wtimeout": 5000}})

**Aggregate**

<https://www.mongodb.com/docs/manual/reference/method/db.collection.aggregate/>

**Groupby**

https://www.mongodb.com/docs/manual/reference/operator/aggregation/group/