**MongoDB official Documentation** - <https://www.mongodb.com/docs/>

**Creating a completely-managed data base** - https://www.mongodb.com/atlas/database

**Mongo compass installation** - https://www.mongodb.com/docs/compass/current/install/

[ Optional ] **MongoDB Installation** - https://www.mongodb.com/docs/manual/installation/

[ Optional ] **Mongo shell installation** -

https://www.mongodb.com/try/download/shell

Recommended books to learn MongoDB -

1. https://www.oreilly.com/library/view/mongodb-the-definitive/9781491954454/
2. https://www.amazon.in/MongoDB-Workshop-Interactive-Approach-Learning/dp/1839210648
3. https://www.amazon.in/Mastering-MongoDB-4-x-high-fault-tolerant/dp/1789617871
4. https://www.amazon.com/MongoDB-Action-Kyle-Banker/dp/1935182870
5. https://www.amazon.in/Seven-Databases-Weeks-Eric-Redmond/dp/1934356921

Playgrounds ->

1. https://www.humongous.io/app/playground/mongodb/new
2. https://mongoplayground.net/
3. https://www.mongodb.com/docs/manual/tutorial/insert-documents/

***HANDS-ON***

EXAMPLE 1 -> **count documents** (Dataset 1)

db.collection.countDocuments({})

EXAMPLE 2 -> **Find docs** (Dataset 1)

db.collection.find( { } )

EXAMPLE 3 -> **$and** (Dataset 1)

db.collection.find({

$and: [

{

capital: "Washington, D.C."

},

{

name: "United States"

}

]

})

EXAMPLE 4 -> **$or** (Dataset 1)

db.collection.find({

$or: [

{

capital: "Washington, D.C."

},

{

capital: "Canberra"

}

]

})

EXAMPLE 5 -> **$in** (Dataset 1)

db.collection.find({

$or: [

{

"capital": "Washington, D.C."

},

{

population: {

$in: [

25681300,

125960000

]

}

}

]

})

EXAMPLE 6 -> **$lt** (Dataset 1)

db.collection.find({

$or: [

{

population: {

$lt: 125960000

}

},

{

population: {

$in: [

25681300,

328239523

]

}

}

]

})

EXAMPLE 7 -> **$gt** (Dataset 1)

db.collection.find({

$or: [

{

population: {

$gt: 210147124

}

},

{

population: {

$in: [

125960000,

25681300,

328239523

]

}

}

]

})

EXAMPLE 8 -> **$eq** (Dataset 1)

db.collection.find({

$or: [

{

name: {

$eq: "Australia"

}

},

{

population: {

$eq: 125960000

}

}

]

})

EXAMPLE 9 -> **$ ne**

db.collection.find({

$or: [

{

name: {

$eq: "Australia"

}

},

{

name: {

$ne: "United States"

}

}

]

})

EXAMPLE 10 -> **$nin**

db.collection.find({

$or: [

{

population: {

$nin: [

328239523,

25681300,

125960000

]

}

},

{

name: "Brazil"

}

]

})

EXAMPLE 11 -> **$gte**

db.collection.find({

population: {

$gte: 125960000

}

})

EXAMPLE 12- > **$lte**

db.collection.find({

population: {

$lte: 125960000

}

})

EXAMPLE 13 -> **$nor**

db.collection.find({

$nor: [

{

population: "210147125"

},

{

population: "125960000"

}

]

})

EXAMPLE 14 -> **$exists**

SWITCH to MONGOPLAYGROUND from this example onwards

Add extra record in the data ->

{

\_id: "62e5288f4d0440f7811d142d",

name: "India",

capital: "Delhi",

continent: "Asia",

language: "Hindi",

},

QUERY ->

\*\*\*

db.collection.find({

population: {

$exists: true,

$nin: [

210147125,

125960000

]

}

})

EXAMPLE 15 -> **update query**

db.collection.update({

\_id: "62e5288f4d0440f7811d1928"

},

{

$set: {

"capital": "Dubai",

"language": "arabic",

"name": "UAE"

}

})

EXAMPLE 16 -> **$rename**

db.collection.update({

name: "United States"

},

{

$rename: {

"capital": "capital city",

"continent": "kontinent",

}

})

EXAMPLE 17 -> **$inc**

db.collection.update({

name: "United States"

},

{

$inc: {

population: -2

}

})

EXAMPLE 18 -> **$min**

db.collection.update({

name: "United States"

},

{

$min: {

population: 20

}

})

EXAMPLE 19 -> **$max**

db.collection.update({

name: "United States"

},

{

$max: {

population: 40

}

})

EXAMPLE 20 -> **$mul**

db.collection.update({

name: "United States"

},

{

$mul: {

population: 2

}

})

EXAMPLE 21 -> **$unset**

db.collection.update({

name: "United States"

},

{

$unset: {

capital: "",

continent: ""

}

})

EXAMPLE 22 -> **Array ops [ ‘$’ operator ]** (Dataset 2)

db.collection.update({

\_id: 1,

grades: 80

},

{

$set: {

"grades.$": 82

}

})

EXAMPLE 23 -> **Array ops [ ‘.’ operator ]** (Dataset 3)

db.collection.update({

\_id: 4,

"grades.grade": 80

},

{

$set: {

"grades.$.std": 6

}

})

EXAMPLE 24 -> **Array ops [ $elematch ]** (Dataset 3)

elematch returns documents that contain an array field with **at least one** element that matches all the specified query criteria.

db.collection.update({

\_id: 4,

grades: {

$elemMatch: {

grade: {

$lte: 90

},

mean: {

$gt: 80

}

}

}

},

{

$set: {

"grades.$.std": 6

}

})

EXAMPLE 25 -> **embedded data** (Dataset 4)

db.collection.find({

size: {

h: 14,

w: 21,

uom: "cm"

}

})

EXAMPLE 26 -> **embedded data ‘.’ notation** (Dataset 4)

db.collection.find({

"size.uom": "in"

})

EXAMPLE 27 -> **embedded data, mixing $gt with ‘.’ notation** (Dataset 4)

EXAMPLE 28 -> **more criteria matching** (Dataset 4)

EXAMPLE 29 -> **Add to set** (Dataset 5)

EXAMPLE 30 -> **Array add to set** (Dataset 5)

EXAMPLE 31 -> **$pop** (Dataset 6)

EXAMPLE 32 -> **$pull** (Dataset 7)

EXAMPLE 33 -> **multi-pull** (Dataset 8)

EXAMPLE 34 -> **$all** (Dataset 9)

EXAMPLE 35 -> **$all with $elematch** (Dataset 9)

EXAMPLE 36 -> **compare elematch without elematch** (Dataset 10)

EXAMPLE 37 -> **$push** ( Dataset 8 )

EXAMPLE 38 -> **$push with $each for adding multiple values to array** ( Dataset 8 )

EXAMPLE 39 -> **$push with multiple modifiers** ( Dataset 11 )

EXAMPLE 40 -> **$pullall** ( Dataset 12 )

***Datasets*** ->

**DATASET 1** ->

[

{

\_id: "62e5288f4d0440f7811d1928",

name: "United States",

capital: "Washington, D.C.",

continent: "North America",

language: "English",

population: 328239523,

},

{

\_id: "62e5288f4d0440f7811d192b",

name: "Australia",

capital: "Canberra",

continent: "Australia",

language: "English",

population: 25681300,

},

{

\_id: "62e5288f4d0440f7811d192c",

name: "Japan",

capital: "Tokyo",

continent: "Asia",

language: "Japanese",

population: 125960000,

},

{

\_id: "62e5288f4d0440f7811d192d",

name: "Brazil",

capital: "Brasília",

continent: "South America",

language: "Portuguese",

population: 210147125,

},

]

**DATASET 2** ->

[

{

"\_id": 1,

"grades": [

85,

80,

80

]

},

{

"\_id": 2,

"grades": [

88,

90,

92

]

},

{

"\_id": 3,

"grades": [

85,

100,

90

]

}

]

**DATASET 3** ->

[

{

\_id: 4,

grades: [

{ grade: 80, mean: 75, std: 8 },

{ grade: 85, mean: 90, std: 5 },

{ grade: 85, mean: 85, std: 8 }

]

}

]

**DATASET 4** ->

[

{

item: "journal",

qty: 25,

size: {

h: 14,

w: 21,

uom: "cm"

},

status: "A"

},

{

item: "notebook",

qty: 50,

size: {

h: 8.5,

w: 11,

uom: "in"

},

status: "A"

},

{

item: "paper",

qty: 100,

size: {

h: 8.5,

w: 11,

uom: "in"

},

status: "D"

},

{

item: "planner",

qty: 75,

size: {

h: 22.85,

w: 30,

uom: "cm"

},

status: "D"

},

{

item: "postcard",

qty: 45,

size: {

h: 10,

w: 15.25,

uom: "cm"

},

status: "A"

}

]

**DATASET 5** ->

[

{

\_id: 1,

colors: [

"blue, green, red"

]

}

]

**DATASET 6** ->

[

{

\_id: 1,

scores: [

8,

9,

10

]

}

]

**DATASET 7** ->

[

{

\_id: 1,

fruits: [

"apples",

"pears",

"oranges",

"grapes",

"bananas"

],

vegetables: [

"carrots",

"celery",

"squash",

"carrots"

]

},

{

\_id: 2,

fruits: [

"plums",

"kiwis",

"oranges",

"bananas",

"apples"

],

vegetables: [

"broccoli",

"zucchini",

"carrots",

"onions"

]

}

]

**DATASET 8** ->

[

{

\_id: 1,

votes: [

3,

5,

6,

7,

7,

8

]

}

]

**DATASET 9** ->

[

{

\_id: ObjectId("5234cc89687ea597eabee675"),

code: "xyz",

tags: [

"school",

"book",

"bag",

"headphone",

"appliance"

],

qty: [

{

size: "S",

num: 10,

color: "blue"

},

{

size: "M",

num: 45,

color: "blue"

},

{

size: "L",

num: 100,

color: "green"

}

]

},

{

\_id: ObjectId("5234cc8a687ea597eabee676"),

code: "abc",

tags: [

"appliance",

"school",

"book"

],

qty: [

{

size: "6",

num: 100,

color: "green"

},

{

size: "6",

num: 50,

color: "blue"

},

{

size: "8",

num: 100,

color: "brown"

}

]

},

{

\_id: ObjectId("5234ccb7687ea597eabee677"),

code: "efg",

tags: [

"school",

"book"

],

qty: [

{

size: "S",

num: 10,

color: "blue"

},

{

size: "M",

num: 100,

color: "blue"

},

{

size: "L",

num: 100,

color: "green"

}

]

},

{

\_id: ObjectId("52350353b2eff1353b349de9"),

code: "ijk",

tags: [

"electronics",

"school"

],

qty: [

{

size: "M",

num: 100,

color: "green"

}

]

}

]

**DATASET 10** ->

[

{

"\_id": 1,

"results": [

{

"product": "abc",

"score": 10

},

{

"product": "xyz",

"score": 5

}

]

},

{

"\_id": 2,

"results": [

{

"product": "abc",

"score": 8

},

{

"product": "xyz",

"score": 7

}

]

},

{

"\_id": 3,

"results": [

{

"product": "abc",

"score": 7

},

{

"product": "xyz",

"score": 8

}

]

},

{

"\_id": 4,

"results": [

{

"product": "abc",

"score": 7

},

{

"product": "def",

"score": 8

}

]

}

]

**Dataset 11 ->**

[

{

"\_id": 5,

"quizzes": [

{

"wk": 1,

"score": 10

},

{

"wk": 2,

"score": 8

},

{

"wk": 3,

"score": 5

},

{

"wk": 4,

"score": 6

}

]

}

]

**Dataset 12 ->**

[

{

\_id: 1,

scores: [

0,

2,

5,

5,

1,

0

]

}

]